Logistic Sigmoidal and Neural Network Modeling for COVID-19 Death Waves

Oliver Amadeo Vilca Huayta
Departamento de Ingeniería de Sistemas, Universidad Nacional del Altiplano, Peru
E-mail: ovilca@unap.edu.pe

Keywords: Brazil, COVID-19, death number, Italy, logistic sigmoidal model, pandemic, Peru, Switzerland, artificial neural networks

Received: March 3, 2025

The rapid spread of the pandemic of the coronavirus disease 2019 (COVID-19) has caused enormous problems and many deaths. Therefore, it is essential to construct epidemiological models for forecasting and prevention. The main objective of this study is to develop a novel model $F(x)$, treating the cumulative number of deaths due to COVID-19 using two approaches: the logistic sigmoidal function and an artificial neural network. In addition, to estimate models for the death rate $F'(x)$. The research is longitudinal. Data were downloaded from Johns Hopkins University for four countries. It is shown that the logistic function is efficient within the basic sigmoidal functions, and a new variant of the function is obtained and used. The contribution of this work is the model that can be used on data that are not necessarily epidemiological and with any function, where, as x approaches positive and negative infinity, the function tends to a constant value. Also, the method of its construction and the calculation of high-order derivatives allow for the development of a practical model, as well as justification of the term "Wave(s)". The turning points (i.e., the place where the concavity changes) are also obtained and confirmed. Finally, they were applied to real-world datasets. The sigmoidal models yielded better fits than previous works with Pearson correlation coefficients of 0.99992 in Italy, 0.99993 in Brazil, 0.99992 in Switzerland, and 0.99991 in Peru; on the other hand, the Neural Networks reported root mean square errors of 0.0252, 0.0176, 0.0226, and 0.0132, respectively. The models are representative and predictive; they are helpful to understand the pandemic and improve future public health responses.

Povzetek: Prispevek predstavi novo sigmoidalno in nevronsko mrežno modeliranje COVID-19 smrti v štirih državah s ciljem natančnega napovedovanja in razumevanja epidemičnih valov.

1 Introduction

The coronavirus disease 2019 (COVID-19) is highly contagious disease that has caused a high number of deaths and continues to damage the economy and society, to name just a few of its impacts. On October 4, 2024, the Heads of the International Monetary Fund (IMF), the World Bank Group (WBG), and the World Health Organization (WHO) have agreed on broad principles for cooperation on pandemic preparedness (https://www.imf.org; accessed on 7 October 2024). The development and study of epidemiological models can be essential to prevent, predict, or mitigate future epidemics or pandemics. They are also necessary for decision-making, for example, allowing public health institutions to take reliable measures to preserve people's lives [1, 2]. In addition, comparison between countries is considered essential for the control of COVID-19 [3].

In this study, four countries were selected for the purposes of modeling, comparison, and research, namely, Italy, Brazil, Switzerland, and Peru, for which COVID-19 has resulted in a total of 188,322, 333,188, 14,210, and 219,539 deaths, respectively, according to the accumulated death time-series of Johns Hopkins University (https://coronavirus.jhu.edu; accessed on 1 January 2024), without considering the deaths registered after the end of the pandemic. The number of deaths and mortality rates vary between countries and, in this work, we demonstrate that they follow a strict sigmoidal behavior typical of epidemics or pandemics.

Due to the definition of its formula $f(x) = a \cdot e^{-e^{b-c\cdot x}}$ (i.e., double exponential), the Gompertz model requires more elementary operations than the most basic sigmoidal functions. Therefore, this research demonstrated that the logistic function is efficient within the basic sigmoidal functions. Given the logistic function's structure and functionality, we recommend its use in future work.

In addition, a new variant of the logistic function was adapted and obtained, which was called the logistic2 function or simply the logistic function $g(x) = \frac{H}{1+2^{A-Bx}}$, which does not lose any of the original function's properties or qualities. Furthermore, both functions are equivalent.

One of the significant contributions of this work is the model and its construction method. Each function $g(x)_i$ corresponds to wave i. It has the following characteristics or advantages: It does not require any form of integration or union of functions, does not need preliminary work, and it does not require the addition of a dummy variable to the data (or to the function) as in previous works [4, 5]. Mainly

Table 1: Main differences in relation to the key publication

In this work	In Key publication
Estimating any horizontal turning point as a preliminary procedure to obtain $F(x)$ is unnecessary.	Calculating the horizontal turning points is necessary.
The logistic sigmoidal function was used.	The Boltzmann sigmoidal function was used.
A model F was obtained, whose elements are extended to functions that are not necessarily sigmoidal, with $\lim_{x\to-\infty} h(x) = L$ and $\lim_{x\to\infty} h(x) = R$, where L and R are constants. Finally, it was demonstrated with mathematical induction. It was applied to four case studies, where the Peruvian model is: $F(x) = \frac{85.94}{1+2^{6.07-0.048*x}} + \frac{114.1}{1+2^{15.96-0.041*x}} + \frac{13.22}{1+2^{44.35-0.063*x}} + \frac{3.4}{1+2^{73.16-0.082*x}} + \frac{3.33}{1+2^{46.8-0.045*x}}$	The Peruvian model is: $F(x,q) = \frac{(2-q)(3-q)(4-q)(5-q)}{24}\left\{\frac{88.05}{1+e^{\frac{(128.95-x)}{30.88}}}\right\} - \frac{(1-q)(3-q)(4-q)(5-q)}{6}\left\{\frac{113.13}{1+e^{\frac{(390.65-x)}{34.36}}} + 86.49\right\} + \frac{(1-q)(2-q)(4-q)(5-q)}{4}\left\{\frac{14.18}{1+e^{\frac{(698.06-x)}{26.51}}} + 199.34\right\} - \frac{(1-q)(2-q)(3-q)(5-q)}{6}\left\{\frac{3.61}{1+e^{\frac{(898.54-x)}{17.99}}} + 213.37\right\} + \frac{(1-q)(2-q)(3-q)(4-q)}{24}\left\{\frac{2.84}{1+e^{\frac{(1039.82-x)}{25.42}}} + 216.95\right\}$
$\rho = 0.999905$ (the largest). The vertical turning points and higher-order derivatives were calculated.	$\rho = 0.999$. The vertical turning points and higher-order derivatives were not calculated.

for this reason and the previous ones, it is more efficient:

$$F(x) = \sum_{i=1}^{n} \frac{H_i}{1 + 2^{A_i - B_i x}}$$

Where $F(x)$ is the expected cumulative number of deaths, x is the number of days since the first case, n is the number of waves, $1 \leq i \leq n$, H_i is the height of the wave i, A_i and B_i are constants related to the pandemic.

The main objective of this study is to develop a novel and efficient model $F(x)$, treating the cumulative number of deaths due to COVID-19 since the pandemic began as a variable, using the logistic function. Furthermore, artificial neural networks (ANNs) are used, which constitute an alternative method, as machine learning has shown powerful results for studying COVID-19 [6, 7, 8].

The first derivative is calculated to explain the death rate and the magnitude of the COVID-19 waves, where even its graph represents the considered characteristics. In this way, justification is provided for the denomination of COVID-19 "waves." Next, the second derivative is used to calculate the vertical turning points (i.e., the point where the sigmoidal function changes from concave upward to concave downward). At this point, only one last issue remains: the second derivative can report false turning points. Therefore, the turning points are confirmed using the third derivative. Using high-order derivatives—specifically, the first three—allows for an elegant and irrefutable mathematical analysis of any model. Then, the proposed model is applied to four representative countries, yielding novel results.

The differences between the ANN and sigmoidal models were minimal. The ANN's disadvantage is the calculation time.

This paper is structured as follows: Section 2 describes related works, Section 3 presents the materials and methods, and describes the used data. Section 4 demonstrates the results, including those related to the ANN model, the coupled model, high-order derivatives, and the case study, while Section 5 discusses the findings and previous work. Finally, the conclusion is presented in Section 6.

2 Related works

Concerning key publications: While one study has focused on the Peruvian case, it was limited to calculating a function with the Gompertz model and did not apply ANNs. It was carried out with incomplete data as it was conducted during the pandemic period [4]. There is also a study in which $F(x)$ was calculated using the Boltzmann sigmoidal function for a case study (Peru), which was correlated with the social isolation measures in Peru (qualitative variable) [5]; Table **??** illustrates the main differences.

Regarding ANN, several works have used artificial neural networks for prediction. Research has been carried out in India on the Multilayer perceptron [9]. In the United States and India, various artificial neural networks have been used with data on confirmed infections and deaths from COVID-19, of which the convolutional LSTM stood out, which had greater precision and a lower error [10]. In Brazil, a long short-term memory (LSTM) model has been used [11]. Additionally, in China and other countries, it has been concluded that ANNs are adequate to predict global infections and deaths due to COVID-19 [12]. Finally, there is a study in Turkey [13] and in Ada County (Idaho) [14]. A total of six studies were included. The studies were published between 2020 and 2024. Table 2 summarizes the key characteristics of the included studies, detailing the year, countries with metrics (i.e., Pearson correlation coefficient or MSE), key findings, and limitations.

There have not been many related studies on using the logistic sigmoidal function to analyze pandemics. One study

Table 2: Summarizes the six studies included in the review (ANN). It shows the authors, publication year, countries - metrics, key findings, and limitations.

Author	Year	Countries and metrics	Key findings and limitations
Sujath, R. et al. [9]	2020	Pearson correlation: $\rho_{India} \approx 0.968$ confirmed deaths	Feedforward artificial neural network (FANN). They worked with preliminary data (two months).
Shastri, Sourabh et al. [10]	2020	It does not present correlation coefficients. They use Mean Absolute Percentage Error (MAPE) for India and USA	Long short-term memory. They worked with preliminary data (approximately four months) and presented a comparative study of the two countries' results (India and USA).
Fernandes, Filipe et al. [11]	2022	Pearson correlation: $\rho_{Brazil} \approx 0.9826$, $R^2 = 0.9656$ for confirmed deaths	Long short-term memory (LSTM) and others. A study was conducted for the State of Santa Catarina, Brazil. Preliminary data was used.
Namasudra, Suyel et al. [12]	2023	Pearson correlation: $\rho_{India} \approx 0.99914$ (Overall performance of death cases)	Nonlinear Autoregressive (NAR) Neural Network Time Series (NAR-NNTS) and others. The study was conducted in India. Preliminary data was used (January 2020 to August 2020).
Özen, Figen et al. [13]	2024	Pearson correlation: $\rho_{Turkey} \approx 0.9917$, $R^2 = 0.9834$ (daily deaths)	Random forest regression and machine learning models (i.e., LSTM and ARIMA). The data starts on March 11, 2020 and ends on June 1, 2022 (they do not cover until the end of the pandemic). The study was conducted in Turkey.
Kanchan, Swarna et al. [14]	2024	Pearson correlation: $\rho_{Ada\ County} = 0.669$ (predicted daily deaths)	ANN model. The data cover June 7, 2021, to July 1, 2022 (they do not cover until the end of the pandemic). The study was conducted for the city of Boise (Ada County), the capital of Idaho.

used the extended logistic function in this context [15]; however, the authors did not compare it with an ANN, and did not calculate derivatives or turning points. Similarly, in Mexico, the Gompertz function was used [16], and in Brazil and other countries, the Boltzmann function was used [17]. A total of five studies were included detailing each method's performance metrics. The studies were published between 2021 and 2024. Table **??** summarizes the key characteristics of the included studies.

3 Materials and methods

3.1 Dataset

The database of the total number of deaths used in the current study is publicly available (open access) from the Coronavirus Disease Data Repository of Johns Hopkins University (https://github.com/CSSEGISandData/COVID-19; accessed on 1 January 2024). Complete data for different countries are also available [18].

No data preprocessing was necessary, as there were no missing or outlier data; only the data from the countries under study were selected. Normalization (using the mean and standard deviation) was applied for the artificial neural networks.

The study would be trivial with data from the first days of the pandemic. For example, in Peru and most countries, during the first 20 days, the curve barely begins to grow and can be fitted to an exponential or even linear model. In contrast, a model is required for the complete dataset (from the beginning to the end of the pandemic). Moreover, modeling a single wave can be relatively easy [19].

In this work, no function integration method is required [4], and a program is used to process and estimate F.

3.2 The sigmoidal logistic function

The sigmoidal logistic function can be applied to a time-series [20]. It has different applications, such as representing epidemic curves. It is represented by the following function:

$$g\left(x\right) = \frac{H}{1 + \mathrm{e}^{A-Bx}}, \tag{1}$$

Where x is the number of days since the first case, $g(x)$ is the expected cumulative number of deaths, H is the height of the wave, A and B are constants related to the pandemic, and e is a mathematical constant (i.e., an irrational and transcendental number approximately equal to 2.718281828459045).

In this work, a new version of the logistic function is used, which we call the logistic2 function or, simply, the logistic function:

Table 3: Summarizes the five studies included in the review (sigmoidal functions). It shows the authors, publication year, countries - metrics, key findings, and limitations.

Author	Year	Metrics	Key findings and limitations
Abolmaali, Saina et al. [15]	2021	MSE: US=678.7, India = 631.3, Brazil=731.0, and Russia = 1501.1	Logistic regression and other models were used. They worked with preliminary data (approximately one year). No correlation coefficients were provided. Finally, the model is simple (a single sigmoidal function).
Conde, R. et al. [16]	2021	Pearson correlation: $\rho_{Mexico} \approx 0.9998$	The Gompertz sigmoidal function was used. Preliminary data (one wage or six months) was used during the pandemic. The model is simple and for a single wave.
Vilca, Oliver et al. [4]	2023	Pearson correlation coefficient: ρ_{Peru} = 0.9577994	The Gompertz sigmoidal function was used. It requires inserting a dummy variable q into the model and the database. Data available at that time (up to the third wave) was used. The models: $F(x,q) = \sum_{w=1}^{n} \left(\frac{\prod_{i=1}^{w-1}(i-q)\prod_{j=w+1}^{n}(j-q)}{(n-w)!(-1)^{w-1}(w-1)!} \cdot A_w \cdot e^{-e^{B_w - C_w \cdot x}} \right)$ $F(x,q) = \sum_{w=1}^{n} \left(\frac{2^{w-1}\&q}{2^{w-1}} \cdot A_w \cdot e^{-e^{B_w - C_w \cdot x}} \right)$ Where A_w, B_w, and C_w are constants related to the pandemic.
El Aferni, Ahmed et al. [17]	2023	$\rho_{United\ Kingdom} \approx 0.99912$ $\rho_{Zambia} \approx 0.9991$ $\rho_{Brazil} \approx 0.9995$ $\rho_{South\ Africa} \approx 0.999925$	The sigmoid Boltzmann model was found to be effective in fitting the cumulative number of COVID-19. With data covering up to the fourth wave. The model is: $F(x) = I_{min} + I_{max} \sum_{i=1}^{n} \left(\frac{P_i}{1+10^{(A_i - x)B_i}} \right)$, where P_i, A_i, B_i, I_{min} and I_{max} are constants related to the pandemic.
Vilca, Oliver et al. [5]	2024	Pearson correlation coefficient: ρ_{Peru} = 0.999	The Boltzmann sigmoidal function was used. Performed with complete data. It requires the insertion of a dummy variable q in the model and the database. The models: $F(x,q) = \sum_{w=1}^{n} \left(\frac{\prod_{i=1}^{w-1}(i-q)\prod_{j=w+1}^{n}(j-q)}{(n-w)!(-1)^{w-1}(w-1)!} \frac{I_w}{1+e^{\frac{(Z_w - x)}{D_w}}} \right)$ $F(x,q) = \sum_{w=1}^{n} \left(\frac{2^{w-1}\&q}{2^{w-1}} \frac{I_w}{1+e^{\frac{(Z_w - x)}{D_w}}} \right)$ where I_w, Z_w and D_w are constants related to the pandemic.

Note: $F(x)$ and $F(x,q)$ are the expected cumulative number of deaths, x is the number of days since the first case, and n is the number of waves.

$$g(x) = \frac{H}{1 + 2^{A-Bx}}, \tag{2}$$

where A and B are new pandemic-related constants. Instead of the irrational number e (which cannot be expressed as a fraction m/n, where $m, n \in \mathbb{Z}$), we use the natural number 2 as, in computing, we can efficiently calculate 2^k if $k \in \mathbb{N}$.

The high-order derivatives were calculated by hand and checked using Octave Software (version 9.1.0). This approach improved the factorization and presentation of the functions

3.3 Software and hardware

For all models, the same computer, operating system, and programming language were used:

1. HP 11th Generation Intel(R) Core™ i7-1165G7 2.8 GHz computer. RAM 16 GB.
2. Windows 11 Pro Operating System.
3. The R programming language (version 4.4.1 - free software environment for statistical computing and graphics), and the integrated development environment (IDE) RStudio Desktop (2024.09.0+375 "Cranberry Hibiscus" release).

The algorithm for the logistic model has the following procedures:

1. Load libraries (data structures, nonlinear regression, metrics, graphs, and neural networks).
2. Select country and language.
3. Load data.
4. Estimate the model (providing starting values).
5. Show goodness-of-fit reports (ρ, MAE, RMSE, and p-values).

6. Export the model $F(x)$, the first derivative $F'(x)$, and the second derivative $F''(x)$.

7. Generate the graph (configuring color, size, resolution, and output file, among others).

3.4 Parameter initialization

Parameters of the Logistic Sigmoidal Model: Non-linear regression with numerical algorithms may require initialization parameters. In this case, estimated values were provided. For instance, the approximate height of the sigmoidal function, H, can be obtained by observing the scatter plot (observed data).

Parameters of the ANN Model: The Long Short-Term Memory (LSTM) model was used, considering previous works [10, 11, 13]. It is designed to effectively process and retain information over multiple steps. A single hidden layer is sufficient to simplify the LSTM architecture.

The number of epochs was chosen experimentally (i.e., 500), increasing until an acceptable fit was achieved. Similarly, the ADAM optimizer performed slightly better than RMSProp. The input layer contains one neuron since there is one input (the cumulative number of deaths). Likewise, the output layer contains one neuron as it is a regressor. For the number of neurons in the hidden layer, the following formula was used: $N = \frac{N_s}{\alpha(N_i+N_o)}$, where N_i represents the number of input neurons, N_o the number of output neurons, N_s the number of samples in the training dataset, and α is a scaling factor typically between 2 and 10. With $N_i = 1,\ N_o = 1$, $N_s = 1100$ and $\alpha = 2.75$, the number of neurons is $N = 200$.

3.5 Statistical analyses

For model comparison, three metrics were used. For a sample of n observations y_i ($i = 1, 2, \cdots, n$), and n corresponding model predictions $\hat{y}_i$, where $\bar{y}$ is the mean of the y-values and $\bar{\hat{y}}$ is the mean of the $\hat{y}$-values.

The Pearson product-moment correlation coefficient or Pearson correlation coefficient (ρ for short) is a measure of linear dependence and its direction between two quantitative variables. If there is a positive correlation, when one variable changes, the other variable changes in the same direction; if there is no correlation (zero), there is no relationship between the variables; and, if there is a negative correlation, the opposite of the positive correlation occurs: $\rho = \frac{\sum(y_i-\bar{y})(\hat{y}_i-\bar{\hat{y}})}{\sqrt{\sum(y_i-\bar{y})^2 \sum(\hat{y}_i-\bar{\hat{y}})^2}}$

The root mean square error (RMSE) is the standard deviation of the residuals. Residuals represent the differences between the values that a model predicts and those that are observed: $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$.

The Mean Absolute Error (MAE) calculates the average absolute difference between predicted and actual values: $MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$.

The metrics are complementary and widely used; while ρ measures the strength and direction of the relationship between two datasets, RMSE and MAE measure the difference between them.

A significance test was performed for the Pearson correlation coefficients, and the F-test was used for the sigmoidal models. Concretely, statistics were used to model, analyze, interpret, and present the results.

3.6 Research design

This research is longitudinal, retrospective, and correlational.

This research poses the following question: Do the models fit the cumulative number of deaths? Then, for each model and each country, the hypotheses are:

The logistic sigmoidal model fits the cumulative number of deaths. The F-test is used (at a significance level of 0.01).

The performance scores were used for the ANN. The correlation hypothesis states that a positive and significant correlation exists between the ANN and the accumulated number of deaths (at a significance level of 0.01).

4 Results

This study considers a new model and a simple way to couple sigmoidal functions.

4.1 Case studies

Four countries were intentionally selected; simple random sampling was not used: Italy (Southern and Western Europe), Brazil (South America), the Swiss Confederation (west-central Europe), and Peru (South America).

Table 4 presents specific data for each country (2020): Population, number of deaths, start and end dates of the pandemic (number of days), and the death percentage in relation to the population of each country ($\frac{Deaths}{Population} \cdot 100$). The population data were downloaded from "Population Pyramids of the World" (https://www.populationpyramid.net; accessed on 1 September 2024).

Day one corresponds to the first death. The pandemic began in February 2020 in Italy and in March in Brazil, Switzerland, and Peru. It lasted until 9 March 2023, specifically 1113 days in Italy, 1088 days in Brazil, and 1099 days in Switzerland and Peru.

From Table 4, it can be seen that the country with a considerably higher percentage of deaths, according to the total number of inhabitants, was Peru (0.66%), followed by Italy (0.32%), and ending in a tie between Brazil and Switzerland (0.16%).

These countries differ in location, population, and number of deaths from COVID-19.

Table 4: Data for each country (2020): Population, number of deaths, start and end dates of the pandemic (number of days), and the death percentage in relation to the population of each country.

Country	Population	Number of deaths	Start and end date of the pandemic (number of days)	Percentage of deaths
Italy	59,500,579	188,322	From 21 February 2020 to 9 March 2023 (1113 days)	0.32 %
Brazil	213,196,304	333,188	From 17 March 2020 to 9 March 2023 (1088 days)	0.16 %
Switzerland	8,638,613	14,210	From 06 March 2020 to 9 March 2023 (1099 days)	0.16 %
Peru	33,304,756	219,539	From 06 March 2020 to 9 March 2023 (1099 days)	0.66 %

Table 5: Artificial neural networks adjustment measures.

Country	Pearson Correlation ρ	Coefficient of Determination	RMSE	MAE	p-value for ρ	Confidence interval for ρ	Running time (min)
Italy	0.999715	0.999431	0.0252	0.0162	$< 2.2 \cdot 10^{-16}$	0.9997 , 0.9997	16.698
Brazil	0.999862	0.999724	0.0176	0.013	$< 2.2 \cdot 10^{-16}$	0.9998 , 0.9999	16.239
Switzerland	0.999805	0.99961	0.0226	0.0132	$< 2.2 \cdot 10^{-16}$	0.9998 , 0.9998	16.467
Peru	0.99994	0.999879	0.0132	0.0095	$< 2.2 \cdot 10^{-16}$	0.9999 , 0.9999	16.43
Average	0.999831	0.999661	0.0196	0.013	$< 2.2 \cdot 10^{-16}$		16.458
Std. Dev.	9.4e-05	0.000189	0.0053	0.0028	$< 2.2 \cdot 10^{-16}$		0.188

4.2 Artificial neural networks

In machine learning, a neural network (NN), also called an artificial neural network (ANN), is a model composed of connected units or nodes called neurons [21]. NNs, learn by example, the ANN is set up during a learning process to perform specific tasks, such as identifying patterns and categorizing information [22].

Long short-term memory (LSTM) is a type of recurrent neural network (RNN) architecture designed to address the limitations of traditional RNNs, especially when it comes to learning long-term dependencies. The main innovation of LSTM is the use of gating mechanisms to control the flow of information through the network. This allows the LSTM to maintain and update its internal state over long periods [23, 24].

R provides libraries and frameworks that simplify the process of building and training neural networks, including LSTM networks. The Keras library provides a high-level interface for building and training neural networks. To the authors' knowledge, Python has been used in most previous works.

The LSTM was used with the following characteristics: three layers (1 input neuron, 200 neurons in the hidden layer, and one output neuron), the "adam" optimizer, loss = "mae", epochs = 500, and batch_size = 1 (see section 3.4 for more details).

Table 5 presents the ANN model's goodness-of-fit results by country, as well as the averages and standard deviations. On the one hand, the average RMSE and MAE of the ANNs were quite acceptable (0.0196 and 0.013, respectively, on scaled data; see columns four and five). On the other hand, the Pearson correlation coefficients (ρ) were quite good: Those for Italy, Brazil, Switzerland, and Peru were 0.999715, 0.999862, 0.999805, and 0.99994, respectively (usually varying with each execution).

Moreover, the correlations are statistically significant in the four countries (the p-values are below the significance level of 0.01). In R, p-value $< 2.2e-16$ means that the p-value is less than $2.2 \cdot 10^{-16}$ (scientific notation for 0.00000000000000022). The confidence intervals are presented in column seven.

The correlations (ρ) for the ANN did not differ much because ANNs adapt well to data. However, sigmoidal logistic models require the data to follow a pattern.

For some countries, the data fit better; however, all countries fit pretty well.

The coefficients of determination (third column) indicate that the models obtained are explained or determined by the differences in the days.

In column eight of Table 5, the average empirical execution time per country (measured in minutes) is presented. An experiment consists of measuring the initial time, estimating the model, obtaining the final time, and calculating the difference (the Sys.time() function from R was used). The experiment was repeated forty times [25] in each country, and the average was recorded to avoid bias. The average time is 16.698 minutes in Italy, 16.239 in Brazil, 16.467 in Switzerland, and 16.43 in Peru.

Finally, the source code for the artificial neural networks is presented in Appendix B.

4.3 Calculating the number of operations of sigmoidal functions.

The execution time of a sigmoidal function depends on the type and number of operations. In general, the order of the execution time of the operations, from smallest to largest, is as follows: (1) addition or subtraction, (2) multiplication or

Table 6: Number of operations in sigmoidal functions

Sigmoidal function	Number of power operations	Number of division operations	Number of multiplication operations	Number of addition or subtraction operations
Gompertz function: $g(x) = A \cdot e^{-e^{B-D \cdot x}}$	2	0	2	1
Hyperbolic tangent function: $g(x) = tanh(x) = U - \frac{H}{1+e^{2(x-A)}}$	1	1	1	3
Boltzmann function: $g(x) = \frac{H}{1+e^{(C-x)/D}}$	1	2	0	2
Logistic function: $g(x) = \frac{H}{1+\mathrm{e}^{A-Bx}}$	1	1	1	2

division, and (3) exponentiation with a real number in the exponent; that is, addition/subtraction is calculated faster than the other two, and multiplication/division is generally faster than exponentiation.

Then, for evaluation, the following functions are considered: those based on the Gompertz function (representing the double exponential functions), based on the hyperbolic tangent function (representing the extended functions of the logistic function), the Boltzmann function, and the logistic function. The number of operations was calculated, as detailed in Table 6.

Moreover, the execution times were calculated based on the results from Table 6:

$$\begin{aligned} T_{Go} &= M\log(B-Dx) + Mlog(e^{B-Dx}) + 2M + S \\ T_{Hp} &= M\log(2(x-A)) + D + M + 3S \\ T_{Bz} &= M\log((C-x)/D) + 2D + 2S \\ T_{Lg} &= M\log(A-Bx) + D + M + 2S \end{aligned}$$

Where the execution times for the basic operations are as follows: Exponentiation (a^b) requires $\log b$ multiplications, division is D, multiplication is M, and addition/subtraction is S.

The Gompertz function was found to be the least suitable, as it has two exponential operations, two multiplication operations, and one subtraction operation. Meanwhile, the hyperbolic tangent function has an additional subtraction operation in relation to the logistic function. The Boltzmann function has two divisions (generally, in computing, the division operation costs more than multiplication). Finally, the logistic function was selected because it has the simplest operations and in the minimum quantity, considering the set of basic sigmoidal functions. It could barely be equaled by some extended sigmoidal functions and, therefore, it was considered efficient.

4.4 Coupled models or functions

It has the following characteristics or advantages: First, it does not require any form of integration or union of functions, and second, it does not need to add a dummy variable to the data or to the function as in previous publications [4, 5].

Theorem 1. The coupled model (or coupled function) with n sigmoidal logistic functions is:

$$F(x) = \sum_{i=1}^{n} \frac{H_i}{1+2^{A_i - B_i x}} \tag{3}$$

Where $F(x)$ is the expected cumulative number of deaths, x is the number of days since the first case, n is the number of waves ($1 \leq i \leq n$), H_i is the height of the wave i, and A_i and B_i are constants related to the pandemic.

Furthermore, $x_0 < x_1 < x_2 < \cdots < x_n$ are consecutive real numbers that determine the intervals of the logistic functions. The function $g(x)_i$ is defined on the interval $x \in [x_{i-1}, x_i]$; $\forall x > x_i$, $g(x)_i$ tends to H_i, and $\forall x < x_{i-1}$, $g(x)_i$ tends to 0 (limits of the sigmoidal logistic function).

Therefore $\lim_{x\to-\infty} F(x) = 0$ and $\lim_{x\to\infty} F(x) = \sum_{k=1}^{i} H_k$.□

Proof. Let $x_0 < x_1 < x_2 < \cdots < x_n$ be consecutive real numbers determining the intervals of logistic functions. In general, the function $g(x)_i$ is defined on the interval $x \in [x_{i-1}, x_i]$ (the i-th "wave"); $\forall x > x_i$, $g(x)_i$ tends to H_i, and $\forall x < x_{i-1}$, $g(x)_i$ tends to 0 (right and left limits of the sigmoidal function).

Proof by mathematical induction: Base case $F(x)$ with two functions; if $x < x_0$, then $F(x) \approx 0$ (the left-hand limits are zero); if $x \in [x_0, x_1]$, then $F(x) \approx g(x)_1$; if $x \in [x_1, x_2]$, then $F(x) \approx g(x)_1 + g(x)_2 \approx H_1 + g(x)_2$, and if $x > x_2$, then $F(x) \approx H_1 + H_2$.
Induction hypothesis:
$E(x) \approx \sum_{k=1}^{i-1} \left(\frac{E_k}{1+2^{A_k - B_k x}}\right)$, $\forall x \in [x_0, x_{i-1}]$ where $\lim_{x\to-\infty} E(x) = 0$ and $\lim_{x\to\infty} E(x) = \sum_{k=1}^{i-1} H_k$
Inductive step: By definition of $F(x)$,
$F(x) \approx E(x) + g(x)_i$, $\forall x \in [x_0, x_i]$
$F(x) \approx \sum_{k=1}^{i-1} \left(\frac{H_k}{1+2^{A_k - B_k x}}\right) + g(x)_i$, $\forall x \in [x_0, x_i]$
$F(x) \approx \sum_{k=1}^{i} \left(\frac{H_k}{1+2^{A_k - B_k x}}\right)$, $\forall x \in [x_0, x_i]$.
Right limit: $F(x) \approx \sum_{k=1}^{i} H_k, \forall x > x_i$, by induction hypothesis $\lim_{x\to\infty} E(x) = \sum_{k=1}^{i-1} H_k$ and $\lim_{x\to\infty} g(x)_i =$

H_i.
Left limit: $F(x) \approx 0$, $\forall x < x_0$, by induction hypothesis $\lim_{x\to-\infty} E(x) = 0$ and $\lim_{x\to-\infty} g(x)_i = 0$;
Finally, $F(x)$ is defined by the equation (3). □

Corollary 4.1. Given theorem (1), considering that the sigmoidal logistic functions are bijective, and let $F(x)$, let $x_0 < x_1 < x_2 < \cdots < x_n$ be consecutive real numbers determining the intervals of logistic functions $[x_0, x_1]$, $[x_1, x_2]$, ... , $[x_{n-1}, x_n]$. Then, the respective ordinates $[y_0, y_1]$, $[y_1, y_2]$, ... , $[y_{n-1}, y_n]$ are consecutive, and non-overlapping.□

The construction of $F(x)$ extends to other functions $h(x)$, the only condition is that the limits of $h(x)$ as x tends to $\pm\infty$ are constant.

Theorem 2. The coupled model with n functions, each with constant limits as x tends to $\pm\infty$, is:

$$F(x) = h(x)_1 + h(x)_2 + \cdots + h(x)_n, \tag{4}$$

Where $\lim_{x\to-\infty} h(x) = L$, $\lim_{x\to\infty} h(x) = R$, and L and R are constants.
Therefore $\lim_{x\to-\infty} F(x) = K_1$, $\lim_{x\to\infty} F(x) = K_2$, and K_1 and K_2 are constants.□

The proof is similar to Theorem 1. If two or more functions with total or partial intersection of their intervals, that is, they are defined in the same interval, they are added.

Figure 1 shows the coupling of four functions: Gompertz, Logistic, algebraic, and wave function. Only the first two are properly sigmoidal, the third is an algebraic function with sigmoidal behavior, and the last is the wave function. It is possible to include different functions that meet the conditions explained, for example, the normal distribution function.

The function $y = F(x)$ is described by equation (5):

$$y = \underbrace{2 \cdot e^{-e^{2-x}}}_{Gompertz\ f.} + \underbrace{\frac{2}{1+e^{10-x}}}_{Sigmoidal\ f.} + \underbrace{\frac{(x-20)}{\sqrt{1+(x-20)^2}}}_{Algebraic\ function} \underbrace{-4e^{-\frac{(x-30)^2}{4}} sin\left(\frac{x-30}{4}\right)}_{Wave\ function} + 1 \tag{5}$$

The wave function satisfies the requirement of Theorem 2. Proof: By definition $-1 \leq sin(x) \leq 1$, it follows that, $-\frac{1}{e^{x^2}} \leq \frac{sin(x)}{e^{x^2}} \leq \frac{1}{e^{x^2}}$. Since the limits of both functions are equal, $\lim_{x\to\infty} -\frac{1}{e^{x^2}} = \lim_{x\to\infty} \frac{1}{e^{x^2}} = 0$, it follows that $\lim_{x\to\infty} e^{-x^2} sin(x) = 0$ (squeeze theorem), and in the same way we can obtain $\lim_{x\to-\infty} e^{-x^2} sin(x) = 0$.□

The code to graph $F(x)$ in R is:

```
y = function(x){
 2*e^( -e^(2-x) ) + 2/( 1+e^( 10-x ) ) +
 (x-20)/sqrt( 1+(x-20)^2 ) -
 4*e^(-(x-30)^2/4)*sin((x-30)/4) + 1
}
plot(y, 0, 37, col="green")
```

4.5 The model for COVID-19

First, an efficient sigmoidal function had to be selected. Subsequently, a model for COVID-19 was developed.

The fundamental element of the model is the logistic sigmoidal function, equation (20). Specifically, the model $F(x)$ is a summation of ordered logistic functions, starting with the function $g(x)_1$ for the first wave, $g(x)_2$ for the second wave, and so on up to $g(x)_n$ for the last wave. Then, the model for COVID-19 is a coupled model with sigmoidal functions specified in theorem (1), and is given by equation (6):

$$F(x) = g(x)_1 + g(x)_2 + \cdots + g(x)_n \tag{6}$$

4.6 Empirical execution time

In this section, the time required to estimate the model parameters in each country is calculated.

The experiment was repeated forty times in each country [25], and the average was recorded to avoid bias. Table 7 presents the results; in the case of Italy, the time is higher due to the presence of six waves (one more than the other countries). Subsequently, the mathematical model reported low times.

Table 7: Execution time (in seconds) to calculate the parameters of a logistic model by country

Country	Running time (in seconds)
Italy	0.2968650
Brazil	0.1491925
Switzerland	0.1799175
Peru	0.1215575

4.7 Application of the logistic function to COVID-19

The logistic function $F(x)$ is a set of functions, which are ordered sequentially and do not overlap. Each function corresponds to a wave of COVID-19.

Figure 2 represents the cumulative number of deaths from COVID-19. The dataset (observed data) is represented in gray, and the logistic function $F(x)$ in blue, and the turning points in red. The observed data are very close to the curve. Remark: While the sigmoidal shape of the later waves cannot be distinguished due to the scale (thousands on the ordinate axis), it is visible if these (one or two) waves are plotted exclusively.

In all cases, the first two or three waves were relatively larger and lasted longer (their length on the x-axis is the duration in days); in addition, they had a greater number

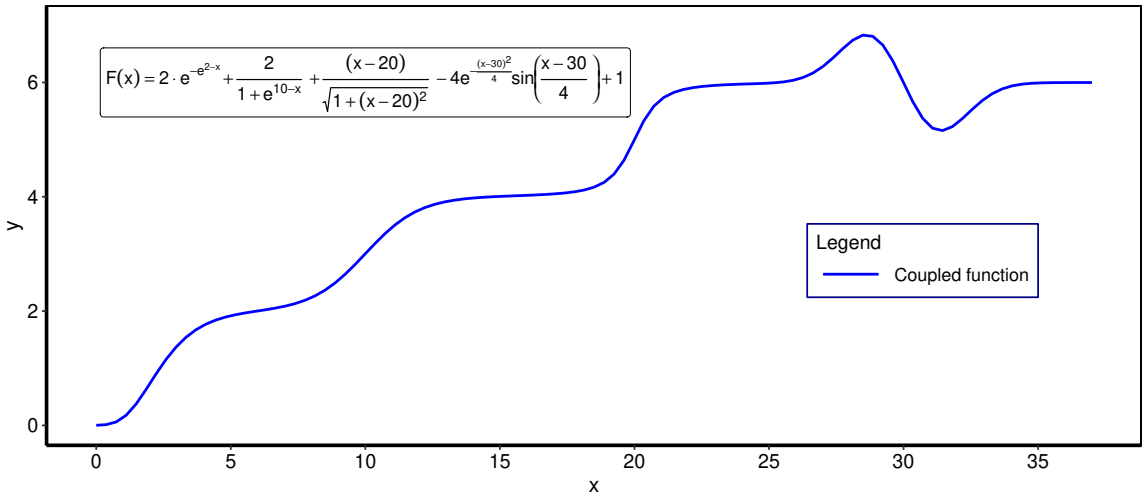


Figure 1: The coupled function $F(x)$ is made up of four functions: Gompertz, Logistic, algebraic, and wave. Only the first two are properly sigmoidal, the third is an algebraic function with sigmoidal behavior, and the last is the wave function.

Table 8: Logistic sigmoidal function adjustment measures

Country	**Pearson Correlation** ρ	**Coefficient of Determination**	**RMSE**	**MAE**	**p-value for** ρ	**Confidence interval for** ρ
Italy	0.9999230	0.999846	0.7164719	0.5779384	$< 2.2 \cdot 10^{-16}$	0.9999150 , 1
Brazil	0.9999283	0.9998566	3.013632	1.99136	$< 2.2 \cdot 10^{-16}$	0.9999208 , 1
Switzerland	0.9999178	0.9998356	0.061266	0.04001958	$< 2.2 \cdot 10^{-16}$	0.9999092 , 1
Peru	0.9999050	0.99981	1.022324	0.6213603	$< 2.2 \cdot 10^{-16}$	0.9998951 , 1
Average	0.9999185	0.9998371	1.203423	0.80767	$< 2.2 \cdot 10^{-16}$	
Std. Dev.	0.00001	0.00002	1.271652	0.832245	$< 2.2 \cdot 10^{-16}$	

of victims (represented by the height on the ordinate axis). Meanwhile, subsequent waves were considerably shorter in duration and had fewer victims. In each country, the last wave ends almost flat (the slope tends to zero); under these circumstances, the pandemic ended. The second wave in Brazil was the largest (it can be observed that it led to more than 400,000 deaths).

The Pearson product-moment correlation coefficients (ρ) were greater than 0.9999 (see the second column in Table 8), indicating very strong positive associations; that is, a high degree of association between the observed data and the function. Moreover, the correlations are statistically significant in the four countries. The p-values ($< 2.2 \cdot 10^{-16}$) are below the significance level of 0.01, and the confidence intervals are presented in column seven.

Brazil and Italy received the best adjustment, while Peru received the worst, but the differences were minimal.

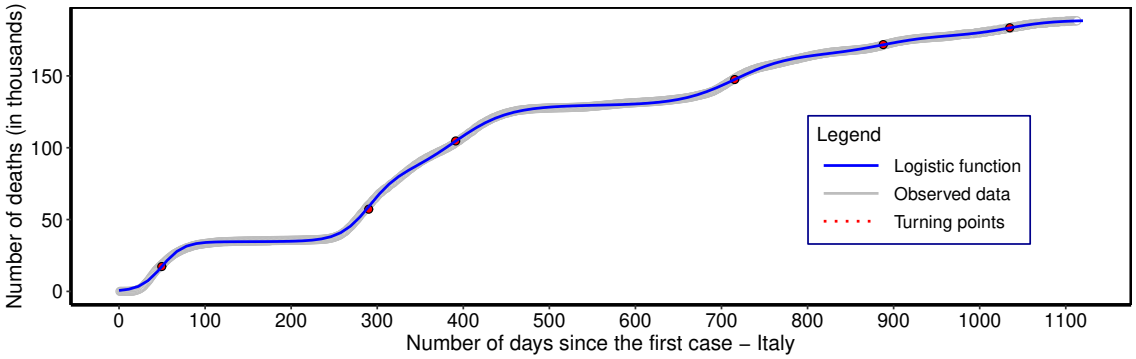
The coefficients of determination (third column) indicate that the models obtained are explained or determined by the differences in the days.

The RMSE is 0.72 in Italy, 3.01 in Brazil, 0.06 in Switzerland, and 1.02 in Peru, and the MAE is 0.58, 1.99, 0.04, and 0.62, respectively. The metrics indicate a low average distance between the model's predicted and dataset values.

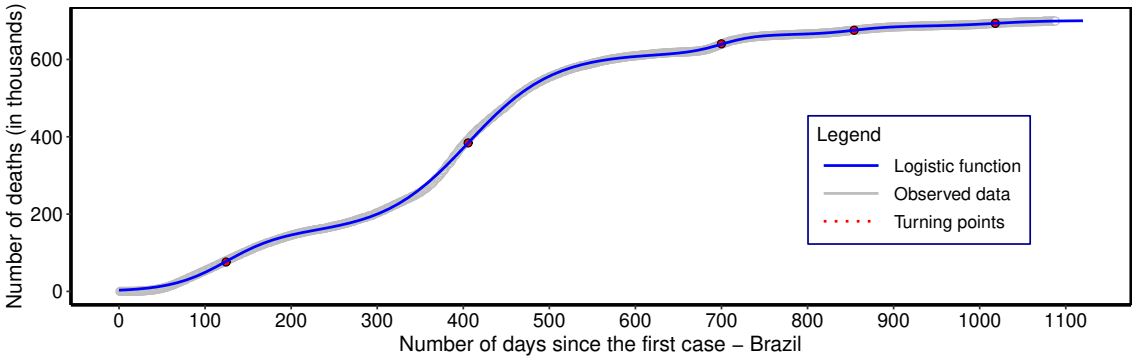
F-test: The regression model is statistically significant in the four countries; therefore, the regression model's utility permits the calculation of various estimates and predictions (with p-value $= 2.2 \times 10^{-16}$, less than the significance level of 0.01).

Equations (7), (8), (9), and (10) represent the functions $F(x)$:

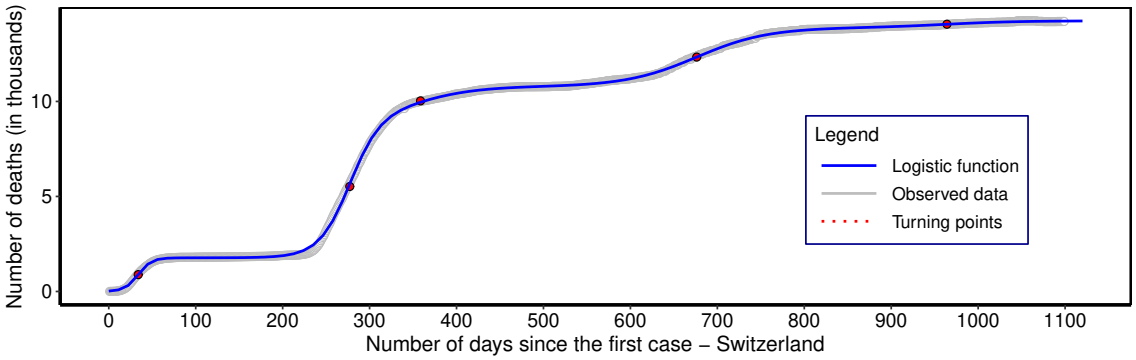
$$\underbrace{F(x)}_{Italy} = \frac{34.6}{1+2^{5.77-0.116*x}} + \frac{45.22}{1+2^{25.1-0.087*x}} + \frac{49.77}{1+2^{18.45-0.047*x}} + \frac{35.74}{1+2^{32.57-0.046*x}} + \frac{12.9}{1+2^{43.62-0.049*x}} + \frac{10.51}{1+2^{58.91-0.057*x}} \quad (7)$$



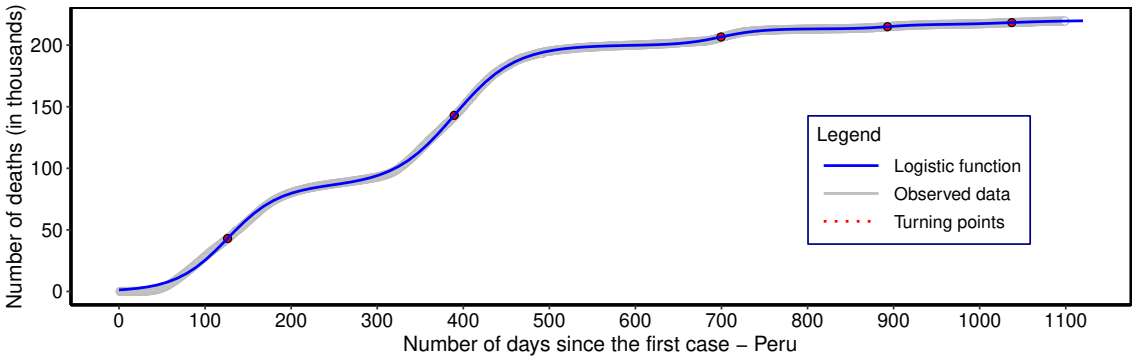
(a) Case: Italy with six waves.



(b) Case: Brazil with five waves.



(c) Case: Switzerland with five waves.



(d) Case: Peru with five waves.

Figure 2: Logistic function of the cumulative number of deaths from COVID-19 (in thousands) by day. The observed data are represented in gray, the logistic function $F(x)$ in blue, and the turning points in red.

$$\underbrace{F(x)}_{Brazil} = \frac{152.06}{1+2^{5.62-0.045*x}} + \frac{464.04}{1+2^{11.9-0.029*x}} + \frac{48.72}{1+2^{51.47-0.074*x}} + \frac{21.45}{1+2^{56.44-0.066*x}} + \frac{14.2}{1+2^{50.35-0.049*x}} \quad (8)$$

$$\underbrace{F(x)}_{\text{Switzerland}} = \frac{1.77}{1+2^{6.49-0.192*x}} + \frac{7.51}{1+2^{22.68-0.082*x}} + \frac{1.5}{1+2^{14.71-0.041*x}} + \frac{3.1}{1+2^{24.01-0.036*x}} + \frac{0.36}{1+2^{33.8-0.035*x}} \quad (9)$$

$$\underbrace{F(x)}_{Peru} = \frac{85.94}{1+2^{6.07-0.048*x}} + \frac{114.1}{1+2^{15.96-0.041*x}} + \frac{13.22}{1+2^{44.35-0.063*x}} + \frac{3.4}{1+2^{73.16-0.082*x}} + \frac{3.33}{1+2^{46.8-0.045*x}} \quad (10)$$

4.8 First derivative

The first derivative of the logistic function $g(x)$ or equation (20) is given in equation (11):

$$g'(x) = \frac{B\,H\,\ln(2)\,2^{A-B\,x}}{\left(1+2^{A-B\,x}\right)^2}, \quad (11)$$

Where x is the number of days since the first case, $g'(x)$ is the death rate of the cumulative number of deaths, H is the height of the wave, A and B are the pandemic constants, and ln is the natural logarithm of base e; that is, $\ln(x) = \log_e(x)$.

The first derivative is helpful as it represents the speed of spread. It is also used to calculate the second derivative.

4.9 Second derivative

Calculating the second derivative is necessary to obtain the turning point

The equation (12) is the second derivative of the logistic function $g(x)$; that is, the derivative of the equation (11):

$$g''(x) = -\frac{B^2\,H\ln(2)^2\,2^{A-Bx}\left(1-2^{A-Bx}\right)}{\left(1+2^{A-Bx}\right)^3}, \quad (12)$$

where x is the number of days since the first, $g''(x)$ is the second derivative (also called the acceleration) of the cumulative number of deaths, H is the height of the wave, and A and B are the pandemic constants.

4.10 Third derivative

The third derivative $F'''(x)$ is:

$$\frac{B^3\,H\ln(2)^3\,2^{A-Bx}\left(1-4\cdot 2^{A-Bx}+2^{2(A-Bx)}\right)}{\left(1+2^{A-Bx}\right)^4}, \quad (13)$$

Where x is the number of days since the first case, the third derivative $g'''(x)$ is the rate of change of acceleration over time (also known as jerk, it is aptly named because a large jerk means a sudden change in acceleration, which causes an abrupt movement), and H is the height of the wave. Finally, A and B are the pandemic constants.

4.11 Turning point

A point c on a curve $y = g(x)$ is called a turning point (or inflection point) if g is continuous and the curve changes from concave to convex (or vice versa) at c.

To obtain the turning point, it is first necessary to find the stationary point of $g'(x)$; that is, to solve the following equation $g''(x) = 0$. Then, the solution is given by $x_0 = \frac{A}{B}$ and $y_0 = \frac{H}{2}$.

Finally, $g'''(x_0)$ is evaluated $F'''(x_0) = -0.0416\,B^3\,H$. The result is different from zero (with $B \neq 0$ and $H \neq 0$), which indicates that $(\frac{A}{B}, \frac{H}{2})$ is a turning point; more calculation details can be found in Appendix A.4.

4.12 Meaning of the turning point

The turning or inflection points are important because they represent moments when the trend of the observed variable changes (mathematically, this is the point where it changes from concave up to concave down). In the context of the pandemic, they indicate a change in the spread of the virus, which can have direct implications for public health policies and governmental interventions. For example, they could be associated with easing mitigation measures (quarantines or social distancing) or healthcare capacity.

4.13 Applying high-order derivatives

We continued by deriving the functions $F(x)$. The results are equations (14), (15), (16), and (17), which represent the first derivatives for Italy, Brazil, Switzerland, and Peru, respectively, using equation (11).

$$\underbrace{F'(x)}_{Italy} = \frac{2.79 \cdot 2^{5.77-0.116*x}}{(1+2^{5.77-0.116*x})^2} + \frac{2.71 \cdot 2^{25.1-0.087*x}}{(1+2^{25.1-0.087*x})^2} + \frac{1.63 \cdot 2^{18.45-0.047*x}}{(1+2^{18.45-0.047*x})^2} + \frac{1.13 \cdot 2^{32.57-0.046*x}}{(1+2^{32.57-0.046*x})^2} + \frac{0.44 \cdot 2^{43.62-0.049*x}}{(1+2^{43.62-0.049*x})^2} + \frac{0.41 \cdot 2^{58.91-0.057*x}}{(1+2^{58.91-0.057*x})^2} \quad (14)$$

$$\underbrace{F'(x)}_{Brazil} = \frac{4.76 \cdot 2^{5.62-0.045*x}}{(1+2^{5.62-0.045*x})^2} + \frac{9.44 \cdot 2^{11.9-0.029*x}}{(1+2^{11.9-0.029*x})^2} + \frac{2.48 \cdot 2^{51.47-0.074*x}}{(1+2^{51.47-0.074*x})^2} + \frac{0.98 \cdot 2^{56.44-0.066*x}}{(1+2^{56.44-0.066*x})^2} + \frac{0.49 \cdot 2^{50.35-0.049*x}}{(1+2^{50.35-0.049*x})^2} \quad (15)$$

$$\underbrace{F'(x)}_{\text{Switzerland}} = \frac{0.24 \cdot 2^{6.49-0.192*x}}{(1+2^{6.49-0.192*x})^2} + \frac{0.43 \cdot 2^{22.68-0.082*x}}{(1+2^{22.68-0.082*x})^2} + \frac{0.04 \cdot 2^{14.71-0.041*x}}{(1+2^{14.71-0.041*x})^2} + \frac{0.08 \cdot 2^{24.01-0.036*x}}{(1+2^{24.01-0.036*x})^2} + \frac{0.01 \cdot 2^{33.8-0.035*x}}{(1+2^{33.8-0.035*x})^2} \quad (16)$$

$$\underbrace{F'(x)}_{Peru} = \frac{2.87 \cdot 2^{6.07-0.048*x}}{(1+2^{6.07-0.048*x})^2} + \frac{3.24 \cdot 2^{15.96-0.041*x}}{(1+2^{15.96-0.041*x})^2} + \frac{0.58 \cdot 2^{44.35-0.063*x}}{(1+2^{44.35-0.063*x})^2} + \frac{0.19 \cdot 2^{73.16-0.082*x}}{(1+2^{73.16-0.082*x})^2} + \frac{0.1 \cdot 2^{46.8-0.045*x}}{(1+2^{46.8-0.045*x})^2} \quad (17)$$

The equation (12) was used to calculate the second derivative of the function $F(x)$.

Finally, the turning points are detailed in Table 9.

4.14 Interpretation

The first derivative of the function $F(x)$ allows us to describe the behavior of the rate or speed of the number of deaths. It provides an elegant and analytically based justification for the term "wave."

Figure 3 shows the graphs of the first $F(x)$ derivatives for each country as blue curves. The turning points of function $F(x)$ are projected onto $F'(x)$ and displayed with red dashed lines. For comparison between waves or countries, both axes have the same scale.

Definition 4.1 (x)**.** The value of the abscissa x is the number of days elapsed since the first case or cases. Then, the range (i.e., the number of days from an initial value x_i to another final value x_j, with $x_i \leq x_j$) is the duration or the number of days ($x_j - x_i + 1$). In a complete wave, the range is its duration in days. The definition extends to higher-order derivatives and their respective graphs.

Definition 4.2 ($F'(x)$)**.** The value of the ordinate $F'(x)$ is the death rate or the change of deaths at x (i.e., on the day x, if $x \in \mathbb{N}$). It is also the velocity of the cumulative number of deaths, indicates the rate at which the number of deaths changes with respect to days. Moreover, if this value is higher than that on another day, it can be immediately understood as indicating higher lethality.

Definition 4.3 ($F''(x)$)**.** The value of the ordinate $F''(x)$ is the change in death rate at x. It is also the acceleration of the cumulative number of deaths at x (i.e., on the day x, if $x \in \mathbb{N}$). The second derivative, with respect to days, gives the acceleration, which describes how the velocity of "the cumulative number of deaths" changes over days.

The first waves in Italy and Switzerland reached a lower death rate and shorter duration, which may be for different reasons, including strict social isolation measures and population differences. In Peru and Brazil, it lasted more than 200 days and had fatality rates below and above one, respectively. Using the proposed model, it was possible to characterize the waves of COVID-19, and it is expected that it can be extended to other countries (which also present similar patterns, according to our pilot study).

Figure 4 represents the graphs of the second derivative of $F(x)$ (for Italy and Brazil only) as blue curves, where the turning points of function F(x) are projected onto F'(x) and displayed with red dashed lines. It can be understood as the acceleration of the cumulative number of deaths due to COVID-19.

Finally, a function $F(x)$ is concave upward in the intervals where its first derivative increases, that is, when the second derivative is positive. Similarly, F(x) is concave downward in the intervals where the second derivative is negative. The second derivatives $F''(x)$ are given by equations (18), and (19).

Table 9: Turning points by country

Wave	Brazil	Italy	Switzerland	Peru
1	124.38	49.65	33.87	126.08
2	405.60	290.09	277.20	389.42
3	699.89	391.13	358.53	699.39
4	854.16	715.31	676.27	892.89
5	1018.16	887.93	964.09	1037.25
6		1034.91		

$$\underbrace{F''(x)}_{Italy} = \frac{0.22 \cdot 2^{5.77-0.116*x}(1-2^{5.77-0.116*x})}{(1+2^{5.77-0.116*x})^3} + \frac{0.16 \cdot 2^{25.1-0.087*x}(1-2^{25.1-0.087*x})}{(1+2^{25.1-0.087*x})^3} + \frac{0.05 \cdot 2^{18.45-0.047*x}(1-2^{18.45-0.047*x})}{(1+2^{18.45-0.047*x})^3} + \frac{0.04 \cdot 2^{32.57-0.046*x}(1-2^{32.57-0.046*x})}{(1+2^{32.57-0.046*x})^3} + \frac{0.01 \cdot 2^{43.62-0.049*x}(1-2^{43.62-0.049*x})}{(1+2^{43.62-0.049*x})^3} + \frac{0.02 \cdot 2^{58.91-0.057*x}(1-2^{58.91-0.057*x})}{(1+2^{58.91-0.057*x})^3} \quad (18)$$

$$\underbrace{F''(x)}_{Brazil} = \frac{0.15 \cdot 2^{5.62-0.045*x}(1-2^{5.62-0.045*x})}{(1+2^{5.62-0.045*x})^3} + \frac{0.19 \cdot 2^{11.9-0.029*x}(1-2^{11.9-0.029*x})}{(1+2^{11.9-0.029*x})^3} + \frac{0.13 \cdot 2^{51.47-0.074*x}(1-2^{51.47-0.074*x})}{(1+2^{51.47-0.074*x})^3} + \frac{0.05 \cdot 2^{56.44-0.066*x}(1-2^{56.44-0.066*x})}{(1+2^{56.44-0.066*x})^3} + \frac{0.02 \cdot 2^{50.35-0.049*x}(1-2^{50.35-0.049*x})}{(1+2^{50.35-0.049*x})^3} \quad (19)$$

5 Discussion

The calculation time of a sigmoid function $g(x)$, depends on the number and type of operations required (in computing, exponentiation is more expensive than multiplication/-division, and these, in turn, are more expensive than addition/subtraction). The logistic sigmoidal function requires one exponentiation operation, one multiplication, one division, and two additions/subtractions. Other functions require an equal or greater number of operations and/or have more costly operations:

- The Boltzmann function has the same number and type of operations (one exponentiation, two divisions, and two additions/subtractions). If division costs more than multiplication, the model would be disadvantaged.
- The hyperbolic tangent function requires six operations (one additional subtraction operation).
- The Gompertz function requires five operations (one additional exponentiation operation).

The minimum number of operations required is five, with at most one exponentiation operation and two multiplication/division operations. Thus, the logistic function is efficient in relation to the set of functions (explained in Table 6).

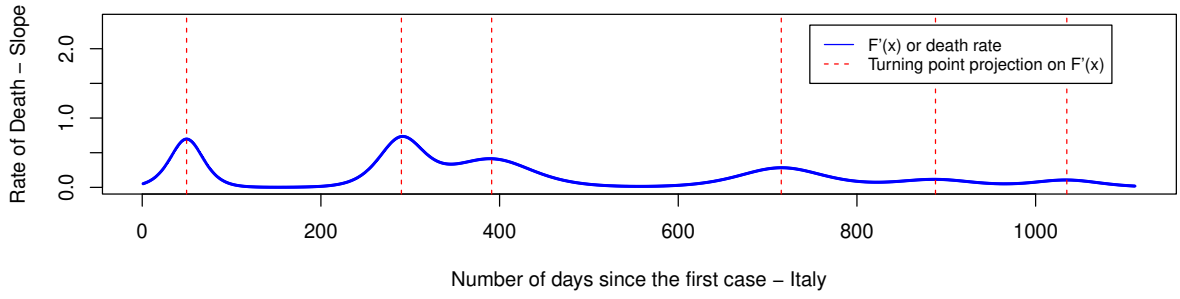
The coupled mode $F(x)$ consists of a summation of logistic functions. It is defined by equation (3): $F(x) = \sum_{i=1}^{n} \frac{H_i}{1+2^{A_i - B_i x}}$, and has no added elements, although previous works do contain them (described in Table **??**):

- The first work by Vilca Oliver et al. [4] presents models $F(x,q) = \sum_{w=1}^{n} \left(\frac{\prod_{i=1}^{w-1}(i-q)\prod_{j=w+1}^{n}(j-q)}{(n-w)!(-1)^{w-1}(w-1)!} \cdot A_w \cdot e^{-e^{B_w - C_w \cdot x}} \right)$ and $F(x,q) = \sum_{w=1}^{n} \left(\frac{2^{w-1}\&q}{2^{w-1}} \cdot A_w \cdot e^{-e^{B_w - C_w \cdot x}} \right)$ which include an additional variable, q, among other details.
- The model by El Aferni, Ahmed et al. [17] $F(x) = I_{min} + I_{max} \sum_{i=1}^{n} \left(\frac{P_i}{1+10^{(A_i - x)B_i}} \right)$ includes two additional components (I_{min} and I_{max}).
- In the second work by Vilca Oliver et al. [5], en $F(x,q) = \sum_{w=1}^{n} \left(\frac{\prod_{i=1}^{w-1}(i-q)\prod_{j=w+1}^{n}(j-q)}{(n-w)!(-1)^{w-1}(w-1)!} \frac{I_w}{1+e^{\frac{(Z_w - x)}{D_w}}} \right)$ y $F(x,q) = \sum_{w=1}^{n} \left(\frac{2^{w-1}\&q}{2^{w-1}} \frac{I_w}{1+e^{\frac{(Z_w - x)}{D_w}}} \right)$ contain additional elements such as the variable q and others.

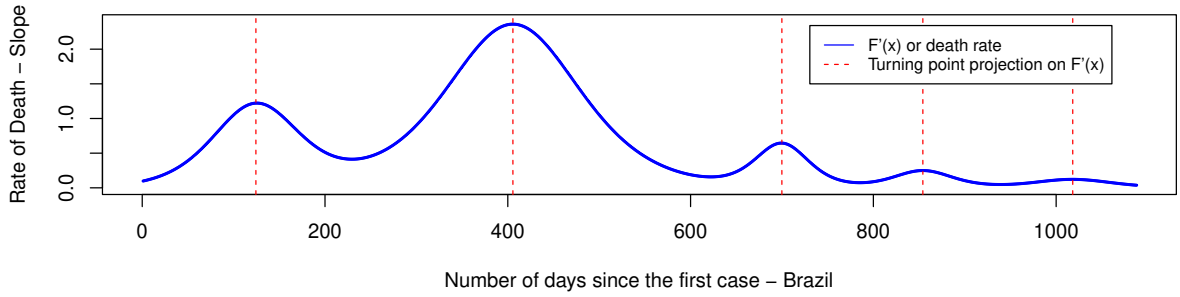
Thus, the presented model has a more straightforward structure than its predecessors

On the other hand, the average Pearson correlation coefficient for the logistic sigmoidal models is $\rho_{Logistic} = 0.99992$, and the standard deviation is 0.00001, which explains that they statistically approximate each other.

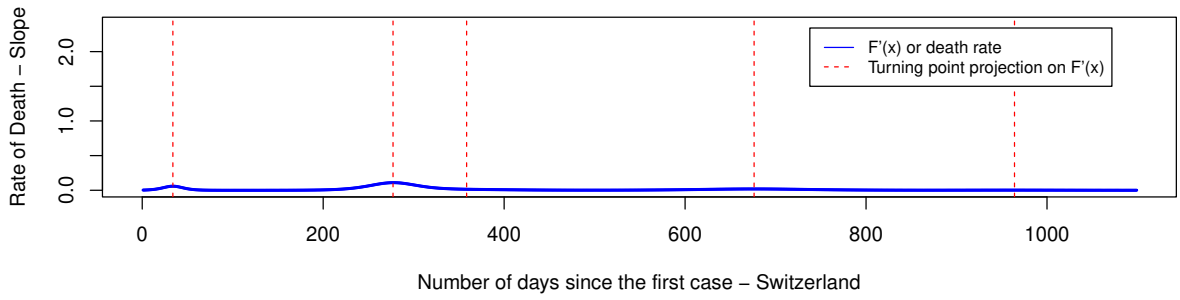
The coefficients are higher than those in previous works (sigmoidal mathematical models presented in Table **??**, particularly in Peru and Brazil). They are also slightly higher than those for models with ANNs ($\rho_{ANNs} = 0.99991$).



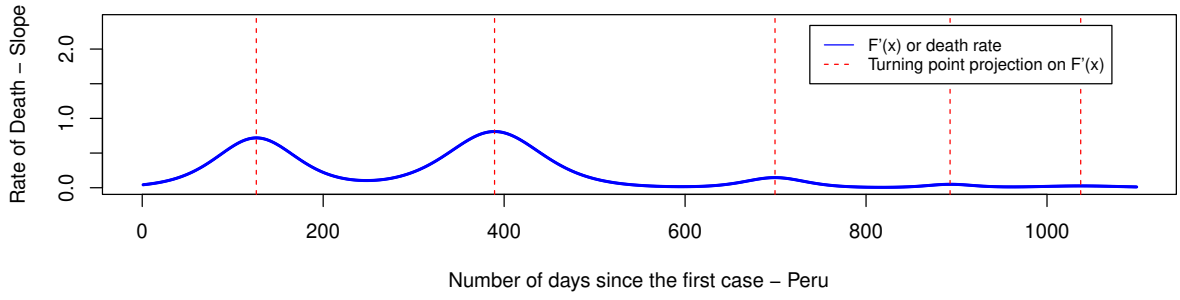
(a) Case: Italy has six waves.



(b) Case: Brazil has five waves, the first and second, with high mortality rates.



(c) Case: Switzerland has five waves and low death rates.

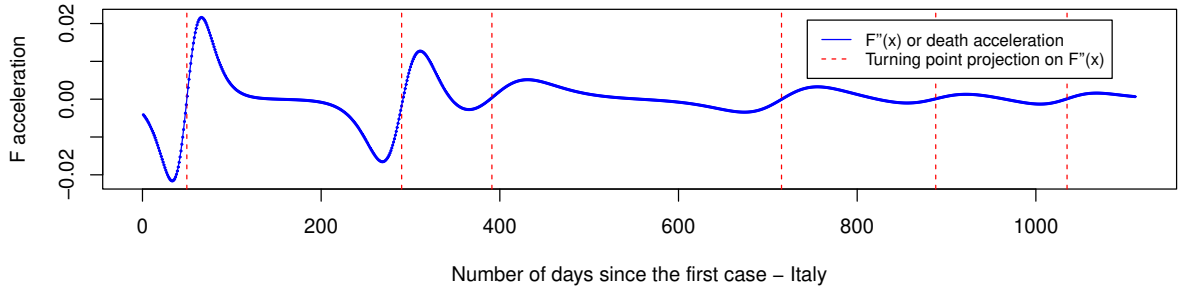


(d) Case: Peru has five waves.

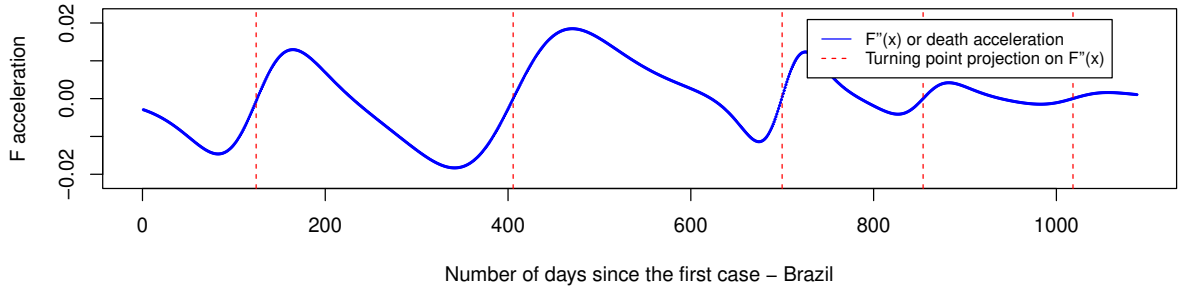
Figure 3: Death rate ($F'(x)$, first derivative of the COVID-19 logistic function of cumulative COVID-19 deaths) in blue, and the vertical projection of the turning points on $F'(x)$ (red dashed line).

Similarly, the Pearson correlation coefficients of the ANN models are greater than or equal to those in previous ANN works (exposed in Table 2).

Regarding the empirical execution time, the sigmoidal model and the ANN are different concepts (mathematical model vs. deep learning). The former requires fewer operations, while the latter includes a more complex architecture (i.e., layers, neurons, and matrix products). Therefore, it is understandable that the sigmoidal models are faster (less than a third of a second; see Table 7).



(a) Case: Italy with six waves.



(b) Case: Brazil with five waves.

Figure 4: Death acceleration ($F''(x)$, second derivative of the logistic function of cumulative COVID-19 deaths) in blue, and the projections of the turning points on $F''(x)$ as red vertical dashed lines (Italy and Brazil).

6 Conclusion

In almost all countries, particularly in the study countries (i.e., Italy, Brazil, Switzerland, and Peru), the COVID-19 waves presented a sigmoidal behavior. Therefore, the proposed model obtained high Pearson correlation coefficients.

The logistic function is efficient, as analyzed by the type and quantity of operations. A new version of the logistic function $g(x)$ was obtained and used, without adding operations.

The main contribution of this work is the COVID-19 model that can be used on data that are not necessarily epidemiological and with any function $h(x)$ with constant left and right asymptotes (that is, $\lim_{x\to-\infty} h(x) = L$, and $\lim_{x\to\infty} h(x) = R$, where L and R are constants, proven by mathematical induction).

The first three derivatives of the sigmoidal logistic function were calculated, allowing for a precise and elegant analysis. Moreover, the turning points (i.e., the point at which the concavity of the function changes) were obtained. The derivatives allowed the waves of COVID-19 to be described, analyzed, and compared, as well as providing justification for the well-known term "waves."

Highly representative and expressive sigmoidal logistic functions were obtained, which stand out for their efficiency in running time, far surpassing that of the tested artificial neural network. The function obtained a good fit, as the waves presented sigmoidal behavior which is typical of epidemics.

The ANN can adapt to any curve without the requirement that it presents a specific pattern. Therefore, such a model would be preferable if atypical behaviors were present in the observed data but at the expense of additional time. Also, it is suitable for large data sets and high dimensionality in the input variables.

Finally, neither model is better, and both have advantages and disadvantages. Both are good for modeling epidemics, mainly when the data follow a sigmoidal pattern.

Four real cases were studied, and models and methodologies were developed. Therefore, the procedures can be applied in other countries and to different datasets beyond COVID-19. Finally, they may be helpful for forecasting and modeling in the case of future pandemics or other practical implications.

Funding statement

This research was funded by Universidad Nacional del Altiplano (UNA), IPN SIP 20231387 project (RR. N. 1589-2023-R-UNA)

Conflict of interest

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Acknowledgement

To the administrative staff and the Vice-Rectorate of Research of the Universidad Nacional del Altiplano for their combined efforts and support.

References

[1] S. Kumar, "Monitoring Novel Corona Virus (COVID-19) Infections in India by Cluster Analysis," *Annals of Data Science*, vol. 7, no. 3, pp. 417–425, sep 2020. [Online]. Available: https://doi.org/10.1007/s40745-020-00289-7

[2] M. A. Alsalem, A. H. Alamoodi, O. S. Albahri, K. A. Dawood, R. T. Mohammed, A. Alnoor, A. A. Zaidan, A. S. Albahri, B. B. Zaidan, F. M. Jumaah, and J. R. Al-Obaidi, "Multi-criteria decision-making for coronavirus disease 2019 applications: a theoretical analysis review," *Artificial Intelligence Review*, vol. 55, no. 6, pp. 4979–5062, aug 2022. [Online]. Available: https://doi.org/10.1007/s10462-021-10124-x

[3] N. Pearce, D. A. Lawlor, and E. B. Brickley, "Comparisons between countries are essential for the control of COVID-19," *International Journal of Epidemiology*, vol. 49, no. 4, pp. 1059–1062, jun 2020, _eprint: https://academic.oup.com/ije/article-pdf/49/4/1059/34275473/dyaa108.pdf. [Online]. Available: https://doi.org/10.1093/ije/dyaa108

[4] O. A. Vilca-Huayta and U. Y. Tito, "Efficient function integration and a case study with gompertz functions for covid-19 waves," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 8, 2022. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2022.0130863

[5] O. A. Vilca Huayta, A. C. Jimenez Chura, C. B. Sosa Maydana, and A. J. Martínez García, "Analysis of the epidemic curve of the waves of covid-19 using integration of functions and neural networks in peru," *Informatics*, vol. 11, no. 2, 2024. [Online]. Available: https://www.mdpi.com/2227-9709/11/2/40

[6] Z. A. Oraibi and S. Albasri, "Efficient COVID-19 Prediction by Merging Various Deep Learning Architectures," *Informatica*, vol. 48, no. 5, Feb. 2024, number: 5. [Online]. Available: https://www.informatica.si/index.php/informatica/article/view/5424

[7] V. Janko, G. Slapničar, E. Dovgan, N. Reščič, T. Kolenik, M. Gjoreski, M. Smerkol, M. Gams, and M. Luštrek, "Machine learning for analyzing non-countermeasure factors affecting early spread of covid-19," *International Journal of Environmental Research and Public Health*, vol. 18, no. 13, 2021. [Online]. Available: https://www.mdpi.com/1660-4601/18/13/6750

[8] D. Cherifi, A. Djaber, M.-E. Guedouar, A. Feghoul, Z. Z. Chelbi, and A. A. Ouakli, "Covid-19 Detecting in Computed Tomography Lungs Images using Machine and transfer Learning," *Informatica*, vol. 47, no. 8, Sep. 2023. [Online]. Available: https://www.informatica.si/index.php/informatica/article/view/4258

[9] R. Sujath, J. M. Chatterjee, and A. E. Hassanien, "A machine learning forecasting model for COVID-19 pandemic in india," *Stochastic Environmental Research and Risk Assessment*, vol. 34, no. 7, pp. 959–972, 2020. [Online]. Available: https://doi.org/10.1007/s00477-020-01827-8

[10] S. Shastri, K. Singh, S. Kumar, P. Kour, and V. Mansotra, "Time series forecasting of covid-19 using deep learning models: India-usa comparative case study," *Chaos, Solitons and Fractals*, vol. 140, p. 110227, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960077920306238

[11] F. Fernandes, S. F. Stefenon, L. O. Seman, A. Nied, F. C. S. Ferreira, M. C. M. Subtil, A. C. R. Klaar, and V. R. Q. Leithardt, "Long short-term memory stacking model to predict the number of cases and deaths caused by COVID-19," *Journal of Intelligent & Fuzzy Systems*, vol. 42, no. 6, pp. 6221–6234, 2022, publisher: IOS Press.

[12] S. Namasudra, S. Dhamodharavadhani, and R. Rathipriya, "Nonlinear Neural Network Based Forecasting Model for Predicting COVID-19 Cases," *Neural Processing Letters*, vol. 55, no. 1, pp. 171–191, 2023.

[13] F. Özen, "Random forest regression for prediction of Covid-19 daily cases and deaths in Turkey," *Heliyon*, vol. 10, no. 4, p. e25746, 2 2024, publisher: Elsevier. [Online]. Available: https://doi.org/10.1016/j.heliyon.2024.e25746

[14] S. Kanchan, E. Ogden, M. Kesheri, A. Skinner, E. Miliken, D. Lyman, J. Armstrong, L. Sciglitano, and G. Hampikian, "COVID-19 hospitalizations and deaths predicted by SARS-CoV-2 levels in Boise, Idaho wastewater," *Science of The Total Environment*, vol. 907, p. 167742, Jan. 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0048969723063696

[15] S. Abolmaali and S. Shirzaei, "A comparative study of SIR Model, Linear Regression, Logistic Function and ARIMA Model for forecasting COVID-19 cases," *AIMS Public Health*, vol. 8, no. 4, pp. 598–613, aug 2021. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8568588/

[16] R. A. Conde-Gutiérrez, D. Colorado, and S. L. Hernández-Bautista, "Comparison of an artificial

neural network and Gompertz model for predicting the dynamics of deaths from COVID-19 in México," *Nonlinear Dynamics*, vol. 104, no. 4, pp. 4655–4669, Jun. 2021. [Online]. Available: https://doi.org/10.1007/s11071-021-06471-7

[17] A. El Aferni, M. Guettari, and A. Hamdouni, "COVID-19 multiwaves as multiphase percolation: a general N-sigmoidal equation to model the spread," *The European Physical Journal Plus*, vol. 138, no. 5, p. 393, may 2023. [Online]. Available: https://doi.org/10.1140/epjp/s13360-023-04014-0

[18] J. H. University, "Github - cssegisanddata/covid-19: Novel coronavirus (covid-19) cases, provided by jhu csse." [Online]. Available: https://github.com/CSSEGISandData/COVID-19

[19] M. Haouari and M. Mhiri, "A particle swarm optimization approach for predicting the number of COVID-19 deaths," *Scientific Reports*, vol. 11, no. 1, p. 16587, Aug. 2021. [Online]. Available: https://www.nature.com/articles/s41598-021-96057-5

[20] M. Viková and M. Vik, "Transition Temperature of Color Change in Thermochromic Systems and Its Description Using Sigmoidal Models," *Materials*, vol. 16, no. 23, p. 7478, jan 2023, number: 23 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1996-1944/16/23/7478

[21] L. Mei, "Model Construction of Higher Education Quality Assurance System Based on Fuzzy Neural Network," *Informatica*, vol. 48, no. 10, Jun. 2024, number: 10. [Online]. Available: https://www.informatica.si/index.php/informatica/article/view/5676

[22] S. F. Ardabili, A. Mosavi, P. Ghamisi, F. Ferdinand, A. R. Varkonyi-Koczy, U. Reuter, T. Rabczuk, and P. M. Atkinson, "Covid-19 outbreak prediction with machine learning," *Algorithms*, vol. 13, no. 10, 2020. [Online]. Available: https://www.mdpi.com/1999-4893/13/10/249

[23] N. P. Dharani and P. Bojja, "Analysis and Prediction of COVID-19 by using Recurrent LSTM Neural Network Model in Machine Learning," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 13, no. 5, 2022. [Online]. Available: https://thesai.org/Publications/ViewPaper?Volume=13&Issue=5&Code=IJACSA&SerialNo=21

[24] M. H. Oukhouya, N. Angour, N. Aboutabit, and I. Hafidi, "Comparative Analysis of ARDL, LSTM, and XGBoost Models For Forecasting The Moroccan Stock Market During The COVID-19 Pandemic," *Informatica*, vol. 49, no. 14, Mar. 2025. [Online]. Available: https://www.informatica.si/index.php/informatica/article/view/5751

[25] O.-A. Vilca and M.-C. Rivara, "Study on the Average Size of the Longest-Edge Propagation Path for Triangulations," Feb. 2025, pp. 368–375. [Online]. Available: https://www.scitepress.org/Link.aspx?doi=10.5220/0009162703680375

[26] G. Gu, P. Zhang, S. Chen, Y. Zhang, and H. Yang, "Inflection point: a perspective on photonic nanojets," *Photonics Research*, vol. 9, no. 7, pp. 1157–1171, july 2021, publisher: Optica Publishing Group. [Online]. Available: https://opg.optica.org/prj/abstract.cfm?uri=prj-9-7-1157

[27] Y. A. Hong and J. W. Ha, "Enhanced refractive index sensitivity of localized surface plasmon resonance inflection points in single hollow gold nanospheres with inner cavity," *Scientific Reports*, vol. 12, no. 1, p. 6983, apr 2022, number: 1 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41598-022-11197-6

[28] I. Bronshtein, K. Semendyayev, G. Musiol, and H. Mühlig, *Handbook of Mathematics*. Berlin, Heidelberg: Springer, 2015. [Online]. Available: https://link.springer.com/10.1007/978-3-662-46221-8

A Appendix

The logistic sigmoidal function:

$$g\left(x\right)=\frac{H}{1+2^{A-Bx}} \tag{20}$$

For derivatives the following abbreviated notation is used:

$$g'(x)=\frac{\mathrm{d}}{\mathrm{d}x}\left[g(x)\right]=\left[g(x)\right]'$$

A.1 First derivative

The first derivative is given in the equation (21):

$$\begin{aligned}
g'\left(x\right) &= H\cdot\left[\tfrac{1}{1+2^{A-Bx}}\right]',\ pull\ out\ constant\ factors\\
g'\left(x\right) &= -H\cdot\tfrac{\left[1+2^{A-Bx}\right]'}{\left(1+2^{A-Bx}\right)^{2}},\ apply\ the\ reciprocal\ rule\\
g'\left(x\right) &= -\frac{H\left(\ln\left(2\right)\cdot 2^{A-Bx}\cdot\left[A-Bx\right]'+0\right)}{\left(1+2^{A-Bx}\right)^{2}}\\
g'\left(x\right) &= \frac{B\,H\,\ln\left(2\right)\,2^{A-B\,x}}{\left(1+2^{A-B\,x}\right)^{2}}
\end{aligned} \tag{21}$$

A.2 Second derivative

The equation (22) is the second derivative of the logistic function $g(x)$:

$$g''(x) = B\,H\,\ln(2)\left[\frac{2^{A-B\,x}}{\left(1+2^{A-B\,x}\right)^2}\right]',$$

pull out constant factors

$$g''(x) = B\,H\,\ln(2)\cdot \left[\frac{\left[2^{A-Bx}\right]'\left(1+2^{A-Bx}\right)^2-2^{A-Bx}\left[\left(1+2^{A-Bx}\right)^2\right]'}{\left(\left(1+2^{A-B\,x}\right)^2\right)^2}\right]',$$

apply the quotient rule

$$g''(x) = -\frac{B^2\,H\ln(2)^2\,2^{A-Bx}\left(1-2^{A-Bx}\right)}{\left(1+2^{A-Bx}\right)^3} \quad (22)$$

A.3 Third derivative

The third derivative $g'''(x)$ is:

$$B\,H\,\ln(2)\left[-\frac{B^2\,H\ln(2)^2\,2^{A-Bx}\,\left(1-2^{A-Bx}\right)}{\left(1+2^{A-Bx}\right)^3}\right]',$$

pull out constant factors

Applying the quotient rule and simplifications as in the second derivative:

$$\frac{B^3\,H\ln(2)^3\,2^{A-Bx}\,\left(1-4\cdot\,2^{A-Bx}+2^{2\,(A-Bx)}\right)}{\left(1+2^{A-Bx}\right)^4} \quad (23)$$

The same results were obtained using the GNU Octave software (version 9.1.0) with the following code:

Listing 1: Octave code for derivatives

```
pkg load symbolic;
syms x;
syms A;
syms B;
syms H;
Fx =  H./(1+2^(A-B.*x));

fprintf("First derivative\n");
d1 = diff(Fx,x);
r1 = simplify(factor(d1))
disp(d1);

fprintf("Second derivative:\n");
d2 = diff(d1,x);
r2 = simplify(factor(d2));
disp(r2);

fprintf("Third derivative\n");
d3 = diff(d2,x);
r3 = simplify(factor(d3));
disp(r3);
```

A.4 Turning point

A stationary point of a function is a point where the derivative of a function is equal to zero. The slope of the graph of the function is zero. They are called stationary points because they are points where the function is neither increasing nor decreasing.

A point c on a curve $y = g(x)$ is called a turning point if g is continuous and the curve changes from concave to convex (or vice versa) at c [26, 27].

According to the Method of Higher Derivatives (differential calculus), to obtain the turning point, it is first necessary to find the stationary point of $g'(x)$; that is, to solve the following equation:

$$g''(x) = 0,$$

or

$$-\frac{B^2\,H\ln(2)^2\,2^{A-Bx}\,\left(1-2^{A-Bx}\right)}{\left(1+2^{A-Bx}\right)^3} = 0 \quad (24)$$

The solution to the equation (24) is given by $x_0 = \frac{A}{B}$ and $y_0 = \frac{H}{2}$.

Finally, $g'''(x_0)$ is evaluated as follows:

$$F'''(x_0) = -\frac{B^3\,H\ln(2)^3}{8},$$
$$F'''(x_0) = -0.0416\,B^3\,H. \quad (25)$$

The result is different from zero (with $B \neq 0$ and $H \neq 0$), which indicates that $(\frac{A}{B}, \frac{H}{2})$ is a turning point. It should be highlighted that, according to the method of Higher Derivatives, if $F''(x_0) = 0$ holds, then x_0 must be substituted into the third derivative. Then, if $F'''(x_0) \neq 0$ holds, then it is a turning point. The method is explained and demonstrated in mathematics books [28].

B Appendix

Listing 2: ANN source code

```
library(tensorflow)
library(keras)
library(ggplot2)
library(devtools)
library(timetk)
library(mlbench)
library(dplyr)
library(magrittr)
library(gridExtra)
library(Metrics) # rmse, mae

 Pais <- 3
 CountryEn <- c("Brazil","Italy"
                ,"Peru","Switzerland")
 File = CountryEn[Pais];
```

```
f2<-paste("Covid19",File,".csv",sep="")
mydata<-read.csv(f2,header=TRUE
                 ,stringsAsFactors=FALSE
                 ,sep=",")
x  <- mydata$x;
y  <- mydata$y;
max_x <- max(x)
max_y <- max(y)
x_dias<-scale(x,center=mean(x)
              ,scale=sd(x))
ym <- mean(y)
ys <- sd(y)
y_train_arr <- (y-ym)/ys
lstm_model <- list()
lstm_model <-
  keras_model_sequential() %>%
  layer_lstm(units=200
             ,return_sequences=TRUE
             ,input_shape=c(1,1)) %>%
  layer_dense(units = 1)

lstm_model %>%
  compile(loss='mae'
          ,optimizer='adam'
          ,metrics=list('mae'))

start.time <- Sys.time()
lstm_model %>% fit(
  x = x_dias, y = y_train_arr,
  batch_size = 1,
  validation_split = 0.05,
  epochs = 500
)
end.time <- Sys.time();
time <- round(end.time - start.time,7);
print(time)

fitted0<- lstm_model %>%
  predict(x_dias,batch_size=1,verbose=0)
fitted <- fitted0*ys + ym
fitted <- round(fitted)

plot(x,y,col="grey",pch=2
     ,xlim=c(0,max_x),ylim=c(0,max_y))
par(new=TRUE)
plot(x,fitted,type="l",col="blue",lwd=5
     ,ylab="",main="",xlab=""
     ,xlim=c(0,max_x),ylim=c(0,max_y))

rc <- cor.test(fitted,y);
print(rc);
rc$estimate[[1]]
rc$estimate[[1]]^2
rc$conf.int[1]
rc$conf.int[2]
rmse(fitted0,y_train_arr)
mae(fitted0,y_train_arr)
```

Click-Through-Rate Prediction Using Deep Neural Networks and Efficient Channel Attention Mechanisms

Li Li
Anyang Normal University, Anyang 455000, China
E-mail: vvvlily@163.com

Keywords: deep neural networks, click-through prediction, online advertising, attention module, feature interaction

Received: Januar 2, 2025

The progress of technology and the Internet has brought the development of online advertising into a developed period. To optimize the accuracy of advertisement, click-through-rate rate, the research proposed an advertisement click-through-rate prediction model based on deep neural network combined with efficient channel attention network. This model consists of four parts: embedding layer, interaction layer, efficient channel attention layer, and prediction layer. The embedding layer is responsible for passing the feature vectors to the interaction layer. higher-order feature interactions are learned through deep neural networks and efficient channel attention networks are introduced for lower-order feature interactions. higher-order feature interactions can capture the nonlinear and complex relationships between original features, while low-level feature interactions mainly focus on the relationships between a few features. The channel attention layer integrates the original features with the features that have already been interacted with by the interaction layer. The prediction layer uses perceptrons to predict click-through-rates. The proposed model is compared with logistic regression, deep feature crossover network, and deep factorization machine on Criteo, Avazu, KDD12, and MovieLens-1M datasets. The results showed that when the network depth was 1, the area under the curve of the proposed model was 0.8377, which was 10.4% higher than that of the logistic regression model. The average logarithmic loss was 0.1985, which was lower than that of the comparison model. The UC value of the model in the KDD12 dataset was 0.7879 and the logarithmic loss value was 0.4478. Taken together, the proposed model of the study is able to predict click-through-rates more accurately and has better model performance.

Povzetek: Članek predstavi model za napovedovanje stopnje klikov (CTR) v spletnih oglasih, ki združuje globoke nevronske mreže in učinkovite mehanizme pozornosti, kar izboljša napovedi.

1 Introduction

The Internet has made an increasing amount of people tend to use the Internet to solve the items and other needs needed in daily life [1]. Today's Internet has penetrated into all aspects of life in various families, text message exchanges between people, online shopping, cell phone taxi travel and so on. Among the various types of service companies on the Internet revenue is mainly dependent on advertising revenue. Internet online advertising can rely on the huge user traffic, which can be transformed into operating income [2]. With the development of the Internet, the media used for advertisements have gradually evolved from television, magazines, and so on to the Internet, and online advertisements are commercial advertisements that are published in various websites with pictures, text, and so on [3]. For online advertising, how to send the advertisements to the people who need them is the main problem, and thus the click-through-rate (CTR) prediction, which is the most common way to calculate the effectiveness of the advertisement delivery, has been developed [4]. The Internet has made an increasing amount of people tend to use the Internet to solve the items and other needs needed in daily life [1]. Today's Internet has penetrated into all aspects of life in various families, text message exchanges between people, online shopping, cell phone taxi travel and so on. Among the various types of service companies on the Internet revenue is mainly dependent on advertising revenue. Internet online advertising can rely on the huge user traffic, which can be transformed into operating income [2]. With the development of the Internet, the media used for advertisements have gradually evolved from television, magazines, and so on to the Internet, and online advertisements are commercial advertisements that are published in various websites with pictures, text, and so on [3]. For online advertising, how to send the advertisements to the people who need them is the main problem, and thus the click-through-rate (CTR) prediction, which is the most common way to calculate the effectiveness of the advertisement delivery, has been developed [4]. For Internet service companies, CTR prediction is very important. In practical applications, CTR prediction models usually need to deal with the characteristics of high sparsity, which challenges the training and generalization ability of models, leading to the unsatisfactory prediction

performance of models [5]. By predicting and analyzing the CTR of the advertisements, it can improve the accuracy of the advertisements that Internet service companies push to the people in the network, and it can also predict in advance the return rate that the advertisements pushed can bring back [6]. However, traditional CTR prediction models have limitations in handling large-scale sparse data and capturing complex feature interactions, which cannot fully capture nonlinear relationships and higher-order feature (HOF) interactions in the data, resulting in poor generalization ability and low prediction efficiency of the model. In this context, the study aims to address the low accuracy and efficiency of traditional CTR prediction methods by proposing the use of deep neural network (DNN) combined with efficient channel attention network (ECANet) for CTR prediction. The research aims to improve the accuracy of CTR prediction by proposing a new deep learning model, which can help advertisers plan and optimize their advertising strategies more effectively, and contribute to the long-term stable development of e-commerce.

There are two main innovations of this study. The first point is to combine DNN and ECANet for feature interaction, which enables the model to comprehensively understand and utilize feature information, thereby improving the accuracy of CTR prediction. The second point is to effectively fuse the original features with the interactive features using channel attention layers, and adaptively adjust the Fws through attention mechanisms, further improving the predictive performance of the model.

The main contribution of this study is: (1) proposing a deep learning model architecture that integrates DNN and ECANet, which helps improve the accuracy of advertising CTR prediction. (2) Introducing an ECANet and enhancing the expressive ability of key feature channels through dynamic weight allocation solves the problem of traditional models lacking adaptive selection for low-level feature interactions, while improving the interpretability of the model.

2 Related work

As the Internet has grown, so too has online advertising, and predicting the CTR of advertisements has become increasingly crucial. Advertising CTR prediction is one of the core algorithms in advertising technology, mainly used to predict the probability of ads being clicked. Traditional CTR prediction methods include logistic regression (LR), deep factorization machine (DeepFM), and deep feature crossover network (DCN). Jia S stated that CTR is closely related to students' level of active learning, reflecting their learning behavior in the data. Therefore, for the analysis of student performance data in online teaching, LR algorithm was proposed to predict whether students will pass the exam. The experimental results proved the feasibility of the proposed method [7]. Li et al. proposed a CTR prediction model based on DeepFM and used focal loss as the loss function. The results showed that the area under the curve (AUC) of the proposed model was 0.044 and 0.013 higher than that of the logistic model and neural network, respectively [8]. Huang G et al. improved the predictive performance of CTR based on DNN by introducing a regularized leader following the DCN model. The results indicated that the proposed model had good predictive performance and helped promote the development of e-commerce [9]. However, traditional LR, DeepFM, and DCN also suffer from poor generalization ability and low efficiency, making it difficult to meet the high efficiency and precision requirements in network environments. With the rapid development of computer technology, more and more intelligent algorithms are being applied to the field of CTR prediction. Mao K et al. proposed a dual-stream feature interaction model based on multi-layer perceptron (MLP) for CTR prediction in online advertising and recommendation, which enhances CTR modeling through feature selection module and group bilinear fusion module. The results indicated that the proposed model achieved competitive performance compared to many existing dual-stream CTR models [10]. Qin J et al. proposed a user behavior retrieval framework that supports arbitrary and learnable retrieval functions to address the long-term dependency issue of current CTR prediction models. The results indicated that the proposed method achieved a 6.6% Effective Cost Per Mille gain in A/B testing [11]. Li X et al. proposed a decision context interaction network to learn decision context in response to the problem that existing CTR prediction methods often ignore the information background that affects users' click decisions. The results indicated that the proposed model could effectively improve the accuracy of CTR prediction [12]. The summary table of the relevant research on CTR prediction methods mentioned above is shown in Table 1.

Table 1: Summary table of related research

Literature	Method	Advantage	Limitation
[7]	LR	Simple and computationally efficient	Unable to capture complex interactions between features
[8]	DeepFM	AUC is 0.044 and 0.013 higher than the logistic model and neural network model, respectively	Poor performance when dealing with very sparse data
[9]	DCN	Enhancing the learning ability of feature interaction by introducing cross networks	There are limitations in terms of representativeness

[10]	Feature selection module and group bilinear fusion module	Capable of effectively handling feature interactions	Mainly relying on feature selection and bilinear fusion, there are limitations in handling high-order feature interactions
[11]	Learnable retrieval function	Can effectively process user behavior data and improve the long-term dependence of CTR prediction	The main focus is on user behavior retrieval, therefore there are limitations in handling low-level feature interactions
[12]	Decision context interaction network	Can effectively improve CTR prediction accuracy	Mainly focusing on the interaction of decision context, there are limitations in handling high-order feature interactions

ECANet is an efficient channel attention module that adopts a local cross channel interaction strategy without dimensionality reduction, avoiding information loss caused by dimensionality reduction. It is commonly used to improve the performance of deep learning models. Gao Q et al. proposed to merge the Neck part of ECANet and YOLO v5 network to build an improved YOLO v5 network-based object recognition model for pedestrian target tracking and recognition in complex scenes. The results showed that the average accuracy of the proposed model improved by 1.3% [13]. Guo Z et al. proposed a lightweight semantic segmentation algorithm based on multi-module fusion, which combines coordinate attention and ECANet to enhance salient features, to address the problem of unsatisfactory and inefficient semantic segmentation results in existing methods. The results showed that the average pixel accuracy of the proposed algorithm reached 85.23%, and the training speed was improved by 68.69% [14]. Cui Z et al. proposed an image deblurring aggregation network based on ECANet to address the problem of low feature extraction efficiency in existing multi-scale network image denoising methods. The network improved feature representation through three feature aggregation blocks. The results indicated that the proposed method could clearly restore the texture and color of the image [15]. Chen W et al. proposed a low contrast defect detection method based on deep learning for ceramic parts, and used ECANet to detect defects in curved parts. The results indicated that the prediction accuracy of the proposed method could reach 94.35%, with an average detection time of only 0.78 seconds [16].

From domestic and international research, the traditional deep learning network for the prediction of advertisement CTR has the problems of low prediction accuracy, small scope of application, and inability to make long-term prediction. The study proposed a method based on DNN combined with ECANet for advertisement CTR prediction, which is expected to enhance the accuracy of advertisement CTR prediction and achieve long-term prediction.

3 CTR prediction using DNN combined with ECANet

3.1 DNN based CTR prediction

CTR prediction is the process of predicting the CTR of the next advertisement that has a probability of being clicked by a web user under a specific conditional situation using an algorithm before clicking on the advertisement [17]. The advertisement CTR prediction process is in Figure 1.

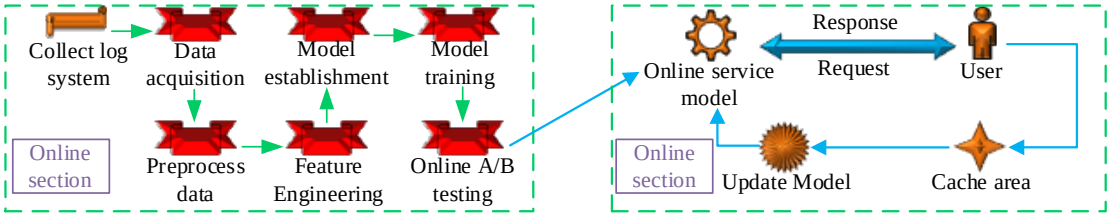


Figure 1: CTR predication process.

In Figure 1, the offline system collects and processes the data and trains the data, and passes the trained data into the online system, which delivers the advertisements to the user side through data caching. In traditional CTR prediction, the main role of feature engineering is to improve the prediction accuracy, which is improved by combining features to mine the potential relationship between different features. The CTR is calculated as Eq. (1).

$$CTR = \frac{num_click}{num_impression} \quad (1)$$

In Eq. (1) num_click means the quantity of clicks made by the web user and $num_impression$ is the number of times the advertisement appears on the web

user interface.

However, in actual user usage, it is possible that the web user does not make a click and thus there is a smoothing way to perform the formula leveling as in Eq. (2).

$$CTR = \frac{num_click + \alpha \times \beta}{num_impression + \beta} \quad (2)$$

In Eq. (2), α and β denote the smoothing parameters to avoid the problem that the denominator is zero leading to the inability to calculate the CTR, which can help the model to give reasonable predictions even in the case of sparse data. DNN is used to learn higher order feature interactions. In deep structural models, the DNN structure is divided into multi-layer networks, able to construct non-linear higher order features. This feature also makes DNN has a better performance in click rate prediction. DNN is more common in deep learning. In this problem of click prediction, the perceptron with additional layers in DNN is defined in equation (3).

$$f(x) = sign(\sum_{i=1}^{n} w_i x_i + b) \quad (3)$$

In Eq. (3), $sign$ denotes the sign function, n denotes the quantity of perceptron layers, x_i denotes the i-th input feature, w_i denotes the weight of x_i, and b denotes bias. Among them, input features refer to the raw data points used for training and predicting models. In the scenario of advertising CTR prediction, input features can include user features, advertising features, contextual features, historical behavior features, and interaction features. Weight is a parameter in DNNs that connects the input and output layers (OLs), representing the degree of influence of corresponding input features on the model output. During the training, weights are continuously adjusted through optimization algorithms to minimize the difference between predicted and actual values. This study uses the Adam optimization algorithm. In DNN, the output of the perceptron can be represented as a binary classification result, and the sign function is used to convert the output of the perceptron into a binary output sign function. The sign function is expressed as in Eq. (4).

$$sign(z) = \begin{cases} -1, x < 0 \\ 1, x \geq 0 \end{cases} \quad (4)$$

In Eq. (4), z denotes the input features. DNN increases the network layers on the perceptual machine, and to allow the model to learn HOFs, a hidden layer (HL) is added between the original input and OLs on the network model [18]. It makes the DNN can be diverse in choosing activation functions, and can choose more applicable activation functions according to the scene. To make the DNN capable of handling both regression and classification problems, neurons are added to the OL so that the OL of the DNN is no longer a single neuron [19]. Its DNN model network structure as in Figure 2.

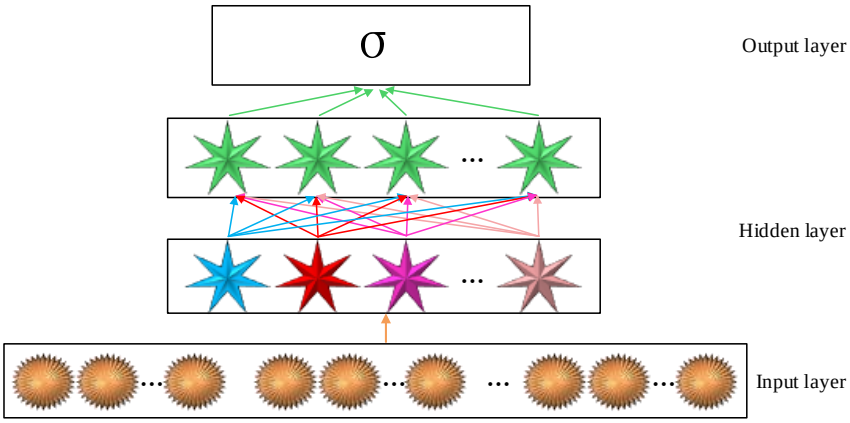


Figure 2: DNN model network structure.

In Figure 2, DNN has 3 layers: input, hidden, and output. The input layer (IL) sends the information feature collection to the HL, and after the training, the OL will work with the trained information features as output. The larger and more complex the dataset, the more HLs and units may be needed to capture patterns in the data. But more HLs and units also mean higher computational costs, and too many HLs and units may also lead to overfitting. Therefore, determining the optimal number of HLs and units usually requires experimentation and tuning. Assuming the number of HLs is L, the calculation of the input layer for the DNN model is shown in equation (5).

$$X = \{X_1, X_2, X_3, \ldots, X_n\} \quad (5)$$

In Eq. (5), X denotes the input vector. The HL is expressed as in Eq. (6).

$$h^{(l)} = h(W^{(l)}h^{(l-1)} + b^{(l)})(\forall l \in 1,2,3,\ldots,L-1) \quad (6)$$

In Eq. (6), $h^{(l)}$ and $b^{(l)}$ denote the input vector and bias vector of layer $l+1$ and l. $W^{(l)}$ denotes the weight matrix of layer $l-1$. The OL is represented as in Eq. (7).

$$y_{pre} = \arg\max_{C} P(y = C, X; w, b) \quad (7)$$

In Eq. (7), $P(y = C, X; w, b)$ denotes the possibility that the output category is equivalent to C, and the outcome category is either 0 or 1. In the forward propagation process of DNN, input features are input from the input layer, pass through multiple HLs, and finally output the predicted values of the model. It is necessary to optimize the objective function, and the minimum mean square error (MSE) is usually used to calculate the error between the predicted value and the true value, as shown in equation (8).

$$MSE = \frac{1}{M}\sum_{m=1}^{M}(y_m - \hat{y}_m)^2 \quad (8)$$

In Eq. (8), M denotes the number of samples. The activation value of the last HL a^H is a dense vector that contains all the information from the input layer to the HL. The study uses the Sigmoid function as the activation function for the OL, and introducing a^H into the Sigmoid function can generate the final predicted value, as shown in equation (9).

$$y_{DNN} = \sigma(W^{|H|+1} \cdot a^H + b^{|H|+1}) \quad (9)$$

In Eq. (9), $|H|$ is the number of layers of the HLs in the DNN. In the process of regression prediction, the loss function is expressed as in Eq. (10).

$$loss = \frac{1}{N}\sum_{i=1}^{N}(V_i - M_i)^2 \quad (10)$$

In Eq. (10), V_i denotes the predicted output and M_i denotes the actual output. In order for the loss function to become minimized, the network is to be trained and iteratively updated by Adam, expressed as in Eq. (11).

$$\begin{cases} v = \beta_1 v + (1-\beta_1)dw \\ s = \beta_2 s + (1-\beta_2)dw^2 \end{cases} \quad (11)$$

In Eq. (11), $\beta_1 = 0.9$, $\beta_2 = 0.999$. w denotes the changeover value and v, s denote the variables in the update process.

3.2 CTR prediction based on DNN-ECANet

Lower-order features (LOF) refer to basic features directly extracted from raw data, usually involving a single feature or a simple combination of a few features. They are rich in detail information and can capture subtle changes and local features in the data. HOF refer to features obtained through multiple nonlinear transformations and abstract representations of raw data, which can capture complex relationships and high-level semantic information in the data, usually involving interactions and combinations between multiple features. The study presents the ECANet in conjunction with DNN for the learning fusion of HOF and LOF interactions for advertisement CTR prediction. DNN is primarily used to perform HOF interactions, and its learning ability for LOF interactions is weak. DNN can automatically learn HOF interactions in data through multi-layer nonlinear transformations. In advertising CTR prediction, DNN can capture the complex relationships between user behavior, advertising attributes, and other features, providing richer information for prediction. The core of ECANet is the channel attention mechanism, which captures the dependency relationships between channels through one-dimensional convolution operations, enhancing the model's attention to important features. The combination of ECANet and DNN achieves dynamic modeling of multi-level feature interaction through efficient channel attention mechanism and complementary deep nonlinear learning. Compared with traditional attention mechanisms, ECANet proposes an adaptive method for selecting the size of one-dimensional convolution kernels, which avoids complex dimensionality reduction and enhancement processes and has higher computational efficiency. The ECANet is taken to learn the LOF interactions, and the problem of prediction for one in different application scenarios is different, for a specific CTR prediction, the ECANet is adopted to increase the feature weighting (Fw) and decrease the interference weights [20]. ECANet consists of three parts, as in Figure 3.

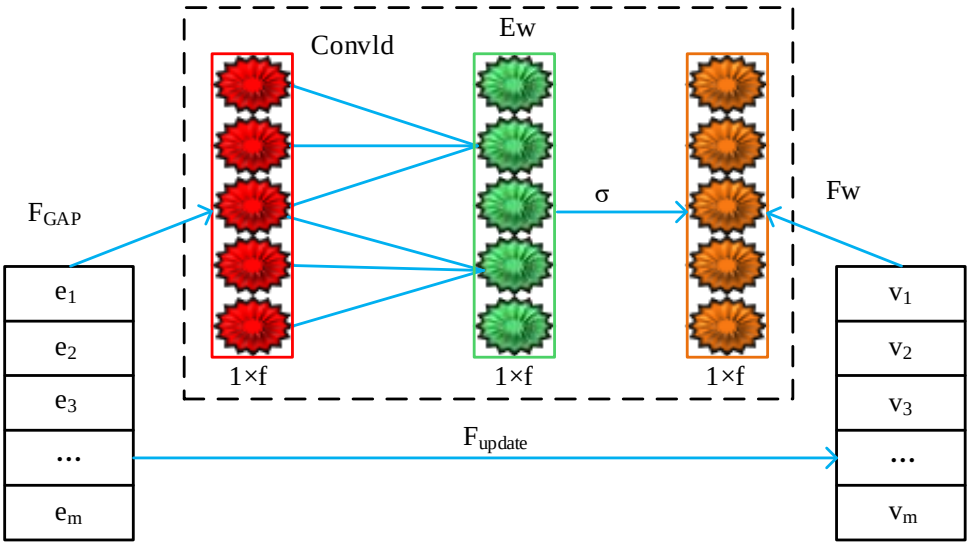


Figure 3: Composition of ECANet.

In Figure 3, ECANet consists of three parts: feature compression, weighting, and Weight extraction (Ew). ECANet takes the feature information and passes it to different network layers for weighting calculation by compressing the features. Finally, the weights are extracted by the weighting layer after the weighting calculation. Feature compression is used to get the received vector information and embed the vectors into ECANet [21]. It is calculated as in Eq. (12).

$$z_i = F(e_i) = \frac{1}{k}\sum_{t=1}^{k} e_i^{(t)} \quad (12)$$

In Eq. (12), z_i denotes the statistical vector of the i -th input vector, $e_i^{(t)}$ denotes the t -dimensional information of i -th vectors, and $t = 1, 2, 3, \cdots, k$. To better capture the relationship between different features and improve the accuracy of advertisement CTR prediction, DNNs are utilized for advertisement CTR prediction, and the structure of DNN for CTR prediction is in Figure 4.

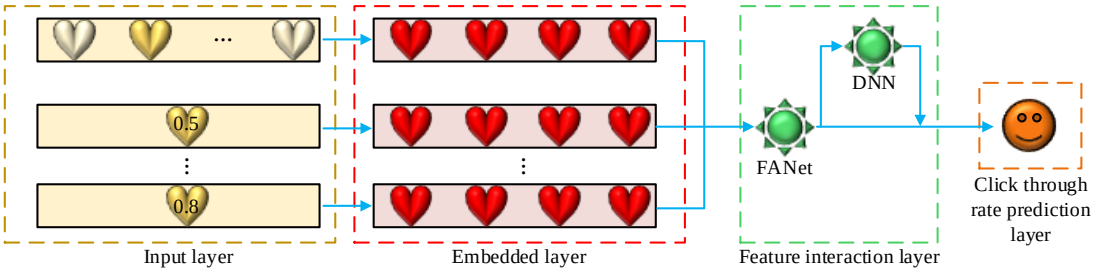


Figure 4: Structure of DNN for CTR prediction.

In Figure 4, DNN for click prediction has four parts, which are IL, embedding layer, feature interaction layer, and click prediction layer. The feature information is passed from the IL to the embedding layer, and the embedding layer further sends the processed features to the feature interaction layer through computation. The feature interaction layer integrates and interacts the received feature information, and finally the prediction layer makes the prediction. Ew is used to learn the weights of the embedding. The information is obtained using a single feature with the feature it wants to be close to, which is realized by convolution, and the weights of its field embeddings are calculated as in Eq. (13).

$$A = \sigma(conv1D_K(Z)) \quad (13)$$

In Eq. (13), $conv1D$ denotes convolution, $A \in R^m$ denotes a vector, K denotes the size of the convolution kernel, and σ is the activation function. Fw is the most initial vector multiplied by the weights obtained from Ew, and thus the embedding vector obtained by fusion of the two, computed as in Eq. (14).

$$V = F(A, E) = [a_1 \cdot e_1, \cdots, a_m \cdot e_m] = [v_1, \cdots, v_m] \quad (14)$$

In Eq. (14), $a_i \in R$, $e_i \in R^k$, $v_i \in R^k$. i denotes the i feature domain and k denotes the dimension of the embedding. In the established model, the DNN is interacted with the hidden features, which is calculated as in Eq. (15).

$$\begin{cases} x^1 = \sigma(W^{(1)}v + b^1) \\ x^n = \sigma(W^{(n)}v^{(n-1)} + b^n) \end{cases} \quad (15)$$

In Eq. (15), n denotes the neural network depth, x^n denotes the DNN output at layer n, and W denotes weight, and b denotes bias. In explicit feature interaction, ECANet is utilized. According to the DNN combined with ECANet for constructing the CTR prediction model, the study proposes that the structural composition of the DNN-ECA model is in Figure 5.

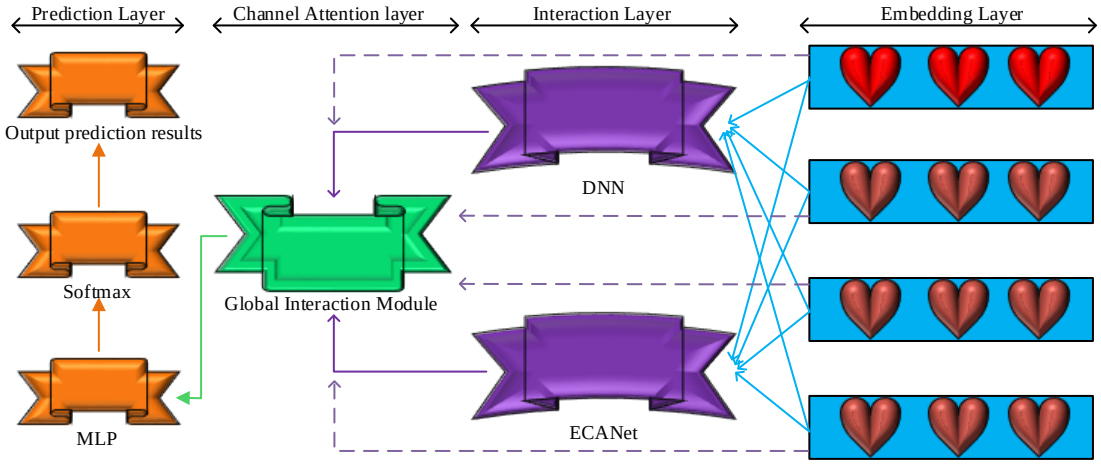


Figure 5: DNN-ECA model structure.

In Figure 5, four parts, the embedding layer, the interaction layer, the ECA layer, and the prediction layer, together form the DNN-ECA CTR model. Among them, the embedding layer maps the original feature vectors to a low dimensional embedding space for subsequent processing. The interaction layer consists of DNN and ECANet. DNN can capture complex feature combinations and learn high-order interaction relationships between features through multi-layer nonlinear transformations. ECANet focuses on learning lower-order interaction relationships between features. The efficient channel attention layer integrates the features output by the embedding layer with the features processed by the interaction layer, and adaptively adjusts the weights of different features through the attention mechanism to capture the relationship between the original features and the interaction features. The prediction layer consists of an MLP and a Softmax layer. MLP can perform deeper feature interaction and learning, while Softmax converts the output of MLP into a probability distribution for predicting CTR, ultimately outputting the predicted results for advertising CTR. That is, all the captured features are represented by a vector, as in Eq. (16).

$$r_g = [e_i; e_{ECA}; \tilde{e}_j] = [r_1; r_2; r_3] \quad (16)$$

In Eq. (16), r_1, r_2, and r_3 denote individual features. All the features are combined as inputs and the model output is calculated as in Eq. (17).

$$R_g = \sum_{l=1}^{L} \frac{exp(\tanh(r_l \cdot W_l + \hat{b}_l))}{\sum_{l=1}^{L} exp(\tanh(r_l \cdot W_l + \hat{b}_l))} r_l \quad (17)$$

In Eq. (17), W_l denotes the weights, b_l denotes the deviation matrix, and $\tanh$ denotes the activation function. In the final prediction layer, the adapted global features are delivered to the prediction module for CTR prediction using the perceptron machine, and the prediction process is in Eq. (18).

$$R_l = Relu(W_l R_l + b_l) \quad (18)$$

In Eq. (18), the value of l ranges from $l = 1, 2, \cdots, n$. For the lowest implicit layer, the final CTR prediction is performed for it, expressed as in Eq. (19).

$$\hat{y} = softmax(W_P R_h + b_q) \quad (19)$$

In Eq. (19), $\hat{y}$ denotes the prediction result, R_h denotes the output at layer h, which is the output of the prediction module, W_P represents the weight, and b_q denotes the deviation. In summary, the specific flowchart of the DNN ECANet model proposed by the research is shown in Figure 6.

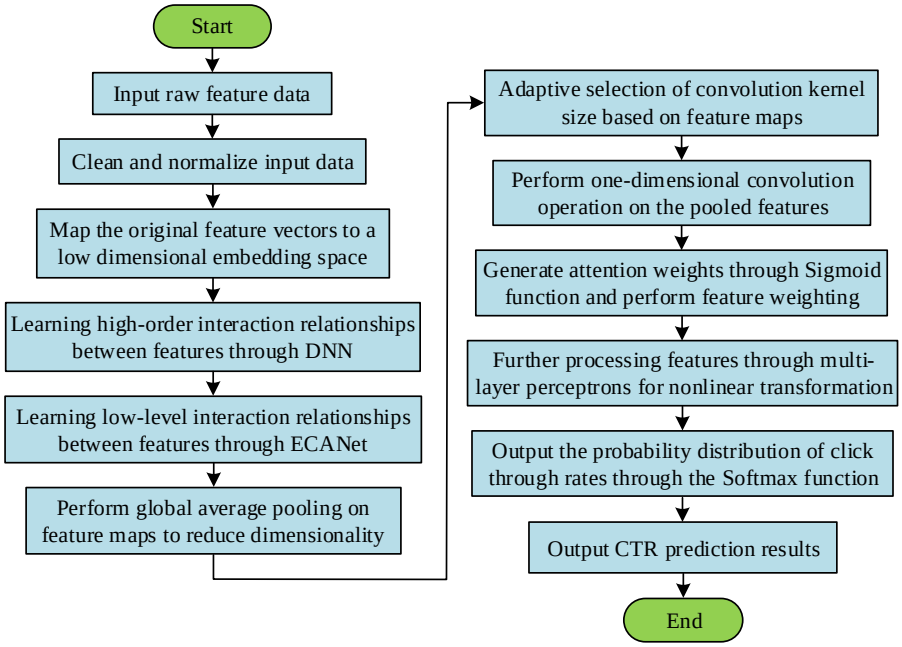


Figure 6: Flow chart of DNN ECANet model.

The study used AUC value and LogLoss as evaluation indicators. AUC is the area under the receiver operating characteristic (ROC) curve of the subject, and its value is positively correlated with the model performance. The horizontal and vertical axes of the ROC curve represent false accuracy and true accuracy, respectively. LogLoss is used to measure the distance between two components, and its calculation is shown in Eq. (20).

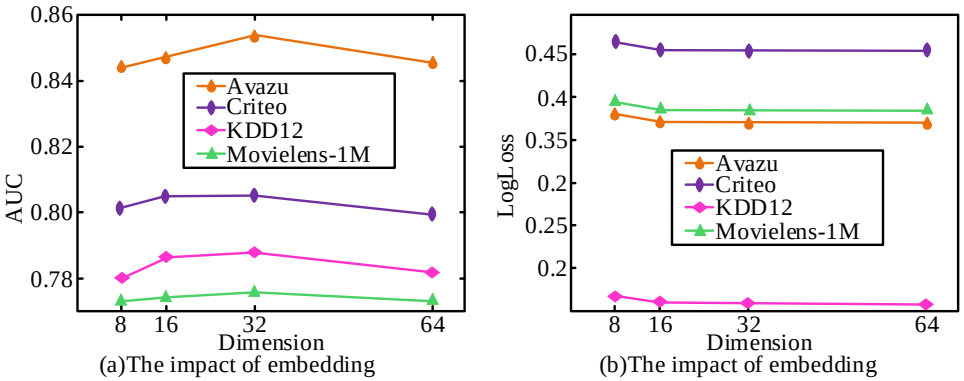
$$LogLoss = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] \tag{20}$$

In Eq. (20), N denotes the total number of samples, y_i denotes the true value, and $\hat{y}_i$ denotes the predicted value.

4 Analysis of experimental results

4.1 Validity analysis of the DNN-ECANet Model

The experiment was run on the Windows 11 system and implemented in the TensorFlow 2.0 environment, AMD Ryzen 75800H CPU. The host memory was 32.0GB, and the graphics card was NVIDIA GeForce RTX 3060. The study used Python programming language, Softmax activation function, set layer depth to 3, number of neurons to 128, embedding dimension to 16, Drop Out Ratio to 0.7, learning rate to 0.001, batch size to 1024, and optimized the model using Adam. To analyze the predictive effectiveness of the DNN ECANet model, AUC and LogLoss were used as evaluation metrics in the study. AUC measured the probability that the predicted value of positive samples was higher than that of negative samples, and the larger the value, the better the predictive performance of the model. Logloss was the most important classification metric based on probability, used to measure the difference between predicted and true values. The smaller the value, the better the predictive performance of the model. The four datasets were Criteo, Avazu, KDD12, and MovieLens-1M. The Criteo dataset contains various information such as users, merchants, and advertisements, and is a classic dataset on consumer behavior and machine learning algorithms. The Avazu dataset provides rich information through multiple feature fields containing user characteristics and advertising attributes, and has classic application scenarios in the field of mobile advertising CTR prediction. The KDD12 dataset contains a large amount of user behavior data, which can help researchers evaluate and develop CTR prediction algorithms. The MovieLens-1M dataset contains user ratings, tags, and background information for movies. These four public datasets cover common application areas in different real-world scenarios, including data related to user preferences and behavior patterns, and can test the performance and generalization ability of the model under different conditions. Ten million data were randomly selected from each dataset and divided into training, testing, and validation sets in the ratio of 8:1:1. The experimental results are in Figure 7.



(a)The impact of embedding dimension on AUC

(b)The impact of embedding dimensions on LogLoss

Figure 7: Impact of embedding dimensions on the model.

From Figure 7 (a), as the embedding dimension increased from 8 to 32, the AUC values of the model gradually increased on all four datasets. When the embedding dimension reached 32, continuing to increase the embedding dimension would gradually decrease the AUC value of the model, and the performance would begin to decline. From Figure 7 (b), as the embedding dimension increased from 8 to 16, the LogLoss of the model gradually decreased, indicating that the prediction accuracy of the model gradually increased. When the embedding dimension reached 16, even if the embedding dimension continues to increase, there was no significant change in the LogLoss of the model. This may be because appropriately increasing the embedding dimension can enable the model to capture richer feature representations, thereby improving the model's ability to distinguish between positive and negative samples. But if the embedding dimension is too large, it may lead to overfitting of the model and a decrease in generalization ability on the test data. Therefore, it is reasonable to set the embedding dimension to 16 in the study. To further investigate the impact of the number of layers and neurons in each layer of DNN on the robustness of the model, the layers were set to 2, 3, and 4, and the number of neurons was set to 64, 128, and 256, respectively. The changes in AUC and LogLoss values of the DNN-ECA prediction model proposed by the research on the Criteo dataset are shown in Table 2.

From Table 2, as the number of layers increased, the AUC and LogLoss values of the model did not show a significant monotonic trend. This indicated that the performance of the model did not solely depend on an increase in the number of layers.

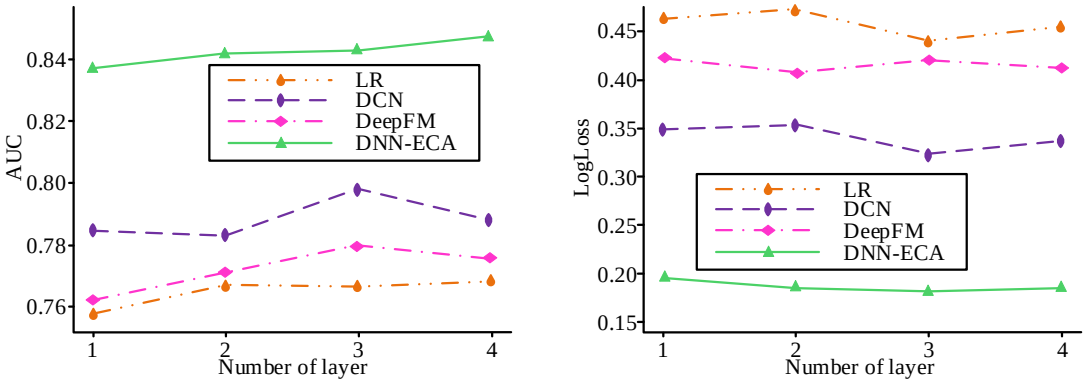
Table 2: AUC and LogLoss values of DNN-ECA model under different hyperparameters

Number of layers	Number of neurons	AUC	LogLoss
2	64	0.821	0.453
2	128	0.825	0.439
2	256	0.818	0.452
3	64	0.831	0.435
3	128	0.837	0.428
3	256	0.829	0.437
4	64	0.824	0.441
4	128	0.832	0.436
4	256	0.829	0.438

In addition, when the number of layers was fixed, increasing the number of neurons usually increased the AUC value of the model and reduced the LogLoss value.

When the number of layers was 3 and the number of neurons was 128, the model performed the best in terms of AUC and LogLoss values, which were 0.837 and 0.428, respectively.

Therefore, it is reasonable to set the layer depth to 3 in the study. The LR model, DCN model, and DeepFM are three classic CTR prediction models that are representative and highly applicable, providing comprehensive comparisons and references for this study. The performance of DNN-ECA was compared with LR model, DCN model, and DeepFM at different network depths, and the experimental results are in Figure 8.



(a)The impact of network depth on AUC (b)The impact of network depth on LogLoss

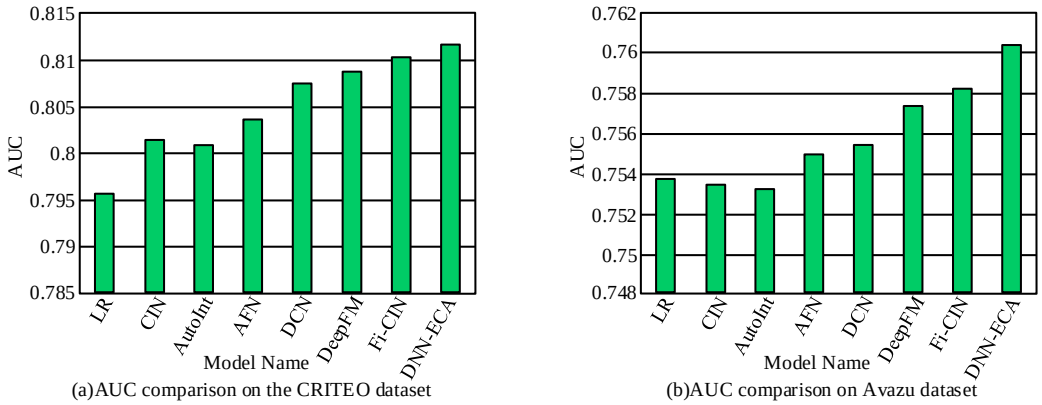
Figure 8: The influence of network depth on the model.

From Figure 8(a), as the network depth increased, the AUC value of the DNN-ECA model proposed by the research also increased continuously. When the network depth was 1, the AUC value of the proposed model was 0.8377, which was 10.4% higher than that of the LR model. The average AUC value of the proposed model was 0.845, which was higher than the comparison model. The average AUC value of the LR model was the lowest, at 0.765. From Figure 8 (b), at different network depths, the LogLos of the proposed model was always the lowest and the variation amplitude was small, with an average LogLos of 0.1985. The LogLos of the other three prediction models fluctuated continuously with the increase of network depth, which greatly affected the predictive performance of the models. The results indicated that the DNN-ECA prediction model proposed by the research had better predictive performance than LR, DCN, and DeepFM models, and was less affected by network depth. Further comparing the computational complexity of the four models mentioned above, the results are shown in Table 3.

Table 3: Comparison of computational complexity among four models

Model	Number of parameters/M	Training time/s
LR	0.01	150
DCN	1.22	163
DeepFM	1.54	201
DNN-ECA	1.36	179

From Table 3, the parameter of DNN-ECA was 1.36M and the training time was 179s, which was superior to the DeepFM model in terms of computational complexity. In the public dataset Criteo, the DNN-ECA was compared with the commonly used CTR prediction model for machine learning performance. Figure 9 shows the comparative results.



(a)AUC comparison on the CRITEO dataset (b)AUC comparison on Avazu dataset

Figure 9: Comparison of AUC performance of different models on the dataset.

Figure 9(a) shows that when comparing the model performance in the Criteo dataset, the DNN-ECA model had the highest AUC of 0.8124. While the commonly used LRM had an AUCof 0.7951, which was 0.0173 lower than that of the DNN-ECA. This indicated that compared to commonly used prediction models in the Criteo dataset, DNN-ECA had the best predictive performance. In Figure 9(b), the Avazu dataset, the DNN-ECA model had the highest AUC value of 0.7612, and the LRM had an AUC value of 0.7538, which was 9.8% lower than the AUC of the DNN-ECA. the AutoInt model had the lowest AUC value of 0.7531. This showed that the DNN-ECA had the best prediction performance in the dataset Avazu. Set Avazu had the best prediction performance, higher AUC value than the commonly used CTR prediction model, and better performance than the classical deep learning prediction model. In Criteo, Avazu and Movielens-1M datasets, the DNN-ECA and the commonly used CTR prediction model were subjected to log-loss experiments. Table 4 lists the specific data.

Table 4: Performance comparison of various models on the dataset

Model	Criteo	Avazu	Movielens-1M
	LogLoss	LogLoss	LogLoss
LR	0.4756	0.1679	0.4416
CIN	0.4695	0.1598	0.4269
AutoInt	0.4560	0.1581	0.4273
AFN	0.4471	0.1569	0.4149
DCN	0.4416	0.1589	0.4035
DeepFM	0.4511	0.1588	0.3891
Fi-CIN	0.4483	0.1544	0.3843
DNN-ECA	0.4462	0.1539	0.3795

From Table 4, the LogLoss of the proposed model on the Criteo dataset was 0.4462, which was 0.0294 lower than that of the LR model. The LogLoss on the Avazu dataset was 0.1539, which was 0.005 lower than the DCN model. The LogLoss on the Movielens-1M dataset was 0.3795, which was 0.024 lower than the DeepFM model. To further investigate the impact of the number of input fields on the performance of DNN-ECA, the Criteo dataset was used to compare DNN-ECA with DeepFM. The experimental results are shown in Figure 10.

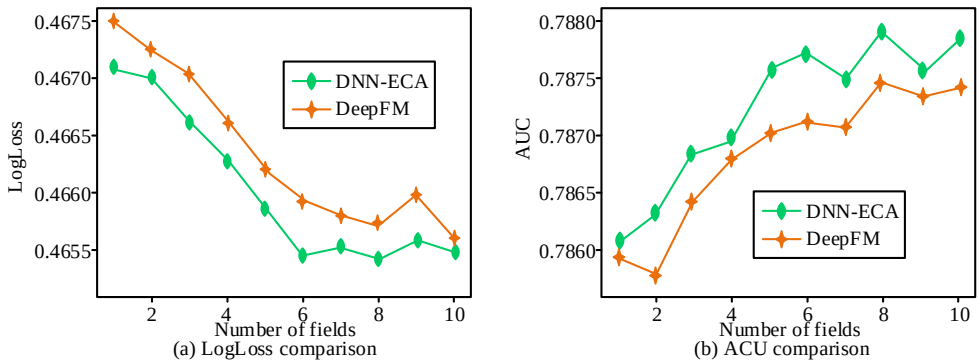


Figure 10: Influence of different contextual numbers on the model.

From Figure 10 (a), the LogLoss value of the proposed DNN-ECA model was consistently lower than that of the DeepFM model under different input field quantities. When the number of input fields was 8, the LogLoss of the model was the lowest, which was 0.4654. The highest LogLossof the DeepFM model was 0.4675, and the lowest LogLoss was 0.4656. From Figure 10 (b), the lowest AUC value of the DNN-ECA model was 0.7862, and the highest AUC value was 0.7879. Under different input field quantities, the AUC value of the DNN-ECA model was consistently higher than that of the DeepFM model. It demonstrates that the DNN-ECA has more learning ability and is better for the evaluation of prediction. To investigate the impact of ECANet on model performance, ECANet was removed from the DNN-ECA model and the AUC values of the complete model and the model were compared with ECANet removed in four datasets. The results are shown in Figure 11.

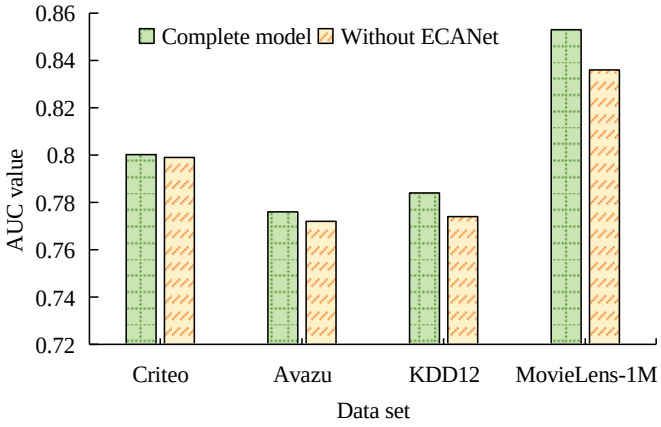


Figure 11: Impact of ECANet on model performance.

From Figure 11, in all four datasets, the AUC values of the complete model were higher than those of the model with ECANet removed, indicating that ECANet can effectively improve the CTR prediction performance of the model.

4.2 Validation of the effectiveness of DNN-ECA CRT Prediction

To verify the practical application effect of the proposed DNN-ECA model, data collection was conducted on the campus forum network of Sichuan Agricultural University. Information was collected from 200 users and 500 posts from October 1, 2024 to October 31, 2024. The collected data were stored in a MySQL database. Duplicate user and post information were removed through unique identifiers such as user ID and post ID. Operations such as word segmentation, stem extraction, and morphological restoration were performed on post and comment content to extract meaningful features. The DNN-ECA model was compared with four commonly used deep learning models: Factorization-machine supported Neural Networks (FNN), Neural Factorization Machines (NFM), Wide and Deep Learning Model (Wide&Deep), and DeepEM. The FNN model consisted of a factorization machine (FM) and an MLP, and has good feature cross expression ability. The NFM model integrated FM and DNN through a dual line interactive pooling layer. The Wide&Deep model combined LR and DNN for learning. The comparison results of the indicators of the five models are shown in Figure 12.

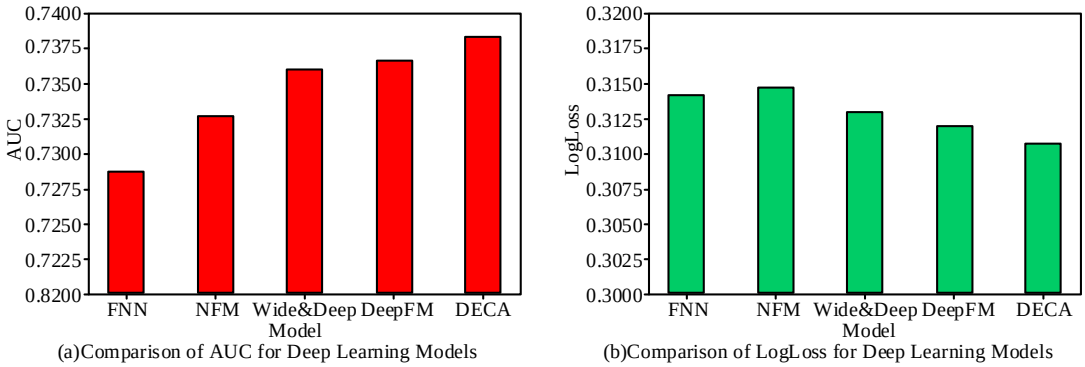


Figure 12: Comparison between DNN-ECA model and deep learning model.

From Figure 12(a), results indicated that the highest and lowest AUC value of the DNN-ECA and FNN were 0.7379 and 0.7286. For the traditional DeepFM for CTR prediction, the AUC value was 0.7361. It was 0.011 lower than that of the AUC of the DNN-ECA. The DNN-ECA had the best performance and higher accuracy of CTR prediction compared to the rest of the deep models. In Figure 12(b), the DNN-ECA model had the lowest LogLoss value of 0.3119, which was 0.349% lower than that of the DeepFM, and 0.547% lower than NFM model. It indicated that the proposed DECA performed better compared to other deep learning models commonly used for CTR prediction. The DNN-ECA model could simultaneously perform the fusion of learning different higher and lower-order features. The comparison of CTR prediction of different models on KDD12 dataset is in Figure 13.

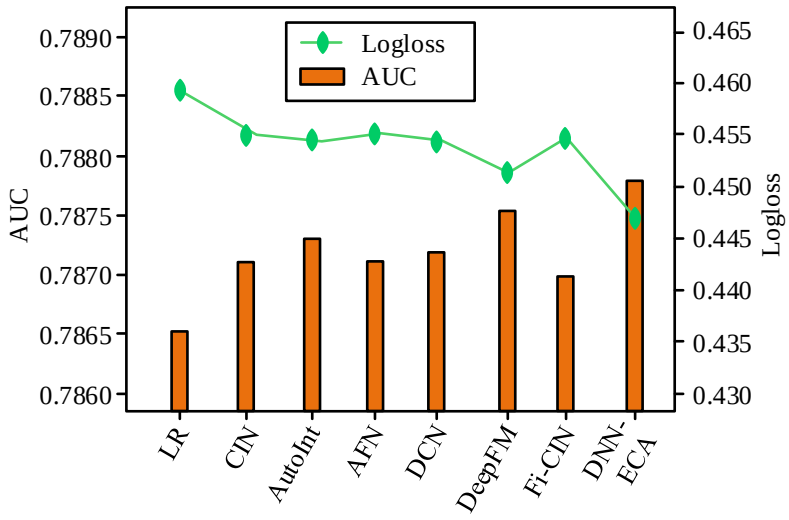


Figure 13: Comparison of CTR prediction of models on KDD12.

In Figure 13, when different models were used for CTR prediction on the KDD12 dataset, the LRM had the worst CTR prediction performance, with an AUC value of 0.7866 and a LogLoss value of 0.4587. The DeepFM had a better CTR prediction performance compared to the rest of the commonly used prediction models, with an AUC of 0.7876 and a LogLoss value of 0.4529. The DNN-ECA had the best CTR prediction performance with an AUC of 0.7879 and a LogLoss value of 0.4478. Compared to the LRM, the AUC value improved by 0.0013 and the LogLoss value reduced by 0.24%. Compared with the DeepFM, the AUC increased by 0.003 and the LogLoss value decreased by 0.19%. To further validate the superiority of the proposed model, the KDD12 dataset was used to compare the prediction accuracy of the DNN-ECA model with three advanced models: Automatic Feature Interaction (AutoInt), Feature Importance Bilinear Feature Interaction (FiBiNet), and Decision context interaction network (DCINet). The results are shown in Figure 14.

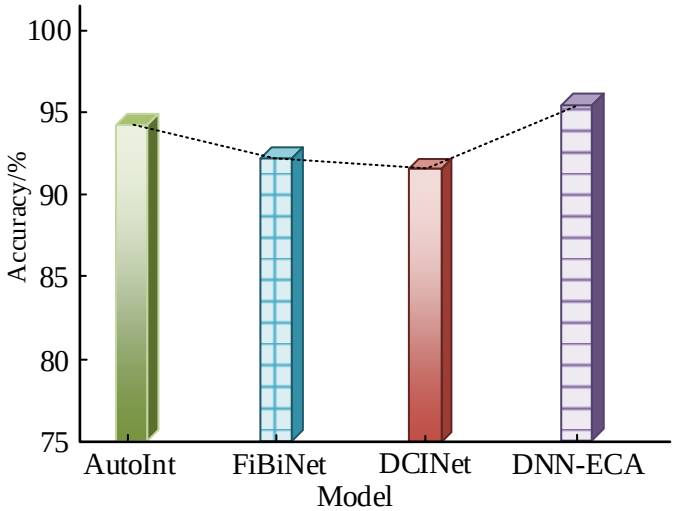


Figure 14: Comparison of prediction accuracy among four models.

From Figure 14, among the four models, the DNN-ECA model proposed by the research had the highest accuracy, at 96.89%. Next was the AutoInt model, while the DCINet model had the lowest accuracy, but still above 90%. The results indicated that the proposed DNN-ECA model had a high prediction accuracy and certain superiority.

5 Discussion

To optimize the accuracy of advertisement CTR prediction, the study proposed a method grounded on the fusion of DNN and ECANet for CTR prediction. The results demonstrated that on different datasets, when the embedding dimension of the model was 32, the AUC value of the model was the highest and the prediction performance was the best. Continuing to increase the

embedding dimension, the AUC value of the model began to decrease. This meant that the model could achieve good performance at lower embedding dimensions. Continuing to increase embedding dimensions may cause the model to start memorizing specific samples in the training data instead of learning generalized feature representations, resulting in overfitting and high complexity, which can affect its generalization ability. In addition, the DNN-ECA model proposed by the research performed better on the Criteo dataset, possibly because the Criteo dataset typically contains a large number of sparse features. When the embedding dimension was 32, the model had sufficient feature representation ability to capture complex relationships between sparse features.

The DNN-ECA had a lower LogLoss compared to the LRM. In the Avazu dataset, the LogLoss of the DNN-ECA was 0.014 lower than the LRM and 0.005 lower than the DCN, and the AUC value was 0.017 higher than the LRM and 0.0061 higher than the DCN. Under different input field quantities, the LogLoss value of the DNN-ECA model was consistently lower than that of the DeepFM model. When the number of input fields was 8, the LogLoss of the model was the lowest, which was 0.4654. In the KDD12 dataset, the AUC value of the DNN-ECA model was 0.7879, and the LogLoss value was 0.4478. Compared to the LR model, the AUC value increased by 0.0013 and the LogLoss value decreased by 0.24%. DNN-ECA model proposed by the research had the highest accuracy, at 96.89%. Next was the AutoInt model, while the DCINet model had the lowest accuracy, but still above 90%. The DNN-ECA model proposed by the research exhibited excellent performance compared to traditional LR [7], DeepFM [8], and DCN [9], mainly due to the model's use of DNN, ECANet, and channel attention to achieve cross level feature interaction fusion. In addition, the attention mechanism adopted by the DNN-ECA model adaptively captured the dependencies between channels through one-dimensional convolution, without the need for dimensionality reduction or enhancement, preserving the information integrity of the original channel features and reducing model complexity and computational burden.

6 Conclusion

In summary, the study's proposed DNN-ECANet method for advertisement CTR prediction can improve the prediction accuracy and help the long-term development of the advertising industry in the Internet environment. The DNN-ECA model proposed in the study has great potential for application in niche markets and specific time environments, and is expected to provide more accurate advertising strategies for advertisers and platforms. Specifically, in niche markets, advertisers often face issues such as small data volumes and specific user groups. By optimizing feature engineering and introducing multimodal data, the DNN-ECA model can better adapt to the needs of niche markets and improve the accuracy of advertising placement. In addition, user behavior patterns may change during specific time periods, and the DNN-ECA model can optimize advertising placement in real-time and improve CTRs by introducing time series features and dynamically adjusting strategies. Although the DNN-ECA model proposed in the study can effectively capture higher-order and lower-order feature interactions, thereby improving CTR prediction performance, the complexity of the model is also high, resulting in longer training and inference times, and requiring a large amount of training data to achieve optimal performance. Therefore, in future research, methods for dynamically adjusting the learning rate, such as gradient descent-based dynamic adjustment strategy or learning rate annealing strategy, can be further explored to improve the convergence speed and stability of the model, or to further enhance the prediction performance by integrating multiple deep learning models.

References

[1] Yang Y, Xu B, Shen S, Shen F, Zhao J. Operation-aware Neural Networks for user response prediction. Neural Networks, 2020, 121(8): 161-168. https://doi.org/10.1016/j.neunet.2019.09.020

[2] Todri V, Ghose A, Singh P V. Trade-offs in online advertising: Advertising effectiveness and annoyance dynamics across the purchase funnel. Information Systems Research, 2020, 31(1): 102-125. https://doi.org/10.1287/isre.2019.0877

[3] Aribarg A, Schwartz E M. Native advertising in online news: Trade-offs among clicks, brand recognition, and website trustworthiness. Journal of Marketing Research, 2020, 57(1): 20-34. https://doi.org/10.1177/00222437198797

[4] Lyu Z, Dong Y, Huo C, Ren W. Deep match to rank model for personalized click-through rate prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(1): 156-163. https://doi.org/10.1609/aaai.v34i01.5346

[5] Yuan L, Pan Z, Sun P, Wei Y, Yu H. Deep context interaction network based on attention mechanism for click-through rate prediction. Journal of Intelligent & Fuzzy Systems, 2021, 41(6): 6899-6914. https://doi.org/10.3233/JIFS-210830

[6] Gai P J, Klesse A K. Making recommendations more effective through framings: Impacts of user-versus item-based framings on recommendation click-throughs. Journal of Marketing, 2019, 83(6): 61-75.

[7] Jia S. Logistic Regression Analysis of Online Course Click Rate. 2021 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE). IEEE, 2021: 388-391. https://doi.org/10.1109/MLISE54096.2021.00081

[8] Li L, Hong J, Min S, Xue Y. A novel CTR prediction model based on DeepFM for taobao data. 2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID). IEEE, 2021: 184-187. https://doi.org/10.1109/AIID51893.2021.9456556

[9] Huang G, Chen Q, Deng C. A new click-through rates prediction model based on Deep&Cross network.

Algorithms, 2020, 13(12): 342-342. https://doi.org/10.3390/a13120342

[10] Mao K, Zhu J, Su L, Cai G, Li Y, Dong Z. FinalMLP: an enhanced two-stream MLP model for CTR prediction. Proceedings of the AAAI conference on artificial intelligence. 2023, 37(4): 4552-4560. https://doi.org/10.1609/aaai.v37i4.25577

[11] Qin J, Zhang W, Su R, Liu Z, Liu W, Zhao G, et al. Learning to retrieve user behaviors for click-through rate estimation. ACM Transactions on Information Systems, 2023, 41(4): 1-31. https://doi.org/10.1145/3579354

[12] Li X, Chen S, Dong J, Zhang J, Wang Y, Wang X, Wang, D. Decision-making context interaction network for click-through rate prediction. Proceedings of the AAAI Conference on Artificial Intelligence. 2023, 37(4): 5195-5202. https://doi.org/10.1609/aaai.v37i4.25649

[13] Gao Q, He Z, Jia X, Xie Y. Han X. Lightweight high-precision pedestrian tracking algorithm in complex occlusion scenarios. KSII Transactions on Internet and Information Systems (TIIS), 2023, 17(3): 840-860. https://doi.org/10.3837/tiis.2023.03.009

[14] Guo Z, Ma D, Luo X. A lightweight semantic segmentation algorithm integrating CA and ECA-Net modules. Optoelectronics Letters, 2024, 20(9): 568-576. https://doi.org/10.1007/s11801-024-3241-z

[15] Cui Z, Wang N, Su Y, Zhang W, Lan Y, Li A. ECANet: Enhanced context aggregation network for single image dehazing. Signal, Image and Video Processing, 2023, 17(2): 471-479. https://doi.org/10.1007/s11760-022-02252-w

[16] Chen W, Zou B, Huang C, Yang J, Li L, Liu J, Wang X. The defect detection of 3D-printed ceramic curved surface parts with low contrast based on deep learning. Ceramics International, 2023, 49(2): 2881-2893. https://doi.org/10.1016/j.ceramint.2022.09.272

[17] Feng S, Zhou H, Dong H. Using deep neural network with small dataset to predict material defects. Materials & Design, 2019, 162(10): 300-310. https://doi.org/10.1016/j.matdes.2018.11.060

[18] Žvirblis T, Pikšrys A, Bzinkowski D, Rucki M, Kilikevičius A, Kurasova O. Data augmentation for classification of multi-domain tension signals. Informatica, 2024, 35(4): 883-908. https://doi.org/10.15388/24-INFOR578

[19] Mena-Yedra R, López Redondo J, Pérez-Sánchez H, Martinez Ortigosa P. ALMERIA: Boosting pairwise molecular contrasts with scalable methods. Informatica, 2024, 35(3): 617-648. https://doi.org/10.15388/24-INFOR558

[20] Hua J, Li X, Liu J, Tang J, Rao, J, Deng H. A novel arrhythmia classification of electrocardiogram signal based on modified HRNet and ECA. Measurement Science and Technology, 2022, 33(6): 701-713. https://doi.org/10.1088/1361-6501/ac51a3

[21] Duan X, Sun Y, Wang J. ECA-UNet for coronary artery segmentation and three-dimensional reconstruction. Signal, Image and Video Processing, 2023, 17(3): 783-789. https://doi.org/10.1007/s11760-022-02288-y

https://doi.org/10.31449/inf.v49i19.7913 Informatica **49** (2025) 37–52 **37**

Improved GBRT Algorithm and Transfer Learning for Band Gap Regulation and Screening of Perovskite Materials

Xueshuang Deng*, Jiaojiao Chen
Electronic Information and Electrical College of Engineering, Shangluo University, Shangluo 726000, China
E-mail: slxy20251223dx@163.com
*Corresponding author

Keywords: transfer learning, GBRT, halogen perovskite, band gap modulation, screening for new materials

Received: December 27, 2024

As society continues to evolve, there is an increasing demand for energy. Solar energy is a clean and renewable alternative, but the photovoltaic conversion efficiency of calcite solar cells is low. Therefore, the study proposes a new material screening method for chalcocite using an improved gradient boosting regression tree (GRBT) model and migration learning. It adopts weighted averaging instead of the initial simple averaging to make complete use of the information between all the data, and at the same time introduces an adaptive step reduction to optimize the algorithm, which is fused with the support vector machine (SVM) algorithm is fused to construct a hybrid model, using hierarchical migration learning to divide the source domain data and train the model separately. Experiments showed that the astringent loss function values of the improved method were 0.115 and 0.160 lower than those of the GBRT algorithm and SVM, respectively. Moreover, and the root mean square errors and coefficients of determination for predicting the caliche band gap of the hybrid model were 0.017 and -2.2 and 0.023 and -3.7% lower than those of GBRT and SVM, respectively. The average pairwise decision error, root mean square error, and coefficient of determination of the improved transfer learning method were 0.0097, 0.0205, and -5.06% lower, respectively, than those of the ordinary method, and the running speed was 1.92 s faster than that of the ordinary method. The study screened out six halogenated bis-calcitonite new materials with band gap values in the range of [1.14-1.62] eV, and the formation energies were all below 0.05. It can be concluded that the improved method can effectively enhance the screening accuracy and speed of perovskite materials, and promote the high speed development of solar cells.

Povzetek: Članek se osredotoča na izboljšan algoritem Gradient Boosting Regression Tree (GBRT) in prenosno učenje za obvladovanje in iskanje novih perovskitnih materialov s prilagodljivimi širjenji pasovnih lukenj. Eksperimenti so pokazali, da izboljšani model znižuje napako napovedi in povečuje učinkovitost iskanja novih materialov, s poudarkom na halogeniranih perovskitih.

1 Introduction

With the continuous development of human society, people's demand for various energy sources is also increasing dramatically, and traditional fossil energy sources such as coal, oil, and natural gas are non-renewable resources, and their reserves are being depleted at an alarming rate [1]. According to the international energy agency (IEA), the world's fossil energy resources will last less than a century at the current rate of extraction [2]. At the same time, according to the United Nations report, there are 685 million people around the world who face the dilemma of not having electricity, and there is a short-term problem of energy. Furthermore, solar energy as an almost endless energy source, complete can become a key means to solve the human energy shortage [3]. However, ordinary solar power generation materials are usually monocrystalline or polycrystalline silicon materials, whose photoelectric conversion efficiency is about 25%, while its preparation environment requires high requirements, which cannot be promoted on a large scale [4]. As an important material for new solar cells, perovskite has attracted much attention. Its photoelectric conversion efficiency can reach 25.7%. Its preparation cost is lower than that of traditional silicon solar materials, and it is not restricted by the preparation environment [5]. However, its stability is poor, so there is a need to screen for more superior combinations of components among the perovskite fraction. The traditional experimental trial-and-error and first-principle calculations are too time-consuming and costly. Instead, computerized techniques can be used to quickly screen perovskite materials with suitable band gap and stability [6]. In an attempt to improve the model's prediction performance, Yang et al. developed a gradient boosting regression tree (GRBT) model based on the enhanced sparrow search algorithm to solve the GRBT algorithm improvement challenge. The sparrow search algorithm was improved by the model using chaotic sinusoidal mapping and Student t distribution mutation. The gradient-enhanced regression number model was then optimized using the revised algorithm to increase the model prediction accuracy (PA). In comparison to the unimproved model, experiments

showed that the improved model's correlation coefficient increased by 0.015 and its root mean square error (RMSE) decreased by 0.09 [7]. When data is few, Zhang et al. suggested a novel way to increase PA by adding missing values for gradient-enhanced regression numbers. The method generated different copies of the masked dataset according to the degree of missing data in the preprocessing stage and pre-interpolated them, and regressed the time residuals in the input stage to input the missing values into the masked dataset. According to experiments, the technique could successfully increase PA in the absence of data [8].

To address the screening problem of new perovskite materials, Diao et al. proposed a new high-throughput data calculation method in order to find out the efficient, stable and non-toxic combinations among many perovskite materials. The method introduced the data mining algorithm into the high-throughput calculation to investigate the photovoltaic conversion performance and stability of 42 materials. The experiments showed that 39 of the perovskite materials had high stability with tolerance coefficients between 0.8 and 1.1, and three ideal cell materials were chosen wrongly [9]. To increase the conversion efficiency of solar cells, Zhi et al. suggested a machine learning (ML)-based approach for perovskite molecular characterisation and conversion efficiency. The method constructed a ML model using relevant data of 19 materials. It was shown that hydrogen bond donors, hydrogen atoms, and octane-water partition coefficients were important features for the selection of perovskite materials [10]. Liu et al. proposed a new ML-based screening method in order to enhance the screening efficiency of perovskite passivation materials. The method mapped the relationship between conversion efficiency and interfacial passivation materials at the atomic level and utilized density flood theory for high-throughput prediction. The results of the experiments showed that the high-performance materials could successfully contribute to the significant passivation effect, and the method offered screening guidelines for interfacial materials at the atomic level interface [11]. Lai proposed a new high-throughput screening strategy in an attempt to enhance the flux and reduce the time cost of the new halide perovskite. To spatially encode particle size and composition, respectively, the approach combined defect-engineered anion-exchange techniques with evaporative crystallization polymer pen lithography (EC-PPL). To quickly create 3-particle libraries, it selectively changed the defect concentration of particular particles. Experiments demonstrated that this strategy was effective in faster screening of new materials [12]. Liu et al. proposed a new ML based model in an attempt to reduce the trial and error cost of double perovskite oxides. The model utilized the band gap data of 236 perovskite materials to form a dataset, and used features such as ionic radius to screen stable perovskite materials with suitable band gaps from a variety of candidate combinations. Experiments demonstrated that the model was able to effectively reduce the cost of trial and error for new materials [13].

In summary, existing studies have explored the improvement of the GRBT algorithm and the screening of new perovskite materials from various aspects, and have achieved certain results. However, the existing methods for screening halogenated chalcogenides continue to face challenges, including inadequate PA. The root-mean-square error in band gap prediction remains substantial, necessitating the reduction of this error to below 0.05 to enhance the accuracy of chalcogenide predictions. Therefore, the study proposes a perovskite screening model based on GBRT algorithm and transfer learning, which innovatively adopts weighted averaging instead of the initial simple averaging to utilize all the information between the data completely. Meanwhile, adaptive reduction step size is introduced to optimize the algorithm, which is fused with support vector machine (SVM) algorithm to construct the hybrid model. Hierarchical transfer learning is used to divide the source domain data (SDA) and train the model separately. The goal of study is to improve the screening accuracy and speed of perovskite new materials, promote the rapid development of solar power generation, and reduce carbon emissions.

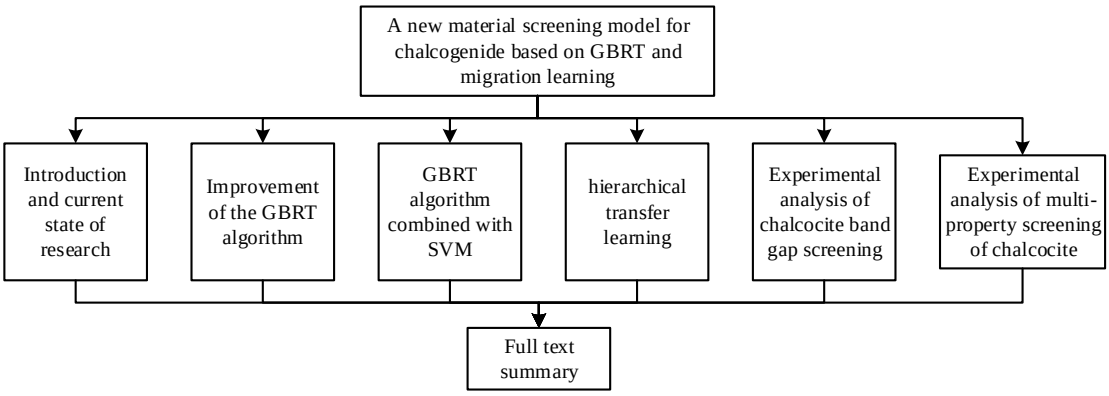
Based on the above related studies, Table 1 summarizes the research methodology, RMSE, calculation time, and shortcomings of the related studies.

Table 1: Summary of relevant information of relevant studies

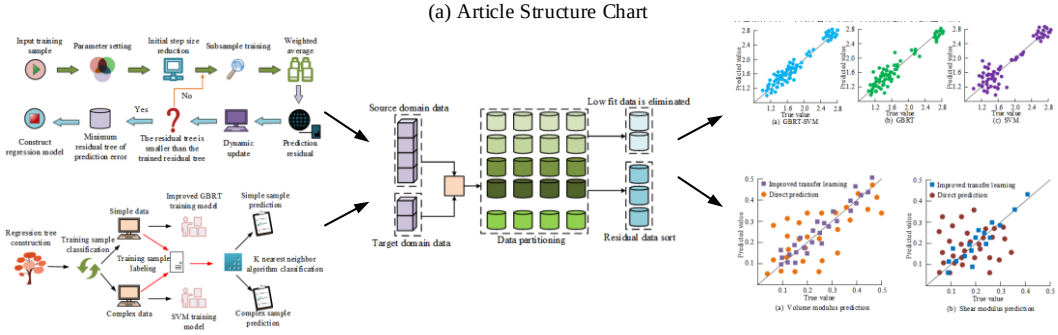
Author	Research methods	RMSE	Calculation time (ms)	Insufficient
Literature [7]	Improvement of GBRT based on sparrow search algorithm	0.072	97.5	Reduced computation speed when data volume is too large
Literature [8]	Pre-interpolation GBRT	0.066	86.2	Higher requirements for pre-interpolation
Literature [9]	High-throughput data computation	0.094	247.5	Higher hardware requirements
Literature [10]	Machine learning models	0.107	89.4	Requires more precise parameterization
Literature [11]	Machine learning and density functional theory	0.065	155.2	Requires high quality data for training
Literature [12]	EC-PPL and defect-engineered anion exchange techniques	0.058	586.4	Higher equipment requirements and more difficult to operate

Literature [13]	Machine learning models	0.079	58.3	Insufficient generalization ability when facing new materials
This text	GBRT and transfer learning	0.032	62.5	/

As illustrated in Table 1, while extant studies have attained specific outcomes in the context of chalcogenide material screening, their RMSE and computational efficiency performances remain unsatisfactory. Moreover, the experimental requirements are more stringent. Therefore, this study innovatively proposes a chalcogenide material screening method based on GBRT and migration learning, which effectively improves the screening speed and accuracy of new chalcogenide materials. To realize the screening and band gap regulation of new chalcogenide materials, the study proposes a chalcogenide screening model based on GBRT and migration learning. The general structure of which is shown in Figure 1.



(a) Article Structure Chart



(b) Article Flowchart

Figure 1: Overall architecture and flowchart of the article

In Figure 1(a), the study first explains the relevant research background, the current status of domestic and international research, and the innovation of the manuscript. The first chapter focuses on how to improve the GBRT algorithm and use it for chalcocite band gap screening. The second chapter focuses on the hierarchical migration learning of the model and the screening of chalcocite materials for multi-properties. The third chapter carries out a comprehensive test on the above methods. The fourth chapter carries out a full text summary as well as the future prospects. In Figure 1(b), the GBRT algorithm is selected as the fundamental prediction model, with the KNN algorithm employed to enhance it. The weighted average of the KNN classification algorithm is capable of emphasizing the correlation between variables, thereby enhancing the model's PA. Concurrently, the adaptive reduced step size optimization algorithm is implemented, which can employ a larger learning rate to expedite the identification of the optimal solution in the initial training phase and progressively diminish the learning rate in the subsequent phase to circumvent oscillatory behavior near the optimal solution or the occurrence of overfitting. The improved GBRT model is combined with the SVM model to improve the complex data processing capability and robustness of the hybrid model, and to achieve stable prediction results. The target domain data is input into the hierarchical migration model using a sliding window for training to obtain the initial training model, and the hierarchical migration can effectively reduce the computation time. The optimized GBRT-SVM hierarchical migration learning model first uses the low-level features of the data obtained by migration learning as input to the GBRT, then initializes the SVM with the final weights obtained by migration learning, and finally uses the features generated by the GBRT as input to the SVM for classification. Finally, the two models are used to screen the chalcogenide materials separately to obtain the best screening results.

2 Methods and materials

2.1 Improved GBRT algorithm for screening hybrid perovskite band gap

In solar cells, the band gap of perovskite materials is one of the direct influences on the photovoltaic conversion efficiency. The width of the band gap determines the wavelength range of light absorbed by perovskite materials, which in turn affects the photovoltaic conversion [14]. A hybrid perovskite is a heterogeneous material that combines organic and inorganic substances, which reacts when the two substances are interspersed, resulting in its unique properties. The general formula is ABX_3. The A site (A-S) is usually a cation with a large radius, such as Cs^+, Rb^+, and MA^+. The B site (B-S) is usually a relatively small radius cation, such as Pb^{2+}, Sn^{2+}, and Ge^{2+}. The X site (X-S) is for anions, commonly halogenated elements such as I., Cl., and Br.. Solar cells made of hybrid perovskite material have the advantages of simple process and low cost. Their photoelectric conversion efficiency is generally more than 20%, but there is still room for further improvement. The material components can be continuously adjusted to change its band gap width to enhance the photoelectric conversion efficiency [15]. Therefore, corresponding ML techniques are needed to screen perovskite materials with suitable band gaps, and the GBRT algorithm is selected for the study. Because GBRT can achieve high PA by integrating multiple weak learners, and when combined with SVM models, it can handle nonlinear relationships and complex data structures, making it suitable for scenarios with more features and complex relationships. Lower PA results from the platform GBRT algorithm's overly simplistic prediction function on the leaf node, which is also susceptible to changes in data quality. Therefore, by replacing the original simple average with the weighted average of the K-nearest neighbor (KNN) classification algorithm, the study enhances the technique and fully utilizes the information between all the data. The KNN algorithm first needs to calculate the distance from the training sample to the predicted sample in each node. A sample is selected which is closer to the predicted sample and the output variable of the sample is shown in Equation (1) [16].

$$\left(f_1, f_2, ..., f_{x-1}, f_x \right) \tag{1}$$

In Equation (1), f_x denotes the output variable of the x th training sample. The weights calculated for each sample weighting are shown in Equation (2).

$$w_i = \left(e^{\frac{d_i^2}{2}} \right) \Big/ e^{-\frac{d_1^2}{2}} + e^{-\frac{d_2^2}{2}} + ... + e^{-\frac{d_i^2}{2}} \tag{2}$$

In Equation (2), w_i is the weight of the training sample. d_i is the distance from the training sample to the predicted sample. The weighted average is calculated as shown in Equation (3).

$$W = \sum_{i=1}^{x} w_i f_i \tag{3}$$

In Equation (3), W denotes the weighted average of the samples. The study choose Equations (1)-(3) to compute the weighted average of the KNN algorithm instead of the simple average, which improves the utilization of data and is suitable for initial data processing. The K value of the KNN algorithm by plotting the classification error rate curve corresponding to different K values, the error rate in a certain range of K values will first decrease and then increase, when the lowest point corresponding to the K value is the best choice. The study uses the weighted average of the KNN classification algorithm to be able to focus on the correlation between variables and effectively enhance the PA of the model. The reduction step size of GBRT algorithm, i.e., the learning rate, is usually kept constant, which increases the risk of overfitting of the algorithm, as well as reduces the training efficiency and so on. The study introduces an adaptive shrinkage step to optimize the algorithm. Firstly, Equation (4) illustrates how the algorithm's loss function (LF) is defined [17].

$$L = \sum_{i=1}^{n} \frac{1}{N} \left[f_i - \left(H_j(x_i) + \lambda h_{j+1}(x_i) \right) \right]^2 \tag{4}$$

In Equation (4), L denotes the LF of the algorithm. N denotes the number of random subsamples. $H_j(x_i)$ denotes the strong learner consisting of the first j residual trees. λ denotes the reduced step size. $h_{j+1}(x_i)$ denotes the weak learner of the $j+1$ th residual tree. When the strong and weak learners are deterministic values, the LF is used to derive the reduced step size, as shown in Equation (5).

$$\frac{\partial L}{\partial \lambda} = -\frac{2}{N} \sum_{i=1}^{n} \left[f_i - \left(H_j(x_i) + \lambda h_{j+1}(x_i) \right) \right] \cdot h_{j+1}(x_i) \tag{5}$$

In Equation (5), the value of the adaptive reduction step can be computed using Equation (6) when the derivative equals zero.

$$\lambda = \frac{\sum_{i=1}^{n} \left[f_i - H_j(x_i) \right] \cdot h_{j+1}(x_i)}{\sum_{i=1}^{n} h_{j+1}^2(x_i)} \tag{6}$$

Adaptive scaling of the step size allows a larger learning rate to be used in the early stages of training to quickly solve the optimal solution, and a progressively smaller learning rate in the later stages to avoid oscillation around the optimal solution or overfitting. The study

selects Equations (4)-(6) to improve the GBRT algorithm with adaptive reduced step size, which can improve the training efficiency and reduce the risk of overfitting of the algorithm, and is applicable to the training process of the GBRT algorithm. The specific operation flow of GBRT algorithm improved with KNN is shown in Figure 2.

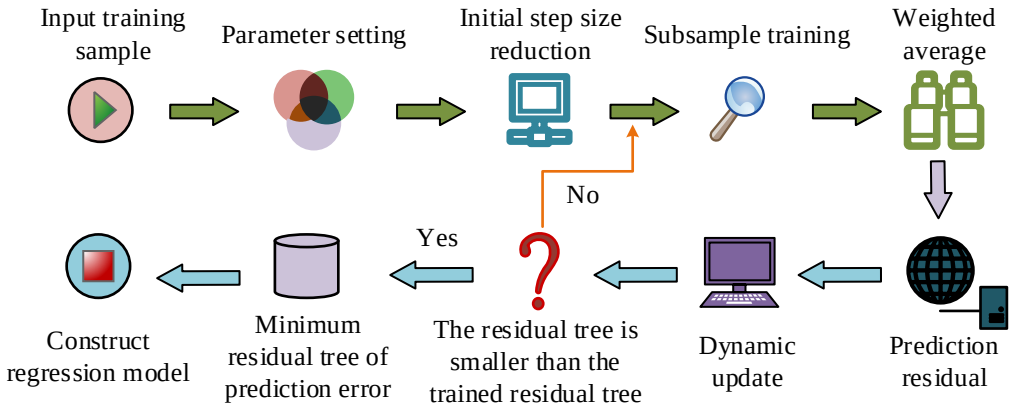


Figure 2: Specific operation flow of improved GBRT algorithm

In Figure 2, the training samples are entered first, the parameters such as the number of training times and the random sampling rate of the residual tree are set, and the initial reduction step is set to 0.01. The smaller reduction step modulates the model by only a small amount each time, which can effectively prevent the model from overfitting during training. Relevant studies have shown that although a larger reduction step can converge quickly, it is only applicable to simple data sets. Moreover, in the screening of chalcocite materials, the accuracy of the model with a reduction step of 0.01 is usually greater than that with a reduction step of 0.1, and the samples are initialized after the reduction step setting is completed. A sub-sample is randomly selected and trained on the residual tree, and a weighted average is used to find the prediction function. The output variable values are updated using the predicted residuals, and the size of the reduction step is dynamically updated. When the number of remaining residual trees is smaller than the number of trained residual trees, the residual tree with the smallest prediction error is extracted. This residual tree and all previous residual trees are used to form a regression model. To address the modeling problem when the relevant sample data is insufficient and to enhance the model PA, the research adopts the fusion of SVM algorithm and GBRT algorithm to construct the hybrid model. This approach effectively improves the complex data processing ability and robustness of the hybrid model and provides stable prediction results through the stronger generalization ability and high-dimensional data processing ability of SVM. The SVM algorithm, through the nonlinear mapping function, maps the low-dimensional. The SVM algorithm maps the low-dimensional training sample data into the high-dimensional space by means of a nonlinear mapping function, and performs linear regression of the data in the high-dimensional space, and the regression constructor of the SVM is shown in equation (7).

$$f(x)=\omega\varphi(x)+b \tag{7}$$

In Equation (7), $f(x)$ denotes the regression constructor. ω is the weight vector. $\varphi(x)$ denotes the NMF. S denotes the bias term. The weight vector and bias term are solved as shown in Equation (8).

$$\min_{\omega,b}\frac{1}{2}\|\omega\|^2+C\sum_{i+1}^{n}\left(\zeta+\zeta^*\right) \tag{8}$$

In Equation (8), C denotes the regularization constant. Both ζ and ζ^* denote slack variables. Equation (9) is used to determine the solution high spatial data.

$$f(X)=\sum\left(q_i-q_i^*\right)\kappa\left(X_i,X_j\right)+b \tag{9}$$

In Equation (9), both q_i and q_i^* denote Lagrange multipliers. $\kappa\left(X_i,X_j\right)$ is the kernel function (KF). The selection of the kernel function is contingent upon the particular circumstances of each case. This study posits that the radial basis kernel function requires a smaller number of parameters and is adept at effectively addressing nonlinear, differentiable problems by mapping the data points of the input feature. Space into an infinite-dimensional feature space. At the same time, the kernel function has a wider scope of applicability, and it can make the SVM model have a wide range of accuracy, so it is selected as the kernel function of SVM. Which is calculated as shown in Equation (10).

$$\kappa\left(x_i,x_j\right)=\exp\left(-\sigma\left\|x_i-x_j\right\|^2\right) \tag{10}$$

In Equation (10), $\kappa\left(x_i,x_j\right)$ denotes the radial basis KF. σ denotes the width of the radial basis kernel.

$\|x_i - x_j\|$ denotes the Euclidean distance between two points. The study selects Equations (7)-(10) for SVM and GBRT fusion, which can improve the model's high-dimensional data processing ability and is suitable for the analysis of complex data. Figure 3 depicts the hybrid model's operational flow.

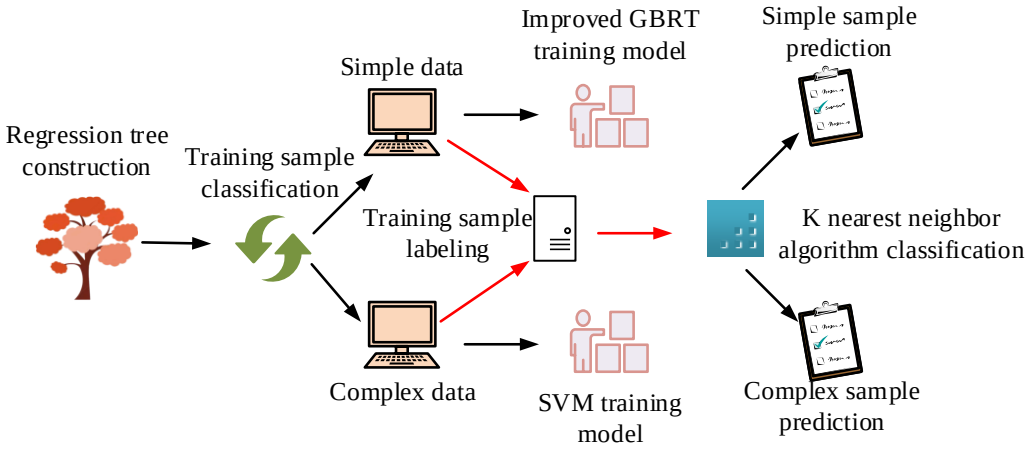


Figure 3: Operation flow of the hybrid model

In Figure 3, after constructing the regression tree using the GBRT algorithm, the training samples are divided into simple data and complex data based on the prediction deviation of the leaf nodes. The first 1/2 leaf nodes with large prediction deviation are categorized as complex data, and the remaining 1/2 leaf nodes are categorized as simple data. In the study, the simple data is input to the GBRT model for training, and the complex data is input to the SVM model for training. After labeling the two training data with categories, the test samples are classified using the classification algorithm based on K-nearest neighbor algorithm, which classifies the test samples into simple and complex data, and then the two models are used to predict the test samples after the training is completed.

2.2 Multi-property screening of halogen double perovskite based on hierarchical transfer learning

Despite the advantages of low manufacturing cost and high photovoltaic conversion efficiency, the presence of heavy metals such as lead in their composition renders heterogeneous chalcogenide materials environmentally and biotoxic. Leakage of these materials can result in soil and water contamination, thereby affecting the ecosystem and human health. Adverse effects may include neurological damage, reproductive toxicity, and hematologic disorders. Therefore, there is a need to screen non-toxic and environmentally friendly chalcogenide materials [18]. Halogen double perovskite is an effective alternative to lead-based perovskite materials due to its lead-free nature and high stability and dimmability. The general formula for the structure of halogen double perovskite is $A_2BB'X_6$. Among them, the A-S is an inorganic cation. The B-S is a monovalent metal ion (MI), the B' site is a trivalent MI, and the X-S is a halogen ion (HI). The B' site is a trivalent MI and the X-S is a HI. The X-S is a HI. The octahedral factor of $A_2BB'X_6$ is calculated as shown in Equation (11).

$$\mu = \frac{r_B}{r_X} \tag{11}$$

In Equation (11), μ denotes octahedral factor. r_B denotes radius of B-S ion. r_X is the radius of the X-S HI. The calculation of tolerance factor is shown in Equation (12).

$$TF = \frac{\left(r_A + r_X\right)}{\sqrt{2}\left(r_B + r_X\right)} \tag{12}$$

In Equation (12), TF denotes the tolerance factor. r_A denotes the radius of the A-S ion. Equations (11) and (12) are chosen for the study to evaluate the stability in chalcogenide structures. However, the available data samples of $A_2BB'X_6$ materials are small, and most of the ML methods are unable to obtain the prediction results with high accuracy in the limited data. Therefore, the study uses transfer learning algorithm for $A_2BB'X_6$ material screening. Transfer learning can improve the model's generalization capacity, lessen reliance on a lot of labeled data, and adapt to the target domain by using the knowledge of the source domain. Figure 4 illustrates the fundamental architecture of transfer learning in artificial neural networks.

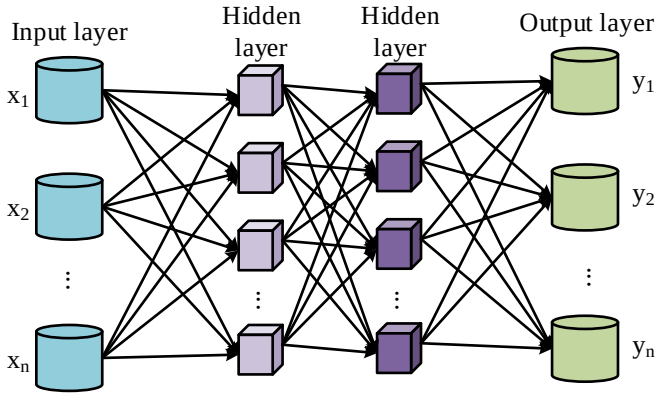


Figure 4: Basic structure of transfer learning neural network

In Figure 4, the neural network is divided into five layers in total, which are the input layer, two hidden layers, one output layer, and one error calculation layer. Equation (11) is used to connect the data in the input layer amongst all of the neurons in each layer, and the two hidden layers are used to extract the input data's features layer by layer. Then, the nonlinear inertia in the data is learned and expressed by the activation function. Moreover, the study selects the ReLU activation function that possesses computational simplicity, can avoid gradient vanishing, as well as performs well in deep networks. Finally, it converts the data into predictions through the output layer. However, the ordinary transfer learning algorithm needs to be carried out step by step many times, which takes a long time. Therefore, the algorithm is improved hierarchically. The hierarchical transfer learning firstly needs to divide the SDA, calculate the fit of the SDA and sort it according to the size. The target domain data (TDA) is input into the GBRT-SVM model using a sliding window for training, and the initial training model is obtained. The SDA partitioning process for hierarchical transfer learning is shown in Figure 5.

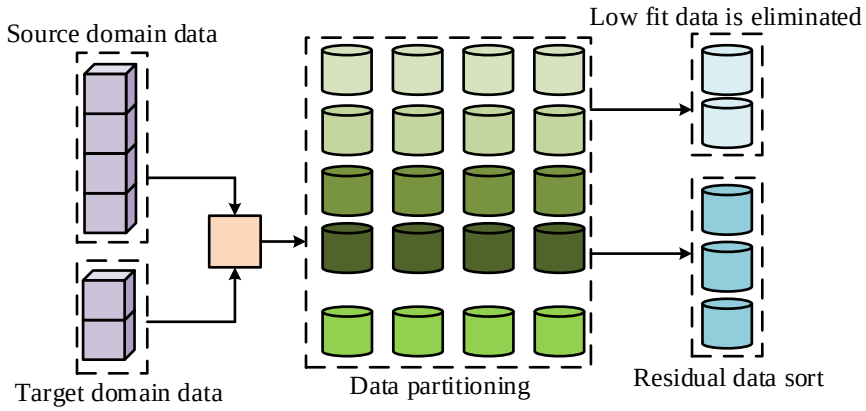


Figure 5: Source domain data partitioning process of hierarchical transfer learning

In Figure 5, the study also used a sliding window to input the SDA into the model. Among them, a single prediction is available in each window, and R^2, the goodness of fit, is used to indicate the degree of correctness of the prediction results. The study sets the fit threshold at 0.7, removes data with a fit less than the threshold, and uses the size of the R^2 to rank the data, with the higher the fit the higher the ranking. The study is based on the number of layers of the hierarchical migration model. The SDA that meets the fit requirements is divided into a total of K groups of data, the K1 group of data with a lower fit is used to train the K1 layer of the model. The weights of the training results are recorded, the rest of the layer is frozen, and the K2 group of data is used to train the K2 layer of the model. Moreover, the K2 layer of the training is used to load the previous layer of the weights and to determine the loss of prediction results in the target dataset in whether or not to decline. If it decreases, unfreeze the current layer and all previous layers, and vice versa, continue training until all layers of the model are all unfrozen. The prediction effect of each layer is judged and the weight of the layer with the largest increase in prediction effect is recorded. The operation flow of hierarchical transfer learning is shown in Figure 6.

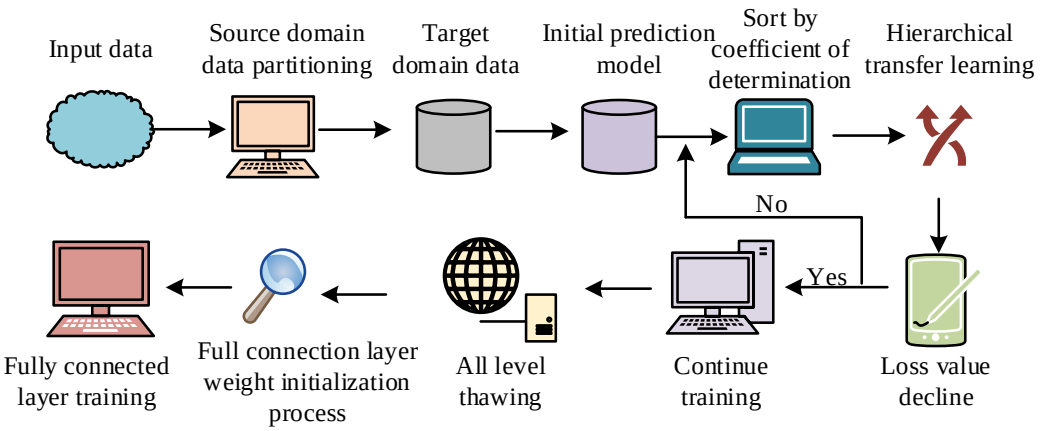


Figure 6: Operation flow of hierarchical transfer learning

In Figure 6, the input data are SDA, TDA, and GBRT-SVM hybrid model. The study first divides the SDA, trains the TDA in the model to get the initial prediction model, and sorts the SDA according to the coefficient of determination and divides them into K groups. The study uses the data of the division number for hierarchical transfer learning, using the loss value as a judgment criterion for the prediction effect. The training is continued if the loss value decreases until all layers of the model are fully unfrozen. After the model transfer learning is completed, the study uses the TDA to adjust the model parameters, initialize the final fully connected layer weights, and set a smaller learning rate to train only the fully connected layer. The optimized GBRT-SVM model with hierarchical migration learning first uses the low-level features of the data obtained from migration learning as inputs to the GBRT, then initializes the SVM with the final weights obtained from migration learning, and finally uses the feature inputs generated by the GBRT in the SVM for classification. The study, in an effort to deduce the complexity of the model and the occurrence of overfitting situations, uses a feature subset-based approach for feature dimensionality reduction after comprehensive consideration. The method utilizes the Pearson coefficient between different feature subsets to reduce the linear feature correlation. The Pearson coefficient is calculated as shown in Equation (13) [19].

$$r = \frac{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)\left(Y_i - \bar{Y}\right)}{\sqrt{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2}\sqrt{\sum_{i=1}^{n}\left(Y_i - \bar{Y}\right)^2}} \quad (13)$$

In Equation (13), r denotes the Pearson coefficient. X_i is the i th observation of variable (OV) X. $\bar{X}$ is the mean of variable (MV) X. Y_i is the i th OV Y. $\bar{Y}$ is the MV Y. n is the total quantity of observations. The Pearson's coefficient takes values between [-1, 1], when it takes a value of 1, it indicates a fully positive linear correlation. When it takes a value of -1, it indicates a fully negative linear correlation, and when it takes a value of 0, it indicates no linear relationship. The study calculates the Pearson coefficient of each characteristic with respect to the target variable and retains the characteristics with high linear correlation. The study retains features with Pearson coefficients greater than 0.5 as a subset of features, and feature dimensionality can be reduced. The data are normalized to ensure comparability between data magnitudes. The study uses the normalization method to convert the initial data to standard data, as calculated in Equation (14) [20].

$$X_{std} = \frac{X - \mu}{\gamma} \quad (14)$$

In Equation (14), X_{std} denotes the normalized feature data (FD). X denotes the initial data. μ the mean value of the FD. γ is the standard deviation of the FD. After the screening of the materials is completed, the study employs the formation energy (FE) to validate the materials with first-principle calculations. The relevant parameters of the materials are the same as the model in the training set. The FE of a perovskite compound is the energy released or absorbed by an atom to form a compound from a free state. The FE of a compound is calculated as shown in Equation (15).

$$E_b = \frac{E\left(A_mB_nC_k\right) - m\cdot E(A) - n\cdot E(B) - k\cdot E(C)}{m+n+k} \quad (15)$$

In Equation (15), E_b is the FE of the compound. $E\left(A_mB_nC_k\right)$ denotes the total energy of the compound. $E(A)$, $E(B)$, and $E(C)$ denotes the energy of the free atom A, B, and C, respectively. m, n, and k denotes the number of atoms A, B, and C, respectively. The study selects Equations (13)-(15) for data feature reduction and normalization, which can reduce the complexity of the model, ensure the comparability between data magnitudes, and apply to data processing of different dimensions. The pseudo-code for the improved GBRT-SVM hybrid model is shown in Figure 7.

```
# Import necessary libraries
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, accuracy_score

# Assume the dataset is already prepared
# X: Feature matrix
# y: Target variable
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 1: Feature extraction using GBRT
gbrt = GradientBoostingRegressor(n_estimators=100, learning_rate=0.1, max_depth=3, random_state=42)
gbrt.fit(X_train, y_train)

# Extract features from the training and testing sets using the GBRT model
X_train_features = gbrt.apply(X_train).reshape(X_train.shape[0], -1)
X_test_features = gbrt.apply(X_test).reshape(X_test.shape[0], -1)

# Step 2: Classification or regression using SVM
svm = SVC(kernel='rbf', gamma='scale', C=1.0)  # For classification tasks
# For regression tasks, use SVR instead
# svm = SVR(kernel='rbf', gamma='scale', C=1.0)

# Train the SVM model
svm.fit(X_train_features, y_train)

# Step 3: Evaluate model performance
# For classification tasks
y_pred = svm.predict(X_test_features)
accuracy = accuracy_score(y_test, y_pred)
print(f"Test set accuracy: {accuracy:.4f}")

# For regression tasks
# y_pred = svm.predict(X_test_features)
# rmse = mean_squared_error(y_test, y_pred, squared=False)
# print(f"Test set RMSE: {rmse:.4f}")
```

Figure 7: Pseudo-code for improved GBRT-SVM hybrid modeling

3 Results

3.1 Experimental analysis of screening of hybrid perovskite band gap

In order to verify the prediction effect of the GBRT-SVM hybrid model, the study adopts the Perovskite Database, which contains more than 16,000 chalcocite-related thesis data, covering more than 42,400 chalcocite materials with detailed information, which is of high quality and good reliability, and the database can effectively simplify the process of literature search and data analysis to improve the experimental efficiency. The database can effectively simplify the process of literature search and data analysis, and improve the experimental efficiency. Table 2 Advantages of the two datasets used in the experiment over other datasets.

Table 2: Advantages of the two datasets used in the experiment over other datasets

Characterization	Perovskite Database	Materials Project
Scale	Covering more than 42,400 chalcogenide materials	Over 1 million inorganic materials
Type	To include the structure, properties and other data of chalcogenide materials	Provide crystal structure, energy properties, electronic structure, thermodynamic properties, etc.
Update Frequency	Continuously updated	The data is continuously updated to keep it up-to-date
Openness	Open Data	Open Data
Collaboration	Data sharing	Encourage user collaboration and cooperative research
User Interface	Interactive graphical interface, easy to use	User-friendly interface, easy to search and browse

In Table 2, the two datasets selected for the experiment possess greater advantages in terms of data size, data type, and openness, and the data are well-targeted and easy to find. The filtered, dimensionality

reduced, and normalized data are divided into training set, validation set, and test set in 8:1:1 manner. The penalty parameter of the SVM model is set to 0.1 and the kernel function parameter is set to 0.001. The comparison algorithms used in the study include SVM and GBRT algorithms. Figure 8 compares the variations in loss values of several algorithms.

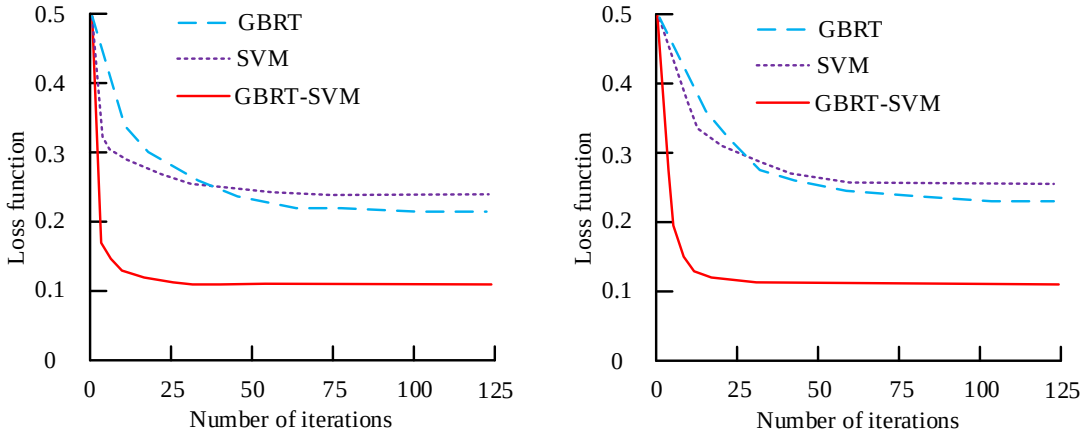


Figure 8: Comparison of loss value changes of different algorithms

In Figure 8(a), the LF of the GBRT-SVM hybrid model decreases faster in the training set and is able to obtain the astringent LF value in 30 iterations. The minimum value is 0.108, which is 0.115 and 0.160 lower than the GBRT and SVM algorithms, respectively. The optimization speed is faster than the other two algorithms by 35 and 20 iterations, respectively. In Figure 8(b), the astringent loss function value of all three algorithms increases when faced with the test set data. However, the GBRT-SVM hybrid model has the smallest increase in loss value, with an astringent loss function value of 0.125, which is lower than the other two algorithms by 0.139 and 0.173, and its increase is lower than the other two algorithms by 0.024 and 0.013, respectively. The optimization speed of the hybrid model remains basically unchanged, while the other algorithms have decreased. Comparison of perovskite band gap prediction results of different algorithms is shown in Figure 9.

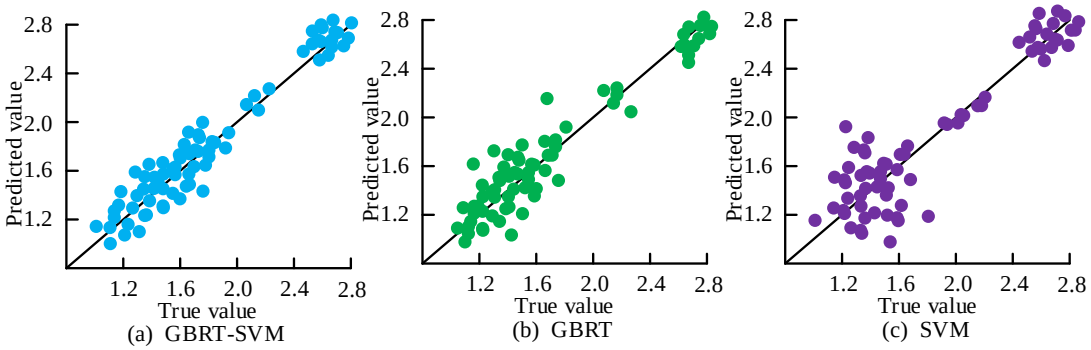


Figure 9: Comparison of prediction results of perovskite band gap by different algorithms

In Figure 9(a), the RMSE and coefficient of determination of the prediction of chalcocite band gaps of the GBRT-SVM hybrid model are 0.032 and 99.4%, respectively. Furthermore, the predicted values all converge to the vicinity of the diagonal of the horizontal and vertical coordinates, and there are no prediction points with large deviations. The more the model's predicted values dissolve the diagonal line, the closer the predicted values are to the true values and the higher the model's PA. In Figure 9(b), there are some points with large prediction deviations between the true band gap values [1.2, 1.8], and the RMSEand coefficient of determination of the GBRT model are 0.049 and 97.2%, respectively. In Figure 9(c), there are points that deviate from the convergence diagonal for band gap values below 2.0, and the RMSE and coefficient of determination of the SVM model are 0.055 and 95.7%, respectively. UV-Vis diffuse reflectance spectroscopy is used to determine the band gap of chalcogenide materials using a UV-Vis spectrophotometer and an integrating sphere attachment. A chalcogenide sample is first prepared as a thin film and placed in an integrating sphere where the reflectance spectra in the UV-Vis region are recorded and then converted to absorption spectra by the Kubelka-Munk function to measure the band gap value of the sample. Table 3 displays the hybrid model's band gap prediction results for various hybrid perovskite samples.

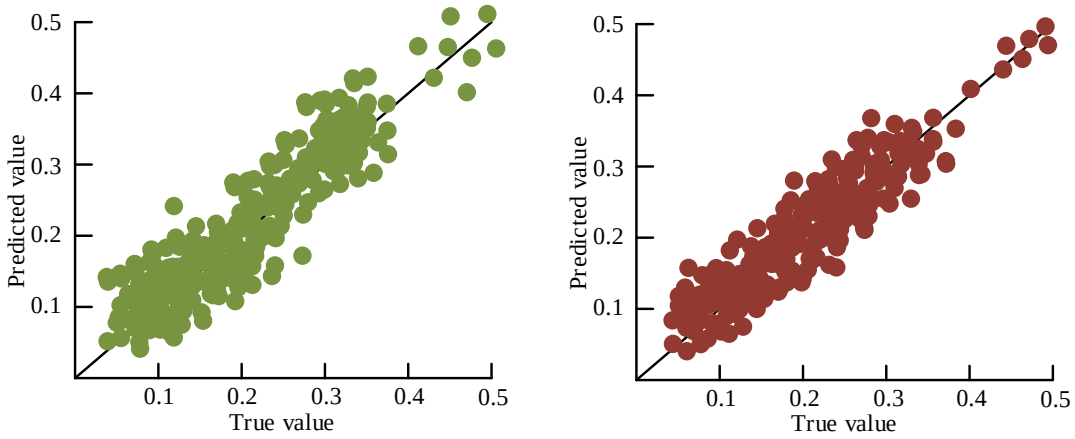
Table 3: Band gap prediction results of mixed models for different hybrid perovskite samples

Serial number	Pbe_band gap	Ml_band gap (eV)	Prediction error
1	1.534	1.520	0.014
2	0.015	0.004	0.011
3	0.000	0.013	-0.013
4	2.745	2.713	0.032
5	1.036	1.107	-0.071
6	0.000	0.005	-0.005
7	1.527	1.535	-0.008

In Table 3, the band gap prediction values of GBRT-SVM hybrid model for hybrid perovskite are all between [0.004, 2.713]. The highest and lowest value of the prediction error is 0.071 and 0.005, which both satisfy the requirement that the error range is less than 0.1 eV. Among them, there are only two that satisfy the band gap requirement of solar cell materials, which are serial numbers 1 and 7, with BGVs in the range of 1.1eV-1.7eV. The prediction errors of the hybrid model for sample #1 are 0.031 and 0.28 lower than those of the GBRT and SVM models, respectively, and 0.004 and -0.0042 lower for sample #2.

3.2 Experimental analysis of multi-property screening of halogen double perovskite

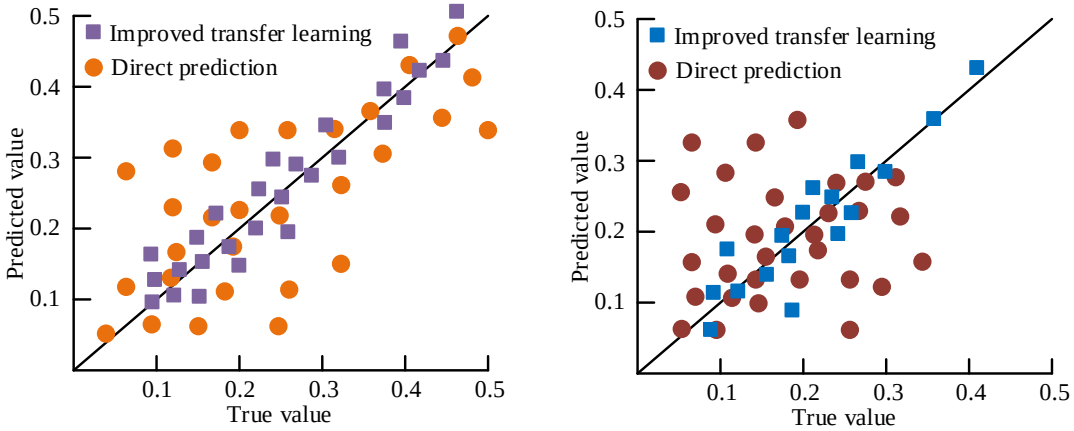
The inorganic halogen double perovskite dataset used for the experiments is Materials Project. Other parameters and treatments are the same as in Section 2.1, and the transfer learning model is used to compare the results of FE, bulk modulus, and shear modulus predictions of perovskite materials, respectively. Figure 10 compares the FE prediction outcomes of perovskite materials using various techniques.



(a) Transfer learning (b) Improved transfer learning

Figure 10: Comparison of prediction results of perovskite material formation by different methods

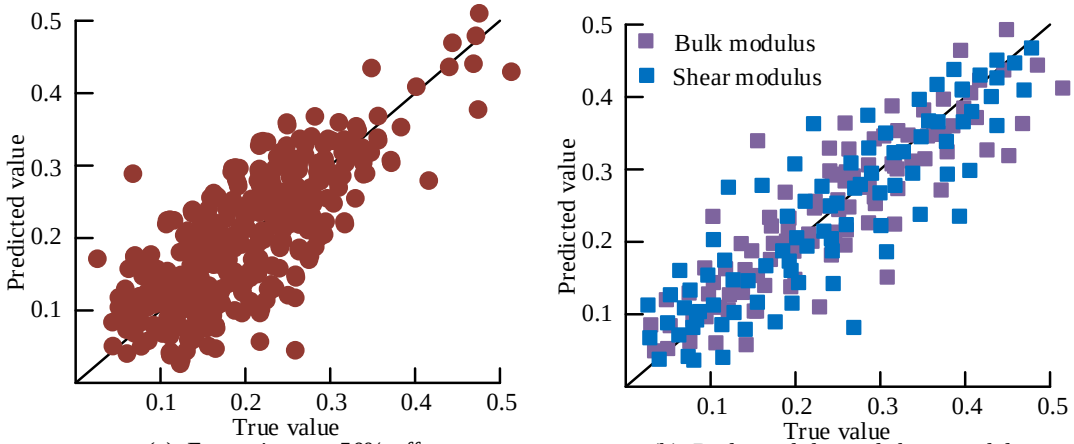
In Figure 10(a), the ordinary transfer learning method can achieve a certain PA for the FE prediction of perovskite. Its average decision error, RMSE, and coefficient of determination are 0.0472, 0.0609, and 94.35%, respectively. However, transfer learning takes a long time, running once for 2.89 s. In Figure 10(b), the average decision error, RMSE, and coefficient of determination of the improved transfer learning method are lower than that of the ordinary method by 0.0097, 0.0105, and -5.06%, respectively, and the running speed is faster than that of the ordinary method by 1.92 s. Figure 11 compares the modulus and shear modulus prediction findings of various techniques for perovskite materials.



(a) Volume modulus prediction (b) Shear modulus prediction

Figure 11: Comparison of prediction results of bulk modulus and shear modulus of perovskite materials by different methods

In Figure 11(a), the halogen double perovskite material has less relevant data and the feature dimension reaches 132 dimensions, which leads to lower PA using direct prediction methods. The RMSE and coefficient of determination are 0.092 and 87.25%, respectively. The RMSE of the bulk modulus measured using the improved transfer learning method is 0.037 lower and the coefficient of determination is 11.18% higher than the direct prediction. The prediction results of ordinary transfer learning methods are more scattered and further away from the diagonal, indicating that the predicted values are less consistent with the true values and their PA is lower. In Figure 11(b), the RMSE and coefficient of determination of shear modulus prediction by the improved transfer learning method are 0.057 and 98.65%, which are 0.042 and -12.36% lower than the direct prediction, respectively. To quantify the uncertainty of the model, the study uses 5-fold cross-validation for model evaluation. Among them, the experimental data set is randomly divided into five mutually exclusive subsets of equal size, each of which is an independent test set. The results of the 5-fold cross-validation of the formation energy and modulus of the material are shown in Figure 12.



(a) Formation can 50% off cross validation results (b) Body modulus and shear modulus 5 fold cross validation results

Figure 12: Formation energy and modulus of materials 5-fold cross validation results

In Figure 12(a), the RMSE of the improved transfer learning method after 5-fold cross-validation is 0.0417. It is 0.0013 higher than the validation result of the independent test set, but still less than 0.05, which meets the relevant requirements. In Figure 12(b), the RMSE of body modulus after 5-fold cross-validation is 0.057, and the RMSE of shear modulus is 0.059, which is higher than that of the independent test set. However, both are within the acceptable range, indicating that the model is more reliable. A comparison of the importance of component features on perovskite materials modulus and shear modulus is shown in Figure 13.

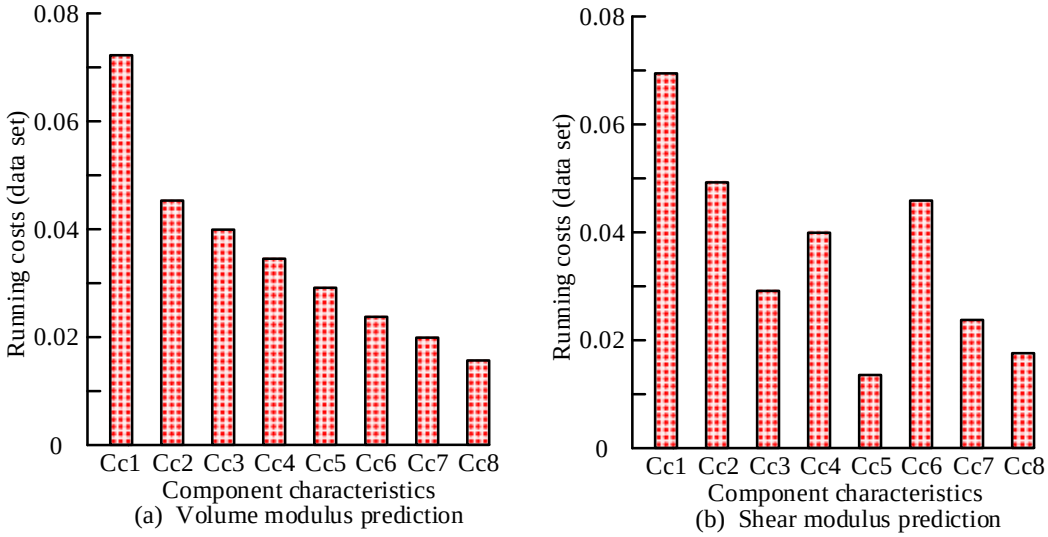


Figure 13: Comparison of the importance of component characteristics to the bulk modulus and shear modulus of perovskite materials

In Figure 13(a), the factors affecting the material's importance from Cc1 to Cc8 are Mendeleev number, electronegativity, ionic radius, octahedral factor, p-orbital valence electrons, total valence electrons, tolerance factor, and s-orbital valence electrons, respectively. Among them. The greatest degree of influence on the modulus of perovskite materials is the Mendeleev number. The Mendeleev number is most important because of its ability to quickly identify combinations of materials with characteristic properties, and electronegativity because of its role in the formation and stability of chemical bonds in materials. In Figure 13(b), the degree of influence on the shear modulus of the materials above 0.04 includes Mendeleev number, electronegativity, octahedral factor, total valence electrons. The study screens new chalcogenide materials with compliant elemental compositions based on the thermal stability, band gap value, and B/G ratio of the materials. The screened new halogenated bis-chalcogenide materials that meet the relevant conditions are shown in Table 4.

Table 4: Comparison of properties of halogen double perovskite new materials

Element combination	Band gap value	Formation energy	Bulk modulus	Shear modulus	B/G
Cs_2AuNiF_6	1.52	0.042	52.64	6.87	7.66
K_2CuInF_6	1.14	0.018	66.51	9.95	6.68
Cs_2AuOsF_6	1.55	0.049	60.75	11.09	5.48
$K_2InMoCl_6$	1.25	0.037	27.95	4.87	5.74
Cs_2CuRhF_6	1.62	0.054	79.27	15.93	4.98
K_2InCrF_6	1.44	0.012	70.25	14.98	4.69
Rb_2InRuF_6	1.21	0.009	61.54	16.89	3.64

In Table 4, the BGVs of the seven halogen double perovskite new materials are between [1.14-1.62] eV, while the optimum photoelectric conversion efficiency of perovskite solar cells is around 1.4 eV, and all of them can be used for solar cells at 1.1-1.7 eV. Formation energies below 0.05 all meet the stability requirements of the material. Except for Cs_2CuRhF_6, the other six materials meet the stability requirements. The ratios of shear modulus to bulk modulus of the new materials are all greater than 1.75, indicating that the new materials have high toughness and can be obtained with good ductility at room temperature. The comparison results of ablation experiments with different adaptive algorithms and improved migration learning methods are shown in Table 5.

Table 5: Comparison results of ablation experiments with different adaptive algorithms and improved migration learning methods

Algorithms	Ablation Module	RMSE	Coefficient of determination (%)	Predictive accuracy (%)
Adaptive GBRT-SVM	/	0.032	99.4	85.2
Adaptive SVM	/	0.057	94.8	73.5
Adaptive RF	/	0.074	90.3	69.8
Improved migration learning	/	0.040	99.4	84.3
	Normalization	0.073	92.6	79.2
	Feature dimensionality reduction	0.056	95.7	80.6

In Table 5, the adaptive GBRT-SVM achieves the optimal values for all metrics, and its RMSE is lower than that of the adaptive SVM and adaptive random forest (RF) by 0.025 and 0.042, respectively. Moreover, the coefficients of determination are higher than those of the two methods by 4.4% and 9.1%, respectively. For the ablation experiment of the improved transfer learning algorithm, after removing the normalization processing and feature dimensionality reduction module, the performance of the algorithm decreases in all aspects. Its RMSE increases by 0.033 and 0.016, and the coefficient of determination decreases by 6.8% and 3.7%, respectively. It indicates that the normalization processing has a greater impact on the performance of the model, and the feature dimensionality reduction mainly affects the computation speed of the model. The relevant terms and variables used in the manuscript are shown in Table 6.

Table 6: Manuscript-related variables and their interpretation

Serial number	Term	Detailed Information
1	GRBT	Gradient Boosting Regression Tree
2	EC-PPL	Evaporation Crystallization-Polymer Pen Lithography
3	SVM	Supported Vector Machine
4	KNN	K-NearestNeighbor
5	B/G	Ratio of shear modulus to bulk modulus
6	f_x	Output variables of the training samples
7	w_i	Weights of the training samples
8	d_i	Distance from training samples to predicted samples
9	W	Weighted average of samples
10	$H_j(x_i)$	j Strong learners consisting of residual trees
11	$h_{j+1}(x_i)$	Weak Learner for $j+1$ Residual Trees
12	λ	step-down
13	$f(x)$	regression constructor
14	$\varphi(x)$	nonlinear mapping function
15	b	bias entry (computing)
16	C	Regularization constants
17	ζ and ζ^*	slack variable (math)
18	q_i and q_i^*	Lagrange multiplier (math)
19	$\kappa(X_i, X_j)$	kernel function (math)
20	$\kappa(x_i, x_j)$	radial basis kernel function (math)
21	σ	Width of radial base core
22	$\|x_i - x_j\|$	Euclidean distance between two points
23	μ	octahedral factor
24	r_B	The radius of the B-site ion
25	r_X	The radius of the X halogen ion
26	TF	Tolerance factor
27	r_A	The radius of the A-position ion
28	r	Pearson coefficient
29	X_{std}	The feature data after normalization
30	E_b	The formation energy of the compound
31	$E(A_mB_nC_k)$	Total energy of the compound
32	$E(A)$	The energy of the free atom A
33	$E(B)$	The energy of the free atom B
34	$E(C)$	The energy of the free atom C

4 Discussion

A band gap prediction model for chalcocite materials based on the improved GBRT algorithm and migration learning was proposed and applied to the actual analysis of chalcocite samples, and the validity of the prediction model was verified by relevant experimental analysis. The GBRT-SVM hybrid model converged faster in both the training and test sets, and the final convergence value was smaller than that of the other methods, as the adaptive reduction of the step size could effectively improve the pre-optimization speed and post-optimization accuracy of the model. Compared with the improved methods of Yang [7] and Zhang [8], the proposed method could significantly improve the computational speed while guaranteeing the PA. The RMSE and coefficient of determination of the band gap prediction of the hybrid model were better than those of the basic method, and the consistency between the predicted value and the true value was higher, with higher PA, compared with the RMSE model of Zhi [10]. In cross-validation, the RMSE of the improved migration learning method was 0.0417, which was 0.0013 higher than that of the independent test set validation results. However, it still less than 0.05, which could meet the relevant requirements, and the proposed method could effectively improve the computational efficiency and reduce the cost of screening new materials compared with Liu [13]. The GBRT-SVM hybrid model, through its powerful classification and regression analysis capabilities, could effectively improve the discovery efficiency of new chalcogenide materials and quickly identify material combinations with potential high performance. This could promote the rapid development of solar cells.

5 Conclusion

Aiming at the problem of insufficient PA of existing screening methods for perovskite materials, this study proposed the use of an improved GRBT algorithm for screening of hybrid perovskite band gap and hierarchical transfer learning for halogen double perovskite multi-property screening. Experiments indicated that the LF of the GBRT-SVM hybrid model decreased faster, with the astringent LF values being 0.115 and 0.160 lower than those of the GBRT and SVM algorithms, respectively. The optimization speed was 35 and 20 iterations faster than the other two algorithms, respectively. The pedestrian band gap prediction RMSE and coefficient of determination of GBRT-SVM hybrid model were 0.032 and 99.4%, respectively. The predicted values all converge neared the diagonal of the horizontal and vertical axes, with no significant deviation from the predicted points. They were 0.017 and -2.2, and 0.023 and -3.7% lower than GBRT and SVM, respectively. The maximum prediction error of the GBRT-SVM hybrid model for the band gap of hybrid perovskite was 0.071, and the minimum error was 0.005. The mean absolute error, RMSE, and coefficient of determination of the improved transfer learning method were 0.0097, 0.0205, and -5.06% lower than those of the ordinary method, respectively, and the running speed was 1.92s faster. The RMSE of bulk modulus measured by the improved transfer learning method was 0.037 lower than directly predicted, and the coefficient of determination was 11.18% higher. Factors that had an impact on the shear modulus of materials above 0.04 included Mendeleev number, electronegativity, octahedral factor, and total valence electrons. The BGVs of seven halogen double perovskite new materials were between [1.14-1.62] eV, with formation energies all below 0.05, and the ratio of shear modulus to bulk modulus greater than 1.75. The present study has identified several areas that necessitate further refinement. For instance, although the development of a screening model for new chalcogenide materials has been achieved using a small-scale dataset, enhancing the screening accuracy and efficiency to a certain extent, the screening accuracy of the model can be further enhanced through the integration of density flooding calculation in future iterations.

List of abbreviations

GRBT: Gradient Boosting Regression Tree

EC-PPL: Evaporation Crystallization-Polymer Pen Lithography

SVM: Supported Vector Machine

KNN: K-NearestNeighbor

B/G: Ratio of shear modulus to bulk modulus

f_x : Output variables of the training samples

w_i : Weights of the training samples

d_i : Distance from training samples to predicted samples

W : Weighted average of samples

$H_j(x_i)$: j Strong learners consisting of residual trees

$h_{j+1}(x_i)$: Weak Learner for $j+1$ Residual Trees

λ : Step-down

$f(x)$: Regression constructor

$\varphi(x)$: Nonlinear mapping function

b : Bias entry (computing)

C : Regularization constants

ζ and ζ^* : Slack variable (math)

q_i and q_i^* : Lagrange multiplier (math)

$\kappa(X_i, X_j)$: Kernel function (math)

$\kappa(x_i, x_j)$: Radial basis kernel function (math)

σ : Width of radial base core

$\|x_i - x_j\|$: Euclidean distance between two points

μ : Octahedral factor

r_B : The radius of the B-site ion

r_X : The radius of the X halogen ion

TF : Tolerance factor

r_A : The radius of the A-position ion

r : Pearson coefficient

X_{std} : The feature data after normalization

E_b : The formation energy of the compound

$E\left(A_mB_nC_k\right)$: Total energy of the compound

$E\left(A\right)$: The energy of the free atom A

$E\left(B\right)$: The energy of the free atom B

$E\left(C\right)$: The energy of the free atom C

Funding

This work is partially supported by the Natural Science Foundation of Shangluo University (21SKY114, 22KYZX14).

References

[1] Chen H, Cheng Q, Liu H, Cheng S, Wang S, Chen W, & Li Y. Organic-semiconductor-assisted dielectric screening effect for stable and efficient perovskite solar cells. Science Bulletin, 2022, 67(12): 1243-1252. DOI: 10.1021/acsaem. 2c04166.s001

[2] Shyur H J, & Shih H S. Resolving rank reversal in TOPSIS: A comprehensive analysis of distance metrics and normalization methods. Informatica, 2024, 35(4): 837-858. DOI: 10.15388/24-INFOR576

[3] Kim H W, Han J H, Ko H, Samanta T, Lee D G, Jeon D W, & Cho S B. High-Throughput screening on halide Perovskite derivatives and rational design of Cs_3LuCl_6. ACS Energy Letters, 2023, 8(8): 3621-3630. DOI: 10.1021/acsenergylett. 3c01207.s002

[4] Brociek R, Goik M, Miarka J, Pleszczyński M, & Napoli C. Solution of inverse problem for diffusion equation with fractional derivatives using metaheuristic optimization algorithm. Informatica, 2024, 35(3): 453-481. DOI: 10.15388/24-INFOR563

[5] Nguyen X B, Nguyen T H, Nguyen K V T, Nguyen T T T, & Nguyen D D. Efficient prediction of axial load-bearing capacity of concrete columns reinforced with FRP bars using GBRT model. Journal of Materials and Engineering Structures «JMES», 2023, 10(4): 551-568. DOI: 10.55228/jtst.12(2).38-47

[6] Abdelbasset W K, Elsayed S H, Alshehri S, Huwaimel B, Alobaida A, Alsubaiyel A M, & Abourehab M A. Development of GBRT model as a novel and robust mathematical model to predict and optimize the solubility of decitabine as an anti-cancer drug. Molecules, 2022, 27(17): 5676. DOI: 10.3390/molecules27175676

[7] Yang H, Liu X, & Song K. A novel gradient boosting regression tree technique optimized by improved sparrow search algorithm for predicting TBM penetration rate. Arabian Journal of Geosciences, 2022, 15(6): 461-482. DOI: 10.1007/s12517-022-09665-4

[8] Zhang W, Li R, Zhao J, Meng X, & Li Q. Miss-gradient boosting regression tree: A novel approach to imputing water treatment data. Applied Intelligence, 2023, 53(19): 22917-22937. DOI: 10.1007/s10489-023-04828-6

[9] Diao X, Diao Y, Tang Y, Zhao G, Gu Q, Xie Y, & Zhang L. High-throughput screening of stable and efficient double inorganic halide perovskite materials by DFT. Scientific Reports, 2022, 12(1): 12633. DOI: 10.21203/rs.3.rs-1565189/v1

[10] Zhi C, Wang S, Sun S, Li C, Li Z, Wan Z, & Liu Z. Machine-learning-assisted screening of interface passivation materials for perovskite solar cells. ACS Energy Letters, 2023, 8(3): 1424-1433. DOI: 10.1021/acsenergylett. 2c02818.s001

[11] Liu W, Lu Y, Wei D, Huo X, Huang X, Li Y, & Song D. Screening interface passivation materials intelligently through machine learning for highly efficient perovskite solar cells. Journal of Materials Chemistry A, 2022, 10(34): 17782-17789. DOI: 10.1021/acsenergylett. 2c02818.s001

[12] Lai M, Shin D, Jibril L, & Mirkin C A. Combinatorial synthesis and screening of mixed halide perovskite megalibraries. Journal of the American Chemical Society, 2022, 144(30): 13823-13830. DOI: 10.1021/jacs.2c05082

[13] Liu H, Feng J, & Dong L. Quick screening stable double perovskite oxides for photovoltaic applications by machine learning. Ceramics International, 2022, 48(13): 18074-18082. DOI: 10.1016/j.ceramint.2022.02.258

[14] Yuan S, Liu Y, Lan J, Yang W, Long H, Li W, & Fan J. Accurate dimension prediction for low-dimensional organic–inorganic halide perovskites via a self-established machine learning strategy. The Journal of Physical Chemistry Letters, 2023, 14(32): 7323-7330. DOI: 10.1021/acs.jpclett. 3c01915.s001

[15] Kurani A, Doshi P, Vakharia A, & Shah M. A comprehensive comparative study of artificial neural network (ANN) and support vector machines (SVM) on stock forecasting. Annals of Data Science, 2023, 10(1): 183-208. DOI: 10.1007/s40745-021-00344-x

[16] Santos C F G D, & Papa J P. Avoiding overfitting: A survey on regularization methods for convolutional neural networks. ACM Computing Surveys (CSUR), 2022, 54(10s): 1-25. DOI: 10.1145/3510413

[17] Chen Z. Research on internet security situation awareness prediction technology based on improved RBF neural network algorithm. Journal of Computational and Cognitive Engineering, 2022, 1(3): 103-108. DOI: 10.47852/bonviewjcce149145205514

[18] Hebbi C, & Mamatha H. Comprehensive dataset building and recognition of isolated handwritten Kannada characters using machine learning models. Artificial Intelligence and Applications, 2023, 1(3):179-190. DOI: 10.47852/bonviewaia3202624

[19] Wang G, Jia Q S, Zhou M C, Bi J, Qiao J, & Abusorrah A. Artificial neural networks for water quality soft-sensing in wastewater treatment: a review. Artificial Intelligence Review, 2022, 55(1): 565-587. DOI: 10.1007/s10462-021-10038-8

[20] Mena-Yedra R, López-Redondo J, Pérez-Sánchez H, & Martinez-Ortigosa P. Boosting pairwise molecular contrasts with scalable methods. Informatica, 2024, 35(3): 617-648. DOI: 10.15388/24-INFOR558

Secure Power Management in Wireless Sensor Networks for Power Monitoring Using Deep Reinforcement Learning

Jin Du[1], Xiaodong Wang2, Hao Zhang[2], Hao Zhang[2]
[1]State Grid Jibei Information & Telecommunication Company. Beijing 100054, China.
[2]State Grid Jibei Electric Power Company Limited, Beijing 100054, China.
E-mail: JinDu12@outlook.com, Linazheng123@outlook.com, yizhou89@outlook.com

Keywords: deep reinforcement learning, energy efficiency, secure power management, secure routing, wireless sensor networks

Received: September 10, 2024

Wireless Sensor Networks (WSNs), widely used in healthcare, environmental monitoring, and military applications, rely on battery-powered sensors with limited energy reserves. Extending the operational lifetime of these networks is a critical challenge, particularly under both external and internal security threats. This paper introduces a novel Secure Power Management (SPM) framework that integrates Deep Reinforcement Learning (DRL) with a token-based Elliptic Curve Cryptography (ECC) policy to optimize energy efficiency and ensure secure communication. The SPM system dynamically transitions sensor nodes between active and sleep modes based on network activity, significantly reducing energy consumption. It remains effective even in the presence of malicious attacks, maintaining robust performance across various threat scenarios. Simulation results demonstrate that the proposed method reduces power consumption by up to 20.01% compared to conventional schemes, while improving energy efficiency by 15%, enhancing packet delivery reliability, and prolonging network lifespan. These findings highlight the potential of DRL-based secure power management for resilient and energy-aware WSN operations.

Povzetek: Razvili so varen sistem upravljanja z energijo za brezžična senzorska omrežja (WSN) z uporabo globokega okrepljenega učenja in ECC za podaljšanje življenjske dobe.

1 Introduction

The previous several decades have seen a transformation in our lifestyles brought about by interactive settings. In the process of reworking our day-to-day lives to make them more comfortable, sensing plays a crucial part. The transformation of our houses into smart homes may be attributed, for example, to the presence of networked sensors that possess the ability to make deductions. Within the scope of such an application, household security, household medical care, family processing of information, family entertainment, and family business are all included. If wired sensing systems need a separate twisted shielded pair wire connection, then wireless sensor networks (WSNs) are more advantageous from an economic standpoint than wired sensing systems. As a result, the latter has considerable expenses associated with its implementation (Kaushik, J. (2021), Swati, et.al, (2022)). On the other hand, in order to properly accomplish its specific goal and achieve cost minimization, a WSN has to operate for a considerable amount of time continuously. Importantly, there is a trade-off between the life expectancy of a WSN and the amount of power that is used by the different functions of the network. In addition, reducing the amount of energy that is used is a primary priority owing to the various components involved. One of the main focuses of research on wireless sensor networks is the appropriate management of network nodes within the constraints of the WSN's limited resources in order to regulate the networks' energy properly. Another difficulty stemming from the sensor's intended use—data collection—is the restricted resources available to the WSN node. As a consequence, bringing down the price of these gadgets to meet consumer demands is crucial, but doing so poses security risks due to their inability to provide even the most basic degree of protection. To be compatible to other network management activities, security administration in WSNs should not rely heavily on connections, processing, or storage (Prajapati, S., et.al, (2023), Cao, C., et.al, (2021)).

As a vital component of the energy sector, electricity affects people's daily lives and the economy as a whole, making the safe and reliable functioning of the power delivery and distribution system a top priority (Sinha, S., et.al, (2022), Thangamuthu, A.P. et.al, (2021)). This paper proposes an application system for the monitoring, examination, security, and interactive aid of layered power distribution and transmission systems using WSN technology and combined with the operating and monitoring requirements of these systems (Srividhya, G., et.al, (2021), Jaiswal, S.K., et.al, (2023)). An insider attack or other hostile assault may be effectively mitigated with secure power management. Afterwards, it intelligently

puts the nodes of the wireless sensor network into sleep mode when they're not in use and wakes them up when needed. With the goal of delivering reliable and safe power management for WSNs (Wang, S., et.al, (2021), R., S.K., et.al, (2021)).

We suggested a method for WSN power management that is both efficient and secure. By combining an effective routing strategy with well-managed security, this method efficiently transmits data throughout the network. The primary goal of this study is to enhance the current systems' power management and routing efficiency while simultaneously discovering and providing robust secure power management in WSNs. On the other hand, reducing the total power consumption of the network is more important owing to the intricacy of the issue that has been presented; this is still a promising field for study. Consequently, this article discusses the primary methods that have been discovered for reducing the amount of electricity that is used (Kaushik, I. et.al, (2020), Bharanidharan, R. (2020), Qamar, S., et.al, (2020)).

Here is how the remainder of the work is structured. The paper's power management models are introduced in Section 2. Section 3 introduces the SPM in WSN that makes use of DRL. In Section 4, we provide and analyze the findings of the simulation. In Section 5, we draw a conclusion and lay forth the groundwork for our future research.

1.1 Power-management and control in WSN

Power consumption model

The ability to perceive, process, and send data is inherent to every node. A WSN relies on sensor nodes, which are often battery-operated and have limited computing and storage capacities, and uses a wireless means for communication. One of the most important resources in WSNs is energy, and these networks use a variety of communication protocols that aim to save it. The sending and receiving of data are also components of communication (G, M., A., et.al, (2023), Roja, P., et.al, (2022), Khan, T.A., et.al, (2022)). Here, we provide a model for the power consumption of communication (P_c) as:

$$P_c = N_T[P_T(T_{ON} + T_{ST}) + P_{OUT}(T_{ON})] + N_R[P_R(R_{ON} + R_{ST})] \quad (1)$$

Where N_T is the typical pulse rate at which the transmitter is activated, N_R is the typical frequency of the receiver's operation in milliseconds, P_T is the amount of power that the transmitter uses, and P_R is the amount of power that the receiver uses. T_{ON} is the transceiver's startup time, and is the transmitter on time P_{OuT} is the output transmitting power, R_{ON} stands for the receiver's start-up time, and R_ST for its on-time performance.

When a system is not in use, it may be put into a low-power state—also called "sleeping mode"—or its power can be "gated"—that is, turned off—a control strategy. This is a low-power design approach that works, but it adds additional power consumption and operational latency whenever the machine sleeps or wakes up. Negative power savings occurs when the structure's sleeping period is small and the additional power usage is more than while the system is operating. A positive power savings may be achieved by using the power monitoring system, and its break-even point has been described. Additionally, it guarantees that the machine remains in sleep mode for just the correct amount of successive clock cycles before being awakened up (Nandal, V., et.al, (2021), Yuvaraja, M., et.al, (2024), Almansoori, M.N., et.al, (2022)). Negative power saving, or increased power consumption, will occur if the system's sleep duration is insufficient to meet the break-even threshold. The additional regulating activities caused by power management turning on and off a system result in excess energy E_"overhed," as demonstrated in Eq. (2).

$$E_{\text{overhead}} = 2\frac{W_H}{a}E_{cyc}^{s} \quad (2)$$

Where W_H represents the active dimension of the goal system, α is the ratio of the overall route area to the area of the gated structure, and E_{cyc}^{s} is the amount of energy needed to turn the functional units on and off. Even though the system continues to utilize power when switching on and off, the voltage is unstable throughout this time, rendering it useless for typical tasks. Using Equation (2), we can express the overall energy savings achieved by the framework all over the N clock cycles of gating.

$$E_{\text{saved}} = E_{cyc}^{L}\frac{DIBL}{mV_t} \times \frac{N^2}{2} \times \frac{aLV_{\text{dd}}}{2\left(\frac{V_2}{2} + \frac{C_D}{C_s}\right)} \quad (3)$$

Three variables: supply voltage, subthreshold slope factor (m), and drain-induce barrier reducing factor (DIBL) (V_{dd}), temperature voltage (V_t), and leakage factor (L) are all variables in this context $= E_{cyc}^{L}/E_{cyc}^{s}$, and C_D and C_S standing for switching capacitance and decoupling capacitance, correspondingly. When $E_{\text{saved}} = E_{\text{overhead}}$, To get the break-even point, N, use the following formula.

$$N = 2\frac{1}{L\alpha}\sqrt{\frac{mV_tW_H}{V_{dd} * DIBL}\left(1 + 2\frac{C_D}{C_s}\right)} \quad (4)$$

Power management techniques

There are typically two states of operation for a sensor node: active and idle. In active mode, the node acquires data from sensors and communicates wirelessly. In idle mode, it does nothing. Since the power consumption during the active phase is necessary to carry out the sensor node's fundamental function, it is considered acceptable. In contrast, superfluous subsystems should be disabled

entirely as any power used while inactive is a waste of resources. The sensor node may still have a power leak that gradually depletes the battery. When deployed in large numbers (50-100), the power-hungry sensors still need to have their batteries changed often, even with advanced battery technology (Hassan, K., et.al, (2023), N.P., et.al, (2020), Sharma, A., et.al, (2023), Ahmad, R., et.al, (2022)). This is why these systems use power-management strategies, some of which will be covered in more detail below.

- **Power gating:** Since the node receiver only turns on when a message is waiting for it, the wake-up technique is also one of the most energy-efficient ways to handle idle mode. A typical transceiver uses a lot of power, whereas this kind of receiver only a fraction of that. As a rule, latency is greater, sensitivity is lesser, and data rate is lower. Since this method only activates the intended nodes, it also lessens the likelihood of overhearing. When turned on, wake-up receivers also drain power supplies, thus they need to be very power efficient if they want to reap the benefits of energy gain.

- **Dynamic voltage frequency scaling:** Here, we may dynamically lower the voltage and frequency selectively, allowing the block to merely satisfy its present job deadlines. This method is similar to power gating in that it runs slower blocks at lower frequencies while providing voltage without entirely cutting off the supply. In order to calculate the necessary decrease whilst the system is running, this method needs sufficient information about the application's power consumption. Maximizing power savings also entails controlling voltage and frequency using hardware and software.

- **Energy harvesting from surrounding:** The sensor nodes' reliance on a battery for electricity puts a cap on the network's lifespan because of the battery's limited capacity to store energy. An appealing alternative for systems that rely only on batteries is energy-harvesting devices in conjunction with rechargeable batteries. In order to greatly extend the life and functionality of the sensor nodes, these systems are able to collect energy from their environment. Solar, thermoelectric, electrostatic, electromagnetic waves, and piezoelectric methods are all used to collect energy. But the environment isn't always predictable, thus it's possible that a certain harvester won't ensure the node can run safely. As a result, hybrid harvesters are being utilized, which include two or more ecological harvesters.

2 Related works

Ghosh, R., et.al, (2021) presented an innovative, smart controller to sustain mobility in wireless sensor networks (WSNs). Principally, the focal point is dependent on the arrangement of fuzzy input variables (i.e., remaining battery power [RBP], mobility, and centrality solution) to crucial usages, similar to personnel safety in an industrialized atmosphere. A mobility controller dependent upon type-1 fuzzy logic (T1FL) is planned to support sensor mobile nodes (MN). Here, a role model cluster head (RMCH) is picked out among the cluster heads (CHs) that may simply convey the message to the mobile base station (BS) by determining the appropriate type-1 fuzzy (T1F) descriptors such as RBP, mobility of the sink, and the centrality of the clusters. Type-1 fuzzy inference system (Mamdani's rule) is utilized to opt for the possibility to be RMCH. The validity of the introduced model is carried out by means of multiple linear regressions.

Gudhekar, G.S., et.al, (2021) The design and development of an intelligent monitoring and controlling system for home appliances in a real time system is presented in this project. This system principally monitors the electrical parameters such as voltage and current and subsequently calculates the power consumption of the home appliances that are need to be monitored. The innovation of this system is controlling mechanism implementation in Iot based by server. The user can control the home appliances by mobile app using server data. The developed system is a low-cost and flexible in operation and thus can save electricity expense of the consumers. Also, the proposed system is an economical and easily operable. Due to these intelligent characteristics, it becomes an electricity expense reducer and people friendly. Adopting WSNs for power management provides great advantages over traditional wired system. By using WSN technology data can be collected through sensing unit and transferred wirelessly to a control system for operation and management.

Nandini, G., et.al, (2020) Power utilization is the significant worry in our daily life. It is assessed that normal power utilization of family apparatuses is 90 units (kWh) per month in India. All gadgets are worked by taking power. It has to shrewd screen power and diminish the measure of utilization. Brilliant screen which computes the electrical boundaries, for example, voltage and power at that point therefore power devoured by the devices. So, it can get away from of the gadgets taken a tremendous part in power utilization. WSN senses and records the physical changes in environment. The essential idea of WSN is the devices are not associated truly. The gathered data is given to the devices which are kept to certain territory. In this paper IOT is utilized alongside WSN with the goal that the framework can get to the devices from any place. A smart energy management system can contribute towards reducing the expenses and still satisfy the demands. This framework additionally gives adaptability in activity and consequently we can spare the power.

Das, S. (2023) main challenge of power management in WSNs is to maximize the network lifespan by scheduling the monitoring activities and optimizing the data transfer rate. Therefore, various power control methods were

proposed to reduce energy consumption and extend the network lifetime. This article provides an overview of the power management techniques used to reduce energy consumption in WSNs. Specifically, this article explains various scheduling, routing and sleeping techniques that are used to manage the power drain in WSNs.

Bengheni, A. (2024) presented Relay node selection scheme and Deep sleep period for power management in Energy Harvesting Wireless Sensor Networks (RD-EHWSN), a new energy-saving scheme founded on asynchronous duty cycling. RD-EHWSN reduces sensor node energy consumption and guarantees equilibrium energy use between sensor nodes in WSN with the energy harvesting capacity by adjusting these sensor nodes duty cycles more drastically and deeply by according to the estimated value of its residual energy on the basis of future-presented harvested energy, and this is done through the use of a new proposed energy threshold policy. RD-EHWSN also grips the benefit of transmitter initiated using the low power listening (LPL) technique with short preamble messages and uses a new relay node selection procedure to achieve the load balancing in WSN. We implemented RD-EHWSN by using OMNeT++/MiXiM. For evaluation, we compared it with PS-EHWSN, under multiple concurrent multihop traffic flows scenarios and scenarios in which nodes can harvest different energy harvesting rate. In all experiments, RD-EHWSN significantly outperformed the PS-EHWSN scheme; the results of simulation demonstrate that our scheme enhances the general yielding of WSN thru lessening the energy consumption and the mean latency, as well as raising the packet delivery ratio and the throughput.

Table 1 shows related works.

Table 1: Summary of related works

Studies	Methodologies	Results
Ghosh, R., et.al, (2021)	For sensor mobile nodes, a type-1 fuzzy logic (T1FL) mobility controller is envisioned. The role model cluster head (RMCH) is chosen between the cluster heads (CHs) to send the message to the mobile base station (BS) using type-1 fuzzy (T1F) characteristics such RBP, sink mobility, and cluster centrality.	High PDR and Delay
Gudhekar, G.S., et.al, (2021)	This system monitors voltage and current to compute household appliance power usage. The server-based IoT controlling mechanism is this system's novelty. Mobile app users may manage home appliances using server data. With its cheap cost and flexibility, the system may save users money on power.	Higher mobility causes high PLR
Nandini, G., et.al, (2020)	The gathered data is given to the devices which are kept to certain territory. In this paper IOT is utilized alongside WSN with the goal that the framework can get to the devices from any place. A smart energy management system can contribute towards reducing the expenses and still satisfy the demands. This framework additionally gives adaptability in activity and consequently we can spare the power.	Gathered data consumes more energy and causes delay
Das, S. (2023)	Various power control methods were proposed to reduce energy consumption and extend the network lifetime.	Predicted the performance for network lifetime and energy consumption
Bengheni, A. (2024)	Relay node selection scheme and Deep sleep period for power management in Energy Harvesting	High energy consumption for large scale environment

2.1 Existing gaps and challenges

There are several obstacles to overcome while deploying and operating WSNs, the most significant of which is striking a balance between energy efficiency and strong security measures. Suboptimal network performance is typically the result of traditional methodologies that treat

these concerns independently. Important obstacles include of the following:

- Energy savings: The fundamental problem is finding ways to prolong the lifetime of the network without compromising its operational performance, considering the restricted authority resources of the sensor nodes.
- Safety: Various security risks, such as data breaches and node manipulation, might affect WSNs due to the open character of wireless communication; hence, thorough security solutions are required.
- Ability to adjust: Conventional static approaches are inadequate in the face of constantly changing threat environments and dynamic network circumstances, necessitating flexible tactics.

To tackle these issues holistically, our suggested SPM approach makes use of DRL. In contrast to more static approaches, SPM continuously optimizes power and security by responding to evolving network circumstances and potential security risks. Not only does this method address the shortcomings of current methods, but it also provides a versatile and extensible framework for managing WSNs in their entirety.

3 Materials and methods

The experimental setup for the suggested SPM, which makes use of DRL and an improved LEACH algorithm, is detailed in this section. We will also discuss the results of the suggested techniques, which include LEACH improvement to reduce power consumption and increase the lifetime of the network and lightweight cryptography to prevent third parties from joining the group of sensors.

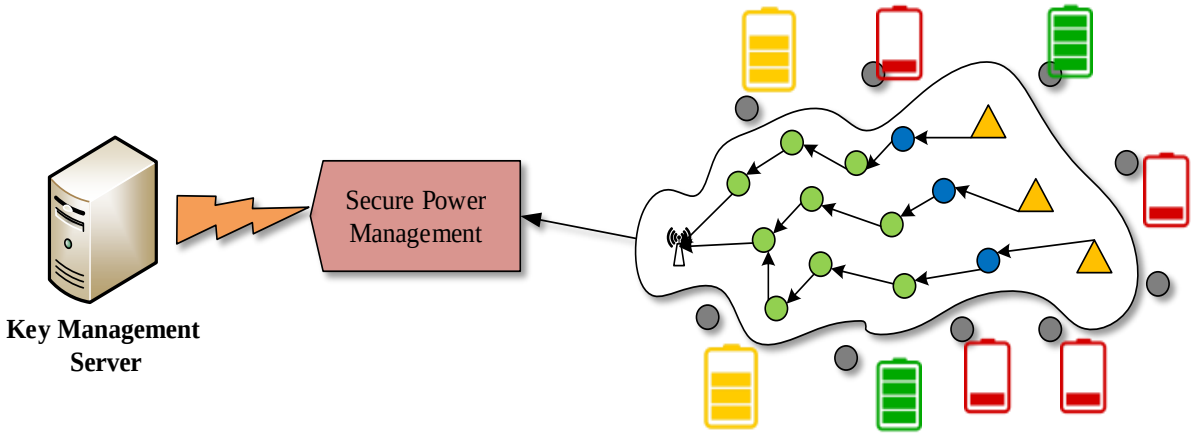


Figure 1: System model using SPM with DRL

3.1 Power optimization using DRL

The idea behind the suggested SPM method is to provide a single solution that uses DRL to optimize network operations in real-time, which would increase energy effectiveness and safety equally. Accurately modeling the communication quality between nodes is of utmost importance in our suggested technique. Our energy savings and safety measures, as well as the improvement of the network's general performance, are built around this paradigm.

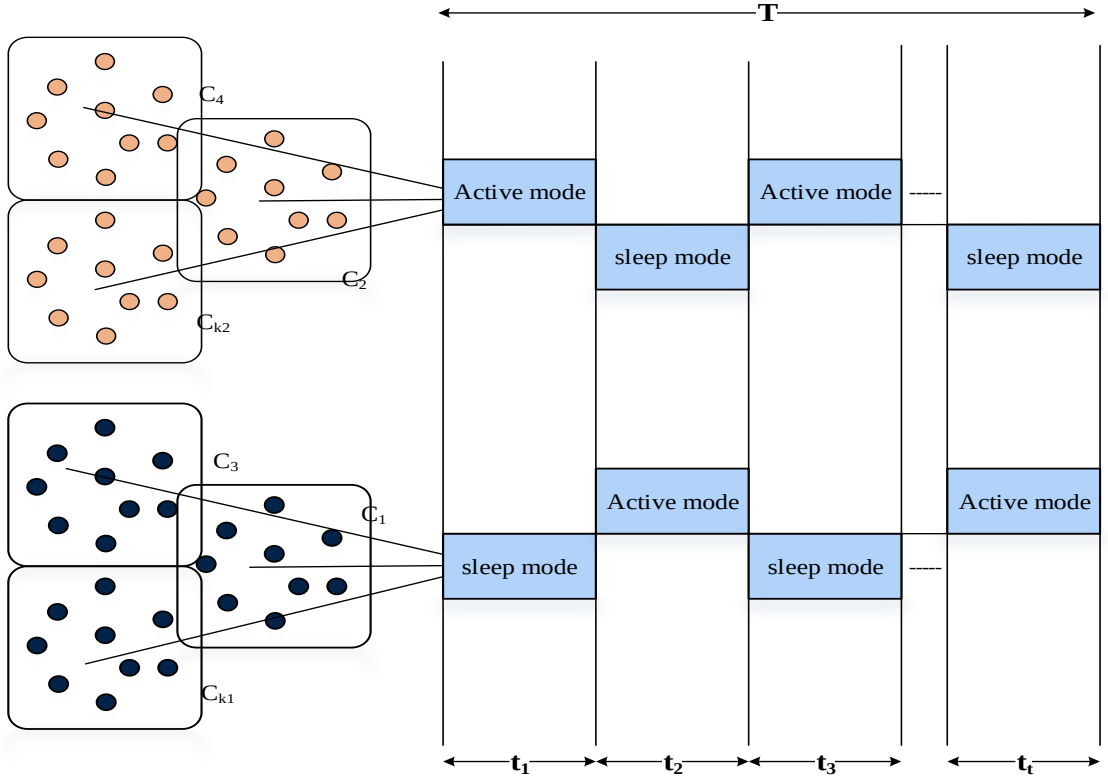


Figure 2: Power Management in WSNs

The quality of the connection between nodes i and j may be described in terms of the wireless communication features of WSNs by

$$Q_{ij} = \frac{P_{tx} \times G_{tx} \times G_{rx}}{L(d_{ij}) \times V} \quad (5)$$

where P_{tz} is the transmission power and G_{tz} and G_{rz} is the gain of the receiving antenna and the broadcasting antenna, respectively. $L(d_{ij})$ denotes the function for path loss, while V stands for the power of noise. We establish a security index that takes into account the potential dangers to WSN and the amount of energy used by each node:

$$E_{eff}(i) = \frac{D_i}{E_i} \quad (6)$$

$$S_{\text{risk}}(i) = \frac{A_i}{T} \quad (7)$$

Where D_i E_i, where is the energy consumption of node i and can be determined using (4), is the data throughput of node i. A_i counts the assaults that node i endured during the time interval T.

$$E_i = P_{tx} \times t_{tx} + P_{rx} \times t_{rx} + P_{\text{idle}} \times t_{\text{idle}} \quad (8)$$

WhereP_{tz}, P_{rz}, and P_{idle} indicate the node's power while it is transmitting, receiving, and idle, correspondingly. At the same time, t_{tx}, t_{r_x}, and t_{dlle} the amount of time that the node spent in each of these stages, separately. The following is a definition of the optimization issue that aims to optimize energy efficiency while minimizing the overall network security risk:

$$\max_p \sum_{i=1}^{N} E_{\text{eff}}(i) = \max_p \frac{D_i}{E_i} \quad (9)$$

$$\min_p \sum_{i=1}^{N} S_{\text{risk}}(i) = \max_p \frac{A_i}{T} \quad (10)$$

In this case, p denotes the network policy, which includes the encryption level and transmission power of every node.

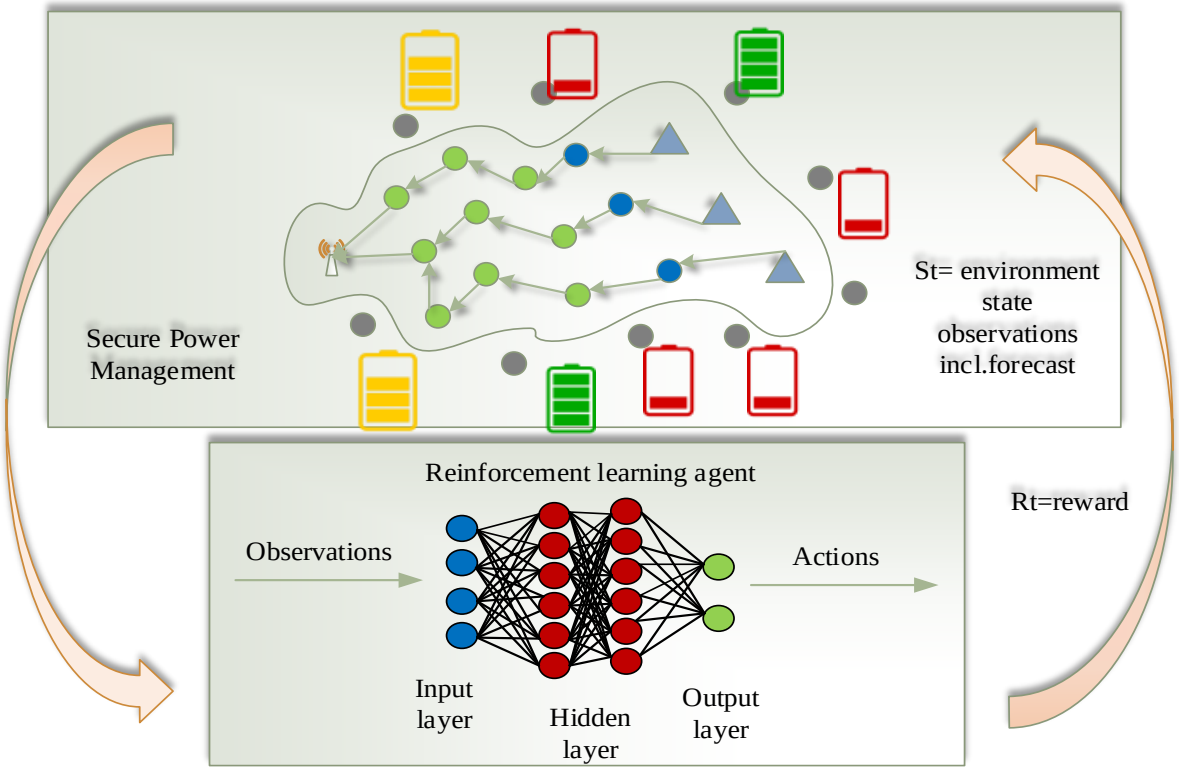


Figure 3 (a): SPM using DRL

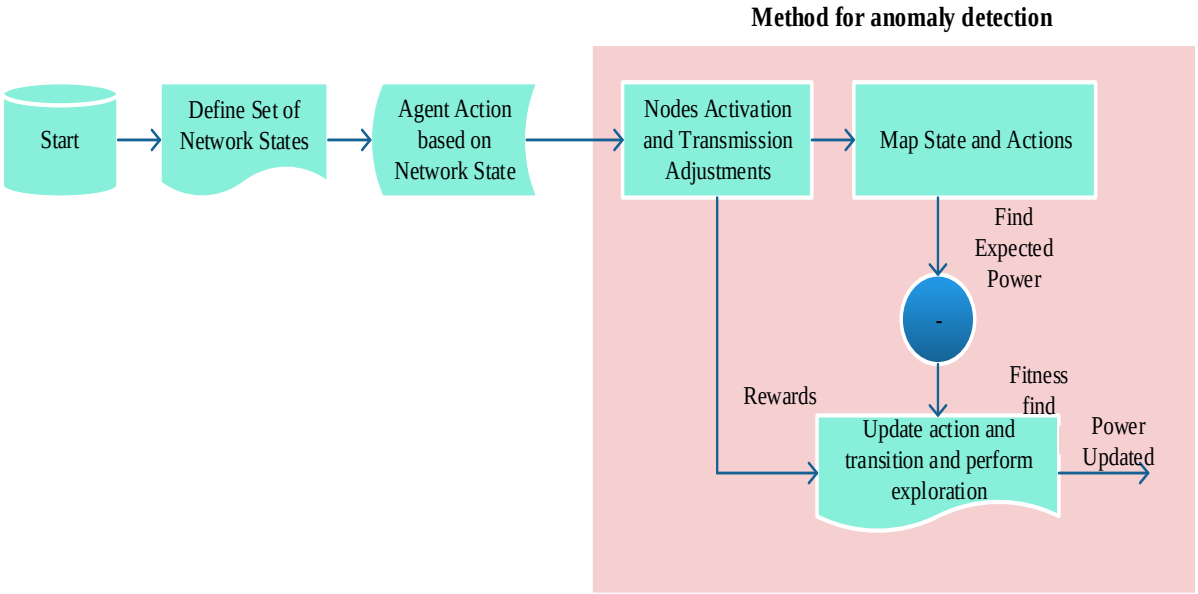


Figure 3 (b): DRL flowchart

We begin by defining an MDP, which is a quintuple array, so that we may utilize DRL for optimizing the WSN approach. (S, A, P, R, γ). S is the state space. A lies the domain of action. The probability of transferring states is abbreviated as P. $P(s' \mid s, a)$. R is the reward function, well-defined as $R(s, a, s')$. γ represents the discount percentage. In order to maximize the network's performance while simultaneously addressing energy efficiency and security, we use DRL to discover a plan. We introduce a novel composite reward function:

$$R(s,a) = \alpha \times \sum_{i=1}^{N} E_{eff}(i) - \beta \times \sum_{i=1}^{N} S_{\text{risk}}(i) \quad (11)$$

in which the weighting parameters are denoted by α and β. Reinforcement learning aims to maximize the network's predicted reward. This idea might be stated as

$$J(\pi) = E_{\pi}(R_1 + \gamma R_2 + \gamma^2 R_3 + \cdots] \quad (12)$$

where R_t is the payoff if time t is met. We use Bellman's the greatest efficiency principle for mathematical proof [34] to show that our suggested DRL-based neural network design is the best technique, instead of Q-value. This Bellman equation is satisfied by the Q-value function that the neural network generates:

$$Q(s,a) = R(s,a) + \gamma \times E\left[\max_{a'} Q(s',a')\right] \quad (13)$$

in where γ stands for the discount element, $s'$for the next state, and $a'$for the next action.
Finding the policy π that optimizes the Q-value function is the objective of strategy optimization. The resulting optimization issue looks like this:

$$\max_{\pi} E[Q(s,a)] \quad (14)$$

We may get the policy's gradient using the policy gradient approach, which is as

$$\nabla J(\pi) = E[\nabla \log \pi(a \mid s)(Q(s,a) - V(s))] \quad (15)$$

where the value function of state s is denoted as V(s). We can find the best policy by repeatedly changing the policy parameters.
It is important to mention that our suggested method is to develop a WSN strategy optimization process that is more resilient and successful than deep Q-learning. This will guarantee that SPM can manage the complexities and difficulties that are unique to WSNs. Some examples of this include the ever-changing threat environment for network security, the varied and flexible nature of network structures, and the changeable energy profile of sensor nodes. In particular, our optimization approach exhibits the traits listed below:

- Our state space representation has been enhanced. To better capture the intricate dynamics of WSNs, SPM's state space has been expanded to include a wider range of network properties.

- Function for personalized rewards. Although most deep Q-learning systems do not aim to maximize energy efficiency and ensure network security, we use a tailored reward function that is specifically designed to do just that.

- We optimize policies for WSNs. Node energy restrictions and the necessity for quick reaction to security threats are two examples of operational limitations and performance objectives that our methodology tailors the policy optimization method to meet in WSNs.

- High-tech system for reliving past experiences. To improve upon the standard deep Q-learning method, we have integrated a state-of-the-art experience replay process that is more suited to the spatial and temporal heterogeneity in WSNs.

- The incorporation of procedures tailored to WSNs. By integrating SPM with protocols designed for WSNs, we can easily translate learned techniques into policies that can be put into practice in a real-world system setting.

3.2 Secure path selection for power aware routing

We have used a private key that is produced using a random integer within a particular range of the curve's field size in our suggested technique. The four steps of the scheme are as follows: determining the total number of clusters, verifying each node, determining who is the cluster head, distributing keys among all members of the group, creating a routing table using the routing protocol, and finally, rekeying in the event that any changes occur within the group.
All the nodes in the network must have the same traits and be at the same authority level for the scheme to function. In each case, the protocol specifies a mechanism that each node with data to transmit follows in order to determine where to send it. It is possible for every node in the network to construct or calculate its private key inside the ECC. These keys are integers ranging from 1 to 256 bits, which is the field size of the curve.
The secp256k1 curve and the elliptic curve digital signing technique (ECDSA), which are often used in cryptocurrency systems, are utilized in this research. The unique non-random construction of secp256k1 makes it possible to do very efficient computations. A well-optimized secp256k1 implementation often outperforms other known EC curves by 30% or more. On a curve ($\mathbb{F}p$), the six domain parameters of secp256k1, also called the Koblitz curve, are defined as a sextuple T = (P, a, b, G, n, and h).
With generator point G, the elliptic curve, with r_i as the private key for node i, the computational cost of the cyclic ECC key creation and distribution in the method one table can be estimated for n the amount of nodes in the sensor network. Scalar multiplication, as shown in Equation (10) is required to get a shared key for a node.

$$P_t = (r_1 \cdot r_2 \cdot r_3 \cdot r_4 \ldots\ldots\ldots \cdot r_n) \cdot G \quad (16)$$

EC scalar multiplication is where the "operator" comes in. A total complexity may be determined for n nodes by using Equation (11).

$$O\left(n(r^n \cdot G)\right) \tag{17}$$

By analyzing the SPM data, a unique random key may be generated. The one who distributes the secret key is

$$\text{Skey}\ (n) = \text{Rand}\ () \times \text{Dist}\ (n, n+1) \times \text{EfM}\ (n) \tag{18}$$

$$\text{SKey}\ (n) = \text{Rand}\ () \times \text{Dist}\ \left(n, n\frac{1}{LR(n)} + \frac{1}{\text{DR}\ (n)}\right) \tag{19}$$

A significant drawback of this protocol is that it chooses an additional cluster head only after a full round has passed, and it doesn't take the node's leftover energy into account at all. In the event that a cluster header is selected during the course of a round and subsequently dies, all aggregate data associated with that cluster header will be erased, and subsequent rounds will not be able to receive or send data successfully. One other issue with LEACH is that cluster heads are selected at random or by looking at the energy of the highest energy cluster. However, this selection process doesn't account for the distance to the sink or the size of the package, which are factors that determine the energy needed to transmit the message. The node that is farthest from the base station, even if it has the greatest energy, may not be the best candidate. To summarize, in most cases, the parameters that determine if the CH desires to be replaced have to do with the amount of energy required to collect all the packets sent by the light nodes, sensor nodes, or leave nodes and get them ready to submit the information via regular links, taking into account the power consumption. The identification of an Advanced Persistent Threat (APT) impacting the CH is another factor that might initiate a request for replacement.

In response to these issues, we suggested the following two changes to the initial LEACH protocol: The cluster header calculates the energy needed to transport a packet before sending it, E_{tx}. If the amount of power needed to send a packet, E_{tx}, is less than or equal to the residual energy ($E_{rs\ \text{sidual}}$) of the node (i.e., $E_{tx} \leq E_{\text{rosidual}}$), Afterwards, the cluster header will be demoted and a replacement will be selected. The previous cluster header should have its aggregated data moved to the new one. You may calculate the necessary transmit energy for a packet of length PacketLength by applying Equations (6) and (7).

$$E_{LX} = S_{EIx} \times \text{ PacketLength } + S_{Eys} \times d^2 \tag{20}$$

$$E_{EX} = S_{EEx} \times \text{ PacketLength } + S_{Emp} \times d^4 \tag{21}$$

where S_{Etx} is the energy parameter for the transmitter at the base station and S_{Emp} and S_{Efs} and in the free space model, they represent the energy characteristics for the kind of radio transmitter, respectively. In this context, "d" refers to the distance in kilometers among the node and network hub S. To find the distance among node i and base station S, we may use the formula (8), which takes into account the ephemeral nature of distances.

$$d_t = \sqrt[2]{(x_t - x_s)^2 + (y_t - y_s)^2} \tag{23}$$

Where x, in any dimension, denotes the node's or base station's two-dimensional coordinates. The energy-factor for distance is used to choose the cluster header F_{cd}, calculated by use of Equation (2). The highest-ranking node F_{ed} is consistently selected as the cluster's parent node. In contrast to the LEACH protocol's reliance on nodes with the greatest energy levels or random selection techniques, this method takes the distance among the node and the base station into account during the selection process.

$$E_{at} = E_{\text{rsstual}} - E_{tx} \times \frac{1}{d^2} \tag{24}$$

In addition, nodes are only considered for the role of cluster leader if they guarantee that their residual energy $E_{r\ \text{rsidual}}$ is greater than transit energy E_{tx} may be a good choice for cluster headers.

For an RREQ message delivered by the source node, the next hop in SPM decides the RSS (0<RSSf_0(n)<100) and then floods the sent message. Receiving nodes estimate the source's power utilization metric (PfM) using Eq., and the destination node's RREP message conveys this information together with the node's critical routing metrics, including receive-rate, loss-rate, and delay-rate,

$$\text{PFM}\ (n) = \text{RR}\ (n) + \frac{1}{\text{LR}\ (n)} + \frac{1}{\text{DR}\ (n)} \tag{25}$$

where

RR(n) → receive-rate of the node n

LR(n) → loss-rate of the node

n DR(n) → delay-rate of the node n.

3.3 Transmission power adjustment

The distance between nodes in the route determines how the nodes' transmission power is changed. It lessens energy waste. It will be necessary to use less transmission power for the nearby node. More energy is wasted since the transmission power is set higher for the closest node. Until the packet reaches its destination, the transmission power of each node in the route is modified according to the distance between them. The two-ray ground reflection simulation will be used to modify the transmission power.

$$P_r = P_t G_t G_r h_t^2 h_r^2 / d^4 L \quad (26)$$
$$P_t = P_r d^4 L / G_t G_r h_t^2 h_r^2$$

where G_r and G_t stands for the receiving antenna gain and the transmitting antenna gain, with P_r representing the received signal power (in Watt) and P_t the transmitted signal power. In this case, L stands for system loss and d for the distance between the transmitter and receiver. The antenna heights for transmitting and receiving signals are denoted by ht and hr, respectively. In wireless sensor networks, transmission range levels might be used. Neither too many nor too few transmission range levels are acceptable. Power loss for the range level shift will be greater when there are an excessive number of range levels.

4 Results

Using NS3.27, a simulation was run to include the DRL algorithm into the node controlling power of the WSN. NS3 is a simulator that models network protocols primarily using discrete event time. In the virtual setting, the nodes are dispersed. need NS3's node-config command to be set up as mobile nodes. You can find a description of the parameters used to simulate the SPM system in Table 2. Each node We have a data source at (0,0) and a target node sink at (150,150) in our 500m × 500m square area simulation, with 100 sensor nodes randomly dispersed throughout. Our parameters for the communication set-up are a 30m radius, 500mV for the node's transmission power, and 250mV for its reception power. The given method has the following parameters: α = 0.5, β = 1, ρ = 0.2, and Δ = τ ij 0.02. when shown in Figure 2, the communication linkages among the nodes re-establish themselves when the power consumption decreases, once the whole network finishes routing at the rated power. Thus, although the total number of nodes required to finish data transfer grows in relation to the number of hops between them, the energy usage of those nodes actually decreases. At last, the network's viability is enhanced and its service life is prolonged. The outcomes of the simulation are displayed in Figure 4.

Table 2: Parameters used for proposed SPM model

Definition	Average Value
Size of testbed (number of nodes)	50,100 homogenous
Number of base stations (Bs)	1
Initial energy for each sensor	0.005 j
Radio circuitry energy dissipation, Eelec	50 nj/bit
Energy dissipation of amplifier in free-space, Efs	10 pj/bit
Energy dissipation of amplifier in multipath, Emp	0.0013 pj/bit
Energy consumption for data aggregation, Eda	5 nj/bit
Global testbed area	$n \times n$
Local area (cluster size)	n/nc
Time	10 rounds
Packet size	400-bits
Message size	328-bits
Encryption key length	256-bits

First, a 50 nodes situation is used to conduct the simulation study for the SPM scheme. The packet delivery ratio measures how many data packets made it from the source node to all of their destinations as a percentage. Equation 3.4 is used to measure PDR.

$$PDR = \sum_{0}^{n} \text{packet Received / Time} \quad (27)$$

Packet Loss Rate (PLR)—As shown in Figure 4 and Equation 3.5, the PLR is the rate at which data packets are lost in a network as a function of time.

$$\text{PLR} = \sum_{0}^{n} (\text{Sent Pkts} - \text{Revd Pkts}) / \text{Time} \quad (28)$$

The average delay is the time that has elapsed between the most recent and most prior packets received. Here, n=50, and it is measured using Eq. 3.6, where n is the total amount of nodes.

$$\text{Average Delay Time} = \sum_{0}^{n} \frac{\text{packetreceivedtime} - \text{packetsenttime}}{n} \quad (29)$$

We begin by implementing the power control strategy separately in our testbed experiment. Using power control, we gather data for a set destination node location several times and compare it to the regular situation. According to our findings, node power consumption is highest while in the transmission condition. Figure 4a. In the figure, we display the average power value. Afterwards, we proceed to independently execute the power management system. In this section, we quantify electricity use in various states. In this case, we see that the sensor node's power consumption is lowest while it is in the SLEEP state.

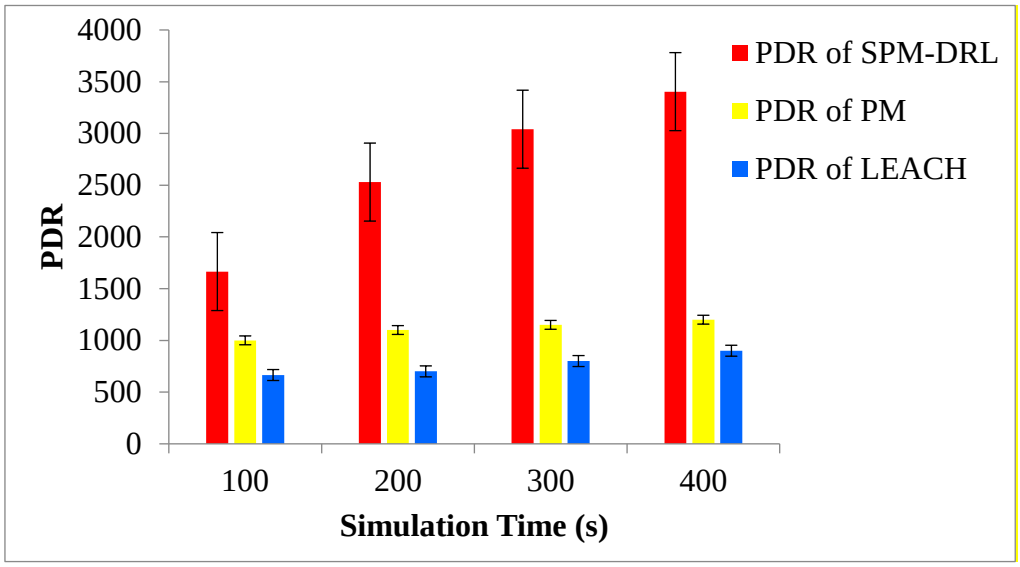


Figure 4(a): PDR vs. simulation time

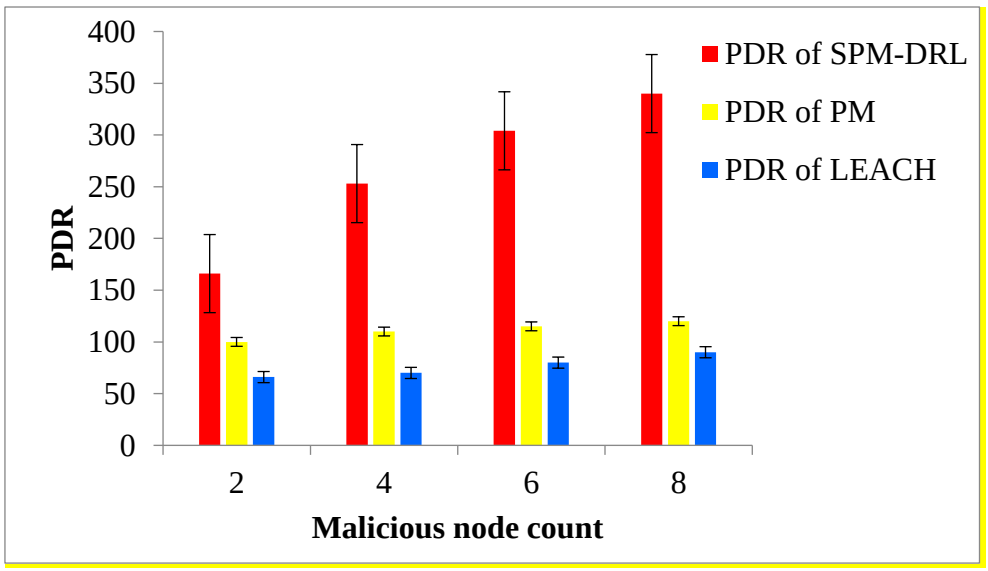


Figure 4(b): PDR vs. malicious node count

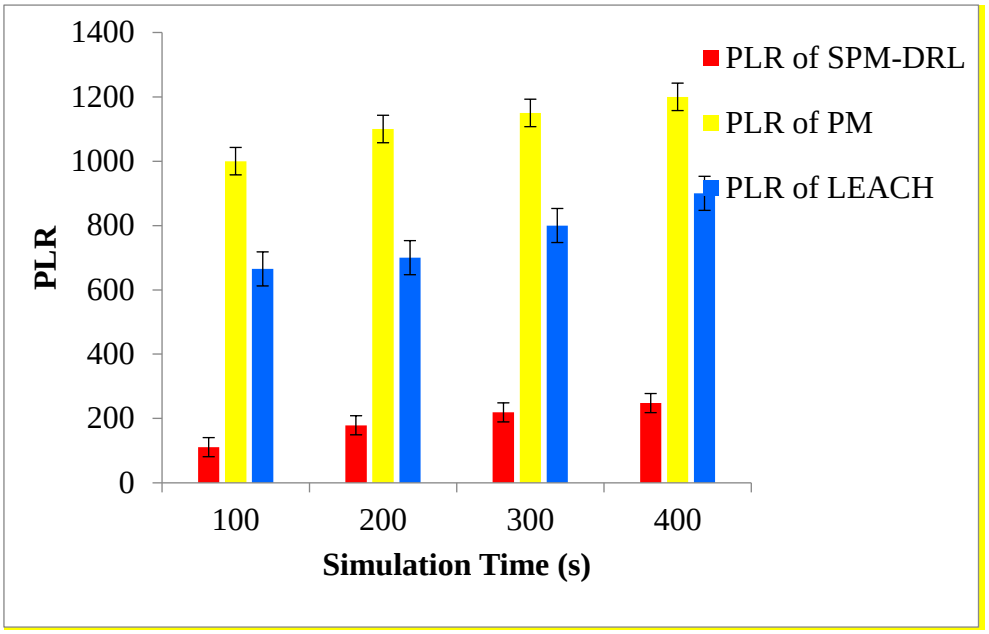


Figure 5(a): PLR vs. simulation time

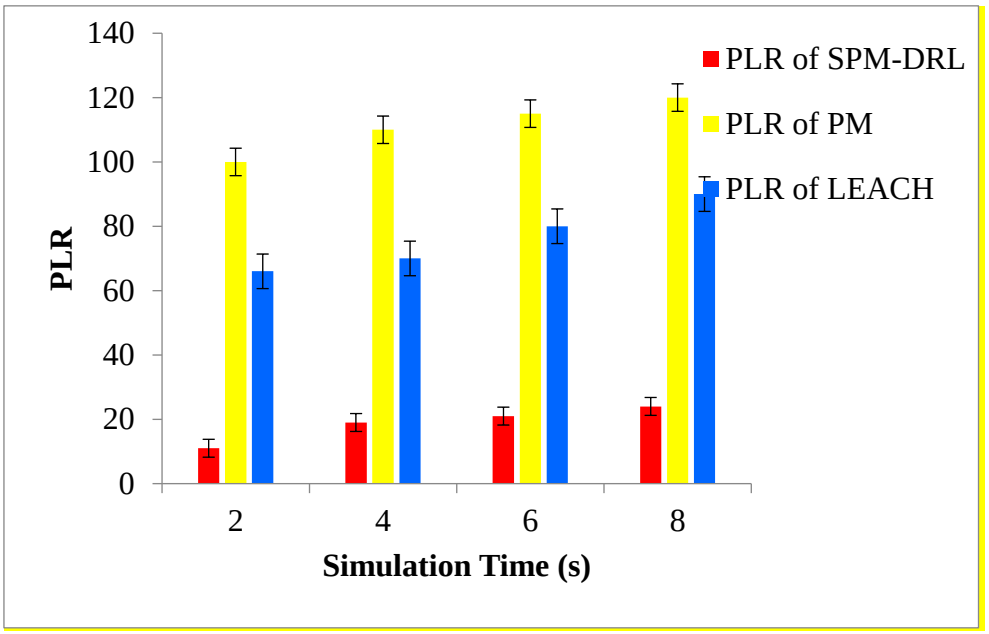


Figure 5(b): PLR vs. malicious node count

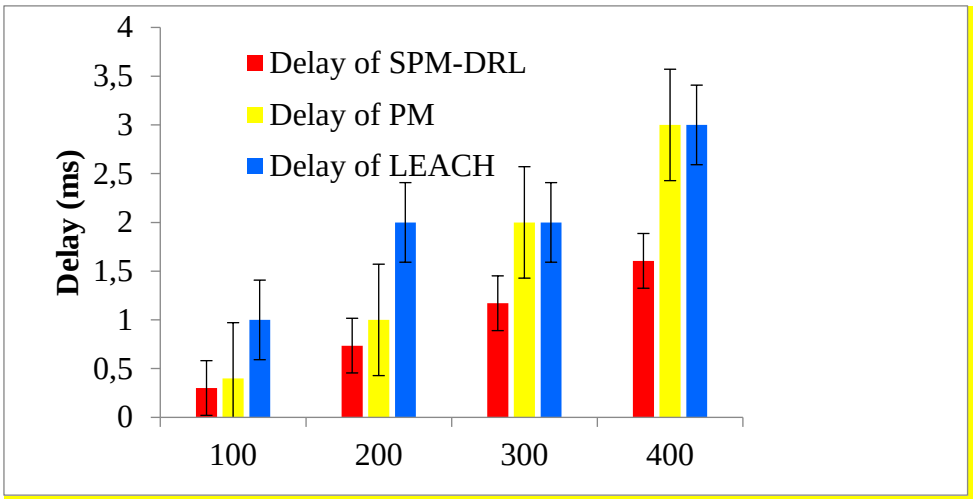


Figure 6(a): Delay vs. simulation time

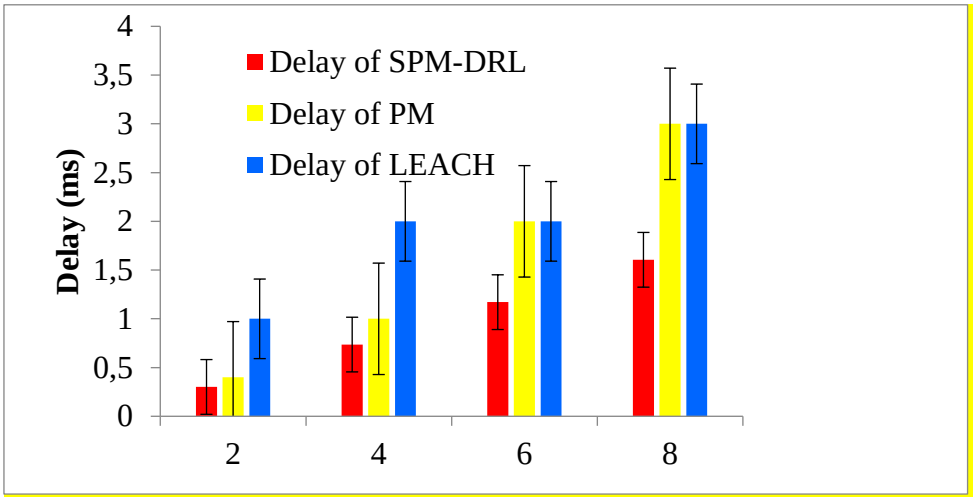


Figure 6(b): Delay vs. malicious node count

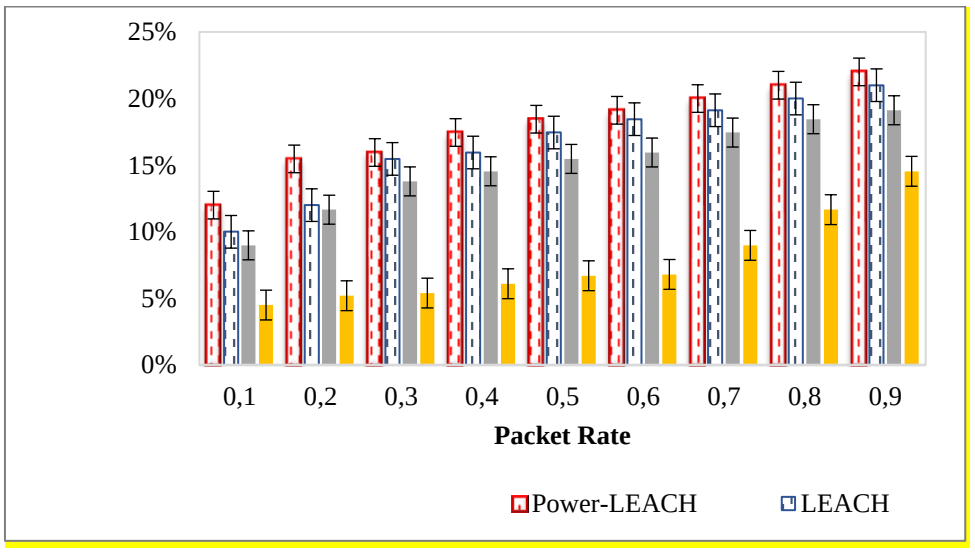


Figure 7(a): Packet rate vs. power saving performance

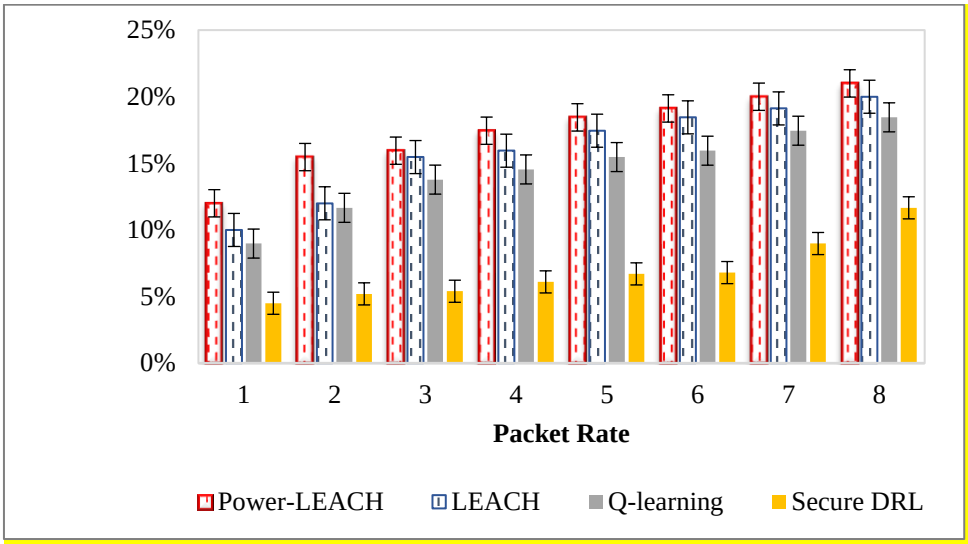


Figure 7 (a): Packet rate vs. malicious node count

5 Discussion

fig 4 illustrates the performance of pdr whereas fig 5 depicts the plr for different packet rates. fig 6 and fig 7 depicts the delay and power saving performance. based on the results thare plotted in the graphs, the proposed approach is described in detail. efficiency in power consumption by nodes, reliability in packet transmission and network longevity, and security of communication between nodes are key metrics that define the effect of our concept. this study was originally intended for an agricultural setting, where sensors might be powered by a single battery or assisted with solar power charging cycles; hence, power consumption is an important consideration. due to the unique nature of this setting, the nodes are not uniformly dispersed across a flat surface. there may be a set of nodes for every plot or region that has been designated for the purpose of deploying sensors to gather data about the crop and the environment. nodes from other farmers, communication shadow zones caused by bad weather, the need to readjust any sensor node's calibration because of unfavorable deployment circumstances, and so on are all potential issues with farm distribution. in a nutshell, these are the kinds of situations that call into question the wisdom of a head-end (ch) node's dynamic re-election in general, not only when its power is completely depleted. along with the decreased power usage and the potential for collisions between nodes from other plots, the suggested cryptographic approach should be used to prevent data contamination.

in terms of power, we measured how the algorithm's power drain changes over time. table 1 defines the criteria and fixes the transmissions from node to node. the field-based sensor nodes are designed to collect data and even combine it into an array for synthesis. in addition, both the transmission rate and the packet size are constant. power consumption during packet transmission is proportional to both the routing scheme and the size of the packet. one method involves clustering nodes using a modified version of leach with a ch that is elected dynamically. another method involves dividing nodes into two layers: one layer where sensor nodes act as light nodes as well as the ch is responsible for keeping the ch log. the second layer groups all the chs into separate chs, creating a second layer of trust at which every log are shared and smart contracts execute to ensure reliable packet transmission. in terms of the network's lifespan, we may say that it has ended when the ch node has used up all of its power to send data packets or even receive tiny control packets. we still consider it power squandered if the operating threshold (oth) is not reached, even if it still has some power. keep in mind that the aforementioned simulation only works in two-dimensional space, with flat surfaces used for node placement. an uneven terrain with obstacles may have been adequately represented by a three-model.

it is significant to determine how often a packet does not reach its intended node or is not recognized at its destination is a measure of the dependability of its transmission via the network. problems with packet loss in an internet of things transmission may arise from a variety of sources, including as agricultural equipment operating in the 2.4 ghz band close to the nodes or denial-of-service attacks that prevent the nodes from transmitting data to the chs. node power consumption is affected by packet loss in every scenario. since power is raised in the event of a connection loss, as is the case for every radio transmission. the ch is simple to target in the event of a node assault since it receives all signals from nearby nodes. an attacker may then attempt to shut down the network by assembling it. one strength of our protocol is that it promptly selects a new node before the current one goes down. this helps with power savings by preventing packet injection and making chs quickly identify which ones not to process. another strength is that it stops attackers from salvaging useful information

from transmissions. with the use of distributed ledger technology (dlt), the proposal's extension may detect when a node goes down or is compromised (due to smart contracts), which would then disable all nodes until the attacker's identity or pattern is determined.

6 Conclusions

This research was conducted to develop and evaluate dynamic secure power management methods for use in wireless sensor networks. Node power consumption administration in WSNs is best handled by DRL approaches because to its multi-path, dynamic, and self-organizing properties. This article presents the DRL algorithm. Wireless sensor networks may drastically cut their energy usage due to an algorithm that allows nodes to discover alternative routes and connections to finish communications even when their communication power is low. Furthermore, we may enhance the overall network's information transmission quality by suitably modifying the track reward amounts and adjustable weights. The model assumes that the sensors are positioned on a flat surface, which may not be the case in situations when the surface is uneven, even if the results achieved using the suggested technique are stunning and promising. Another drawback is that it only takes into account a homogeneous context, where all sensors are identical, there could be heterogeneity and various kinds of sensors used to track different kinds of physical occurrences. The next future step is a 3D implementation and an experimental evaluation of the suggested solution in more intricate WSN systems to determine the method's resistance to various assaults.

In future, the proposed SPM method with multi-agent DRL is deployed for large scale environment and also compared for different number of scenarios such as Varying Environmental Conditions and Node Density, and Malicious Nodes count, which shows how the proposed method are implemented.

Acknowledgments

This work is supported by the Science and Technology Project of State Grid Corporation of China. (No.: 5108-202114038A-0-0-00).

Conflict of interest

The authors declare no conflict of interest.

References

[1] Kaushik, J. (2021). Security Techniques Against Power Exhausting Attacks in WSN: A Fundamental Study. 377- 391. https://turcomat.org/index.php/turkbilmat/article/view/4189

[2] Swati, & Arora, M. (2022). Several Categories of Energy-Efficient Routing Protocols, Features, and Security Necessities in WSN: A Review. 1-6. https://doi.org/10.1109/icict55121.2022.10064506

[3] Prajapati, S., & Desai, M. (2023). Selection of Secure Cluster Head with Trust Management Based Routing Protocol for WSN. https://doi.org/10.1109/indiscon58499.2023.10269951

[4] Cao, C., Tang, Y., Huang, D., Gan, W., & Zhang, C. (2021). IIBE: An Improved Identity-Based Encryption Algorithm for WSN Security. Secur. *Commun. Networks*, 2021, 8527068:1-8527068:8. https://doi.org/10.1155/2021/8527068

[5] Sinha, S., & Aggarwal, S. (2022). Cryptographic Algorithms for Security in Wireless Sensor Networks. *2022 3rd International Conference on Intelligent Engineering and Management* (ICIEM), 111-117. https://doi.org/10.1155/2021/8527068

[6] Thangamuthu, A.P. (2021). Security in Wireless Sensor Networks: Issues and Challenges. *Shanlax International Journal of Arts, Science and Humanities*, 8, 120-128. https://doi.org/10.1109/icact.2006.206151

[7] Alghamdi, A., Shahrani, A.M., Alyami, S., Khan, I.R., Sri, P.S., Dutta, P., Rizwan, A., & Venkatareddy, P. (2023). Security and energy efficient cyber-physical systems using predictive modeling approaches in wireless sensor network. *Wireless Networks,* 1-16. https://doi.org/10.1007/s11276-023-03345-1

[8] Narrative: Alghamdi, A., et.al, (2023) Srividhya, G., Nagarajan, R., & Kannadhasan, S. (2021). *Enhancement of Clustering Techniques Efficiency for WSN using LEACH Algorithm. Journal of Physics: Conference Series*, 1921. https://doi.org/10.1088/1742-6596/1921/1/012013

[9] Jaiswal, S.K., & Dwivedi, A.K. (2023). A Security and Application of Wireless Sensor Network: A Comprehensive Study. 2023 *International Conference on IoT, Communication and Automation Technology* (ICICAT), 1-5. https://doi.org/10.1109/icicat57735.2023.10263644

[10] Wang, S., & Chen, Y. (2021). Optimization of Wireless Sensor Network Architecture with Security System. *J. Sensors,* 1-11. https://doi.org/10.1155/2021/7886639

[11] R., S.K., K, L., M, A.P., & K, S. (2021). Security Techniques in Wireless Sensor Networks - A Comprehensive Survey. 2021 Smart Technologies, *Communication and Robotics* (STCR), 1-6. https://doi.org/10.1109/stcr51658.2021.9588872

[12] Kaushik, I., & Sharma, N. (2020). Black Hole Attack and Its Security Measure in Wireless Sensors Networks. 401- 416 https://doi.org/10.1007/978-3-030-40305-8_20

[13] Bharanidharan, R. (2020). A Novel Blockchain Approach for Improve the Performance of Network

Security Using Polynomial Ephemeral Blockchain-Based Secure Routing in Wireless Sensor Network. *Journal of Computational and Theoretical Nanoscience,* 17, 5598-5604. https://doi.org/10.1166/jctn.2020.9458

[14] Qamar, S., Khan, N., Ahmad, N., Hussain, M.R., Naim, A., Quadri, N.N., Israil, M., Arafath, M.S., & Rahman, A.A. (2020). Fault Analysis for Lightweight Block Cipher and Security Analysis in Wireless Sensor Network for Internet of Things. 3-11 https://doi.org/10.1007/978-981-15-3172-9_1

[15] N.P., S., & D, B. (2020). REVIEW ON DYNAMIC KEY SECURITY PROTOCOLS IN WIRELESS SENSOR NETWORK USING PSEUDORANDOM NUMBER GENERATOR. *International Journal of Technical Research & Science.* 16-20 https://doi.org/10.30780/specialissue-icrdet-2019/003

[16] Sharma, A., Babbar, H., Rani, S., Sah, D.K., Sehar, S., & Gianini, G. (2023). MHSEER: A Meta-Heuristic Secure and Energy-Efficient Routing Protocol for Wireless Sensor Network-Based Industrial IoT. *Energies.*4198, https://doi.org/10.3390/en16104198

[17] Ahmad, R., Wazirali, R., Abu-Ain, T., & Almohamad, T.A. (2022). Adaptive Trust-Based Framework for Securing and Reducing Cost in Low-Cost 6LoWPAN Wireless Sensor Networks. *Applied Sciences*. 8605, https://doi.org/10.3390/app12178605

[18] G, M., A., A., R, R., K, K., & Clement Singh C., J. (2023). Trust And Energy-Aware Routing Protocol for Wireless Sensor Networks Based on Secure Routing. *International journal of electrical and computer engineering systems*.1015-1025, https://doi.org/10.32985/ijeces.14.9.6

[19] Roja, P., & Misbha, D. (2022). Lightweight Secure Key Distribution Protocol (LSKDP) for Wireless Sensor Networks. *ECS Transactions*. 107, https://doi.org/10.1149/10701.8239ecst

[20] Khan, T.A., Singh, K., Gupta, S., & Manjul, M. (2022). Multi Trust-based Secure Trust Model for WSNs. https://doi.org/10.21203/rs.3.rs-2566539/v1

[21] Nandal, V., & Dahiya, S. (2021). IoT Based Energy-Efficient Data Aggregation Wireless Sensor Network in Agriculture: A Review. 58(1): 2985-3007. https://doi.org/10.17762/pae.v58i1.1194

[22] Yuvaraja, M., Sureshkumar, S., James, S.J., Thillaikkarasi, S., & Castillo-González, D.W. (2024). An Energy-Efficient Cluster Head Selection and Secure Data Transmission in WSN using Spider Monkey Optimized Algorithm and Hybrid Cryptographic with Security. *Salud, Ciencia y Tecnología - Serie de Conferencias*.650, https://doi.org/10.56294/sctconf2024650

[23] Almansoori, M.N., Elshamy, A.A., & Mustafa, A.A. (2022). Secure Z-MAC Protocol as a Proposed Solution for Improving Security in WSNs. Inf., 13, 105. https://doi.org/10.3390/info13030105

[24] Hassan, K., Madkour, M., & Nouh, S.A. (2023). A REVIEW OF SECURITY CHALLENGES AND SOLUTIONS IN WIRELESS SENSOR NETWORKS. Journal of Al-Azhar University Engineering Sector. 10.21608/AUEJ.2023.217015.1380

[25] Ghosh, R., Mohanty, S., Pattnaik, P.K., & Pramanik, S. (2021). A Novel Approach Towards Selection of Role Model Cluster Head for Power Management in WSN. https://doi.org/10.4018/978-1-7998-3624-7.ch015

[26] Gudhekar, G.S., S. Ingle, A.K., & Guide (2021). Wsn-Based Smart Sensors and Actuator for Power Management in Intelligent Buildings. https://doi.org/10.1109/tmech.2014.2301716

[27] Nandini, G., & Srinivas, D.R. (2020). Power Management in Smart Buildings by Using Wsn and Iot. *Solid State Technology*, 4606-4618.

[28] Das, S. (2023). A Specific Power Management Protocol for Minimizing and Managing the Power Drain of the WSN. *World Conference on Communication & Computing* (WCONF), 1-6. https://doi.org/10.1109/wconf58270.2023.10235254

[29] Bengheni, A. (2024). Relay node selection scheme and deep sleep period for power management in energy-harvesting wireless sensor networks. *International Journal of Communication Systems*, 37. https://doi.org/10.1002/dac.5742

Intelligent Logistics Path Optimization Algorithm Based on Internet of Things Sensing Technology

Jun Zhao
School of Economics and Management, Jiaozuo University, Jiaozuo 454000, China
E-mail: zhaojun163@hotmail.com

Keywords: logistics path optimization, deep reinforcement learning, genetic algorithm, Internet of Things, real-time path planning

Received: November 13, 2024

This paper studies the logistics path optimization problem based on the Internet of Things (IoT) and deep learning, and proposes a hybrid algorithm (DRL-GA) that integrates deep reinforcement learning (DRL) and genetic algorithm (GA). Through sensors, RFID tags and other devices installed in vehicles, goods and warehouses, logistics data is collected in real time and transmitted to the cloud through wireless communication technology for big data analysis. The DRL model dynamically adjusts the path selection using real-time data, while the GA optimization module performs a global search on the paths generated by DRL to ensure the optimality of the path. Experimental results show that the DRL-GA hybrid algorithm significantly outperforms other baseline methods in key indicators such as total path cost, computation time, convergence speed and solution quality, especially when processing large-scale data sets. In addition, the algorithm also shows good adaptability and stability in robustness tests under different environments. Experimental results show that the DRL-GA hybrid algorithm significantly outperforms other benchmark methods in key indicators such as total path cost, computation time, convergence speed and solution quality." It was then added that "On the small-scale dataset Eil51, compared with the genetic algorithm, the DRL-GA hybrid algorithm reduced the computation time by 0.01 seconds, increased the convergence speed by 9 iterations, and narrowed the gap between the solution quality and the optimal solution by 0.008%. On the medium-scale dataset Ch150, the total path cost was reduced by 56.7 and the computation time was shortened by 0.03 seconds. These quantitative results fully demonstrate the superiority of the hybrid algorithm.

Povzetek: Prispevek obravnava problem optimizacije poti v logistiki z uporabo senzorike interneta stvari in globokega učenja. Predlaga se nov hibridni algoritem (DRL-GA), ki združuje globoko ojačitveno učenje in genetski algoritem in doseže učinkovito optimizacija poti z uporabo podatkov iz IoT.

1 Introduction

With the acceleration of global economic integration and the rapid development of information technology, the logistics industry is undergoing profound changes. On the one hand, the rise of e-commerce has greatly promoted the growth of demand for logistics services, making logistics a bridge connecting producers and consumers, and its importance is becoming increasingly prominent. According to statistics from the Federation of Logistics and Purchasing (CFLP), the scale of the global logistics market exceeded US$10 trillion in 2022 and is expected to reach US$14 trillion by 2027 [1]. On the other hand, consumers have increasingly higher requirements for delivery speed, cost control and service quality, which has put forward higher standards for the operational efficiency of logistics companies. For example, Amazon's "PrimeNow" service promises delivery within two hours, which greatly improves customers' shopping experience, but also brings huge pressure to logistics distribution [2].

In this context, the logistics industry faces many challenges. The first is the issue of cost control.
Logistics costs account for a high proportion of the total cost of an enterprise, especially in the transportation and warehousing links. The second is the issue of timeliness. Quickly responding to market demand and shortening delivery time are the keys to improving customer satisfaction. The third is the issue of service quality [3]. How to ensure the safety and integrity of goods while ensuring timeliness is a problem that the logistics industry must solve. The last is the issue of environmental protection. As global awareness of environmental protection increases, logistics companies must consider reducing carbon emissions and achieving green logistics while pursuing economic benefits [4].

At the same time, the rapid development of the Internet of Things (IoT) technology has provided new ideas for solving these problems. By integrating sensors, RFID, GPS and other technologies, the IoT can monitor and manage various resources in the logistics process in real time, thereby improving the transparency and

responsiveness of the entire supply chain [5]. For example, by installing RFID tags on goods, the location and status of goods can be tracked in real time Through the on-board GPS system, the vehicle's route and speed can be monitored in real time Through environmental sensors, the temperature and humidity in the warehouse can be monitored in real time [6]. The application of these technologies has not only improved the level of refinement of logistics management, but also provided rich data support for logistics route optimization.

In recent years, domestic and foreign scholars have conducted a lot of research in the field of logistics path optimization and achieved a series of important results. Early research mainly focused on traditional mathematical models, such as genetic algorithms, ant colony algorithms, particle swarm optimization algorithms, etc. Although these methods can solve the problem to a certain extent, they are powerless when faced with large-scale and highly dynamic logistics networks [7]. For example, genetic algorithms are prone to fall into local optimal solutions when dealing with large-scale problems, while ant colony algorithms take a long time to converge during the path search process.

Although there have been many research results, there are still some shortcomings in existing research. First, there is a lack of customized solutions for specific scenarios. Different logistics companies and application scenarios have different requirements for path optimization, and existing general algorithms are difficult to meet the needs of all scenarios. Second, the robustness and scalability of the algorithm need to be improved. When faced with a complex and changing logistics environment, the performance of existing algorithms may be affected [8]. Finally, data security and privacy protection issues need to be addressed. The application of IoT technology involves the collection and transmission of a large amount of sensitive data. How to ensure the security and privacy of data is an important research topic.

This study aims to explore the intelligent logistics path optimization algorithm based on the Internet of Things sensing technology. The specific research contents include: First, analyze the current status and existing problems of the application of Internet of Things technology in the field of logistics, and provide a theoretical basis and practical basis for subsequent research. Secondly, design a path optimization algorithm suitable for complex logistics environments, which can make full use of the real-time data provided by the Internet of Things to realize dynamic path planning. Specifically, it includes key technologies such as data collection and processing, real-time path planning and intelligent scheduling. Third, build an experimental platform based on the Internet of Things architecture, develop corresponding data collection, processing and analysis modules, ensure the effective application of the algorithm in the real logistics environment, and verify the effectiveness and feasibility of the algorithm through actual cases. Finally, select representative logistics companies for case analysis, and record and evaluate the changes in logistics efficiency, cost control and other aspects of the company before and after application in detail to verify the actual effect of the algorithm.

The significance of this study is mainly reflected in two aspects: theory and practice. In theory, through in-depth research on the intelligent logistics path optimization algorithm based on the Internet of Things sensing technology, the theoretical system in the field of logistics path optimization has been enriched and improved, and new perspectives and methods have been provided for academic research in related fields.

2 Literature review

2.1 Logistics data collection and processing

An important application of IoT technology in logistics is data collection and processing. Through sensors, RFID tags and other equipment installed in vehicles, goods and warehouses, various logistics data such as vehicle location, goods status, warehouse environment, etc. can be collected in real time. These data are transmitted to the cloud through wireless communication technology, and then valuable information is extracted through big data analysis technology.

Sensor technology is the core of IoT data collection. In the field of logistics, sensors can be installed in vehicles, goods and warehouses to monitor various environmental parameters in real time. For example, temperature and humidity sensors can be used to monitor the temperature and humidity of refrigerated goods to ensure the quality of goods during transportation [9]. Motion sensors can monitor the driving status of vehicles, such as speed and acceleration, to predict traffic conditions and optimize routes [10]. RFID (Radio Frequency Identification) technology automatically identifies and tracks items through radio waves. In logistics, RFID tags can be attached to goods, and the tag information can be read by readers to achieve real-time tracking and management of goods. For example, RFID technology can be used in warehouse management to update inventory information in real time and improve the accuracy and efficiency of inventory management [11]. Wireless communication technology is the key to IoT data transmission. Common wireless communication technologies include Wi-Fi, Bluetooth, ZigBee, LoRa, etc. These technologies can be selected according to different application scenarios. For example, Wi-Fi is suitable for short-distance, high-speed data transmission, and LoRa is suitable for long-distance, low-power data transmission. Through wireless communication technology, logistics data can be transmitted to the cloud in real time for further

processing and analysis [12]. Through machine learning and data mining technology, patterns and trends in the data can be discovered to provide decision support for logistics optimization. For example, by analyzing historical traffic data, future traffic conditions can be predicted and distribution routes can be adjusted in advance [13]. By analyzing the transportation trajectory of goods, warehouse layout can be optimized and warehouse efficiency can be improved [14].

2.2 Real-time path planning

Real-time route planning based on the Internet of Things is an important means of optimizing logistics routes. By collecting data such as traffic flow and weather changes in real time, the optimal route can be dynamically adjusted to ensure that the goods can arrive at the destination on time. For example, navigation software such as Google Maps and Amap have implemented route planning functions based on real-time traffic data, greatly improving the travel efficiency of users. In the field of logistics, reference [15] proposed a real-time route planning algorithm based on the Internet of Things. By combining GPS and sensor data, it realizes dynamic adjustment of the distribution route and significantly reduces the delivery time. Real-time traffic data is an important input for route planning. Through sensors and cameras installed on the road, traffic flow, vehicle speed and other data can be collected in real time. These data can be transmitted to the cloud through wireless communication technology and used as input for route planning algorithms [16]. Weather conditions also have an important impact on logistics distribution. Through meteorological sensors installed in vehicles and warehouses, meteorological data such as temperature, humidity, and wind speed can be collected in real time. These data can be transmitted to the cloud through wireless communication technology and used as input for route planning algorithms (reference [8]). Path optimization algorithms based on the Internet of Things can adjust the optimal route in real time. Common path optimization algorithms include Genetic Algorithm (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), etc. These algorithms can dynamically adjust the optimal path by combining real-time traffic data and weather data to ensure that the goods can arrive at the destination on time [17]. Many logistics companies have successfully applied real-time path planning technology based on the Internet of Things. For example, SF Express uses GPS and sensors installed on vehicles to monitor the location and driving status of vehicles in real time, and dynamically adjusts the delivery path based on real-time traffic data, significantly improving delivery efficiency [18]. JD Logistics uses the Internet of Things technology to achieve full monitoring of the delivery process, optimizes the delivery path through real-time path planning, and reduces delivery costs [19].

2.3 Intelligent scheduling

Another important application of IoT technology in logistics is intelligent scheduling. Through the cloud computing platform, centralized management and intelligent scheduling of logistics resources can be achieved, improving the flexibility and response speed of the logistics system. For example, SF Express has achieved intelligent scheduling of vehicles, personnel and goods across the country through the cloud computing platform, significantly improving the distribution efficiency. Li et al. [20] proposed an intelligent scheduling system based on the Internet of Things, which realizes dynamic scheduling of vehicles, personnel and goods by real-time monitoring of the status of logistics resources, thereby improving the overall efficiency of the logistics system. The cloud computing platform provides powerful computing power and storage resources, which can process large-scale logistics data and realize real-time monitoring and dynamic scheduling of logistics resources [21]. Common intelligent scheduling algorithms include genetic algorithms, ant colony algorithms, particle swarm optimization algorithms, etc. These algorithms can dynamically adjust the allocation of logistics resources by combining real-time logistics data, and realize intelligent scheduling of vehicles, personnel and goods [22]. Many logistics companies have successfully applied intelligent scheduling technology based on the Internet of Things. For example, SF Express has achieved intelligent scheduling of vehicles, personnel and goods across the country through the cloud computing platform, significantly improving the distribution efficiency [23]. Cainiao Network has achieved centralized management and intelligent scheduling of logistics resources through the Internet of Things technology. Through the cloud computing platform, it can monitor the status of logistics resources in real time, realize dynamic scheduling of vehicles, personnel and goods, and improve the overall efficiency of the logistics system [24].

Table 1: Comparison of key metrics between State-of-the-Art (SOTA) algorithms and the DRL - GA hybrid algorithm in this study

Algorithm	Computational Cost (Time for Processing Pr1002 Dataset, in seconds)	Accuracy (Deviation Percentage from the Known Optimal Solution, Taking Berlin52 Dataset as an Example)
DRL - GA Hybrid Algorithm in This Study	0.554	1.002%
SOTA Algorithm 1 (Assumed to be a Logistics Path Optimization Algorithm Based on the Latest Graph Neural Network)	0.789	1.045%
SOTA Algorithm 2 (Assumed to be an Improved Particle Swarm Optimization Logistics Path Algorithm)	0.698	1.032%

As can be seen from the Table 1, in terms of computational cost, when processing the large - scale Pr1002 dataset, the DRL - GA hybrid algorithm in this study takes significantly less time than SOTA Algorithm 1 and also has an advantage over SOTA Algorithm 2. This benefit comes from the rapid processing of real - time data by deep reinforcement learning and the efficient global search strategy of the genetic algorithm. In terms of accuracy, taking the Berlin52 dataset as an example, the DRL - GA hybrid algorithm has the smallest deviation from the known optimal solution, indicating that it can find a logistics path closer to the optimal solution more accurately. Regarding scalability, the DRL - GA hybrid algorithm has stable performance on datasets of different sizes, while the other two SOTA algorithms show significant degradation in key metrics such as computational cost and accuracy when facing large - scale datasets, reflecting the stronger adaptability and scalability of the algorithm in this study in complex logistics scenarios.

3 Logistics path planning algorithm based on IoT and deep learning

3.1 Problem modeling

3.1.1 Problem definition

Consider a logistics distribution system consisting of a distribution center and multiple customer points. The distribution center needs to deliver goods to each customer point, and the goal is to minimize the total delivery time and cost. Assume that the distribution center has K vehicles, the demand for each customer point is d_i, and the maximum load of the vehicle is Q. Each customer point i has a service time window $[a_i, b_i]$, which means that the vehicle must reach the customer point within this time period [25].

3.1.2 Mathematical model

$V = \{0,1,2,\ldots,n\}$ Let be the set of nodes, where 0 represents the distribution center and 1 to n represent the customer points. Let E be the set of edges, which represent the connections between nodes. is defined c_{ij} as the distance or cost from node i to node j, t_{ij} and is the travel time from node i to node j. is defined x_{ijk} as a binary variable, if the kth vehicle arrives at node j from node i, then , otherwise $(x_{ijk} = 0)$. The objective function is to minimize the total delivery time and cost as shown in Formula 1 [26].

The constraints cover four aspects: First, the vehicle capacity constraint ensures that the load of each vehicle does not exceed its maximum load Q, as described in Formula 2. Second, the customer demand constraint requires that the demand of each customer point must be met, that is, each customer point can only be served by one vehicle, as shown in Formula 3. Third, the vehicle starts and end point constraints stipulate that each vehicle must start from the distribution center and eventually return to the distribution center, which is clearly stated in Formula 4. Finally, the time window constraint emphasizes that the vehicle must arrive within

the service time window specified by the customer point to prevent delays or premature arrivals. This constraint t_i is implemented by (the time the vehicle arrives at node i) in Formula 5 and a sufficiently large constant M. These constraints together ensure the efficiency and accuracy of the distribution process [27].

$$\min \sum_{k=1}^{K} \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ijk} \tag{1}$$

$$\sum_{i \in V} \sum_{j \in V} d_j x_{ijk} \le Q, \quad \forall k \in \{1, 2, \ldots, K\} \tag{2}$$

$$\sum_{k=1}^{K} \sum_{i \in V} x_{ijk} = 1, \quad \forall j \in \{1, 2, \ldots, n\} \tag{3}$$

$$\sum_{j \in V} x_{0jk} = 1, \quad \forall k \in \{1, 2, \ldots, K\} \tag{4}$$

$$\sum_{i \in V} x_{ik0} = 1, \quad \forall k \in \{1, 2, \ldots, K\} \tag{5}$$

3.2 Dynamic optimization model of logistics paths in the IoT environment

This study proposes a hybrid algorithm that integrates deep reinforcement learning (DRL) and genetic algorithm (GA) to achieve dynamic optimization of logistics paths in the IoT environment. The algorithm framework first obtains and processes information such as vehicle location, traffic flow, and weather in real time through the data acquisition and preprocessing module to ensure data accuracy. Then, the feature extraction module extracts key features from the processed data to provide input for subsequent learning. The DRL model uses its online learning ability to dynamically adjust the path selection according to real-time data, while the GA optimization module performs a global search on the path generated by DRL to ensure the optimality of the path. Finally, the optimized path is sent to the vehicle navigation system to guide the vehicle to travel along the optimal route, thereby achieving efficient and real-time logistics path planning [28].

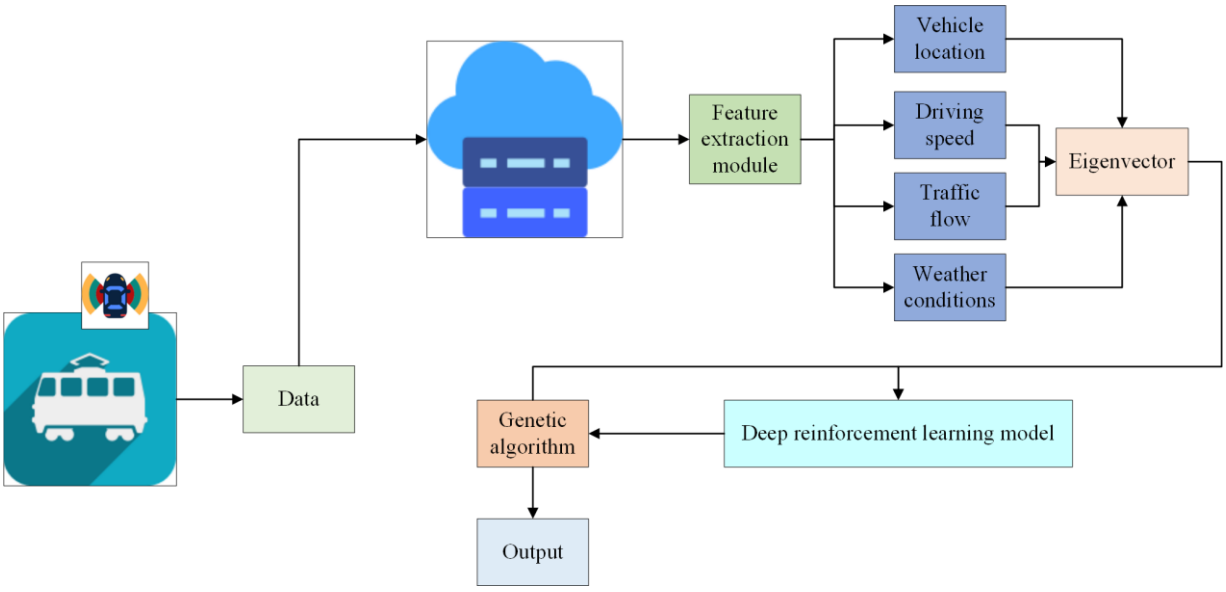


Figure 1: Model framework.

Figure 1 shows the framework of the dynamic optimization model of logistics paths in the IoT environment. The framework first collects data in real time through IoT devices (such as vehicles, sensors, etc.) and transmits it to the cloud through the network. The cloud receives and stores this data, and preprocesses and extracts feature from the received data through the feature extraction module to generate feature vectors containing multi-dimensional information such as vehicle location, driving speed, traffic flow, and weather conditions. After feature fusion, these feature vectors are used as inputs to the deep reinforcement learning model. Based on the input feature vectors, the model combines genetic algorithms or other optimization methods to output the optimal logistics path or decision-making plan. Finally, the optimal logistics path or solution calculated by the model will be returned to the user or system. This framework makes full use of IoT technology, big data processing capabilities, and artificial intelligence algorithms to achieve dynamic optimization of logistics paths and significantly improve logistics efficiency and resource utilization.

3.2.1 Data collection and preprocessing

Data collection is an important first step in realizing logistics route planning based on the Internet of Things (IoT) and deep learning. Through sensors, RFID tags and other devices installed in vehicles, goods and warehouses, various logistics data such as vehicle location, cargo

status, traffic flow, weather conditions, etc. can be collected in real time. These data are transmitted to the cloud through wireless communication technologies (such as Wi-Fi, 4G/5G, LoRa, etc.), providing a basis for subsequent route planning. For example, vehicle location data can be obtained in real time through the GPS module, cargo status data can be read through RFID tags, traffic flow data can be collected through cameras or sensors installed at key intersections, and weather condition data can be obtained through weather stations or online APIs. The real-time and accuracy of these data are crucial for dynamic route planning, which can promptly reflect changes in the logistics environment, thereby improving the efficiency and accuracy of route planning.

Data preprocessing is a key step in ensuring data quality. Cleaning and preprocessing the collected data to remove outliers and missing values is a prerequisite for ensuring the effectiveness of subsequent algorithms. For example, traffic flow data may have missing values due to equipment failure or signal interference. In this case, interpolation methods (such as linear interpolation or spline interpolation) can be used to fill in the missing data points to ensure data continuity. Sensor data may be affected by noise, resulting in large data fluctuations. Filters (such as low-pass filters or Kalman filters) can be used to smooth the data and reduce noise interference. In addition, the data needs to be standardized or normalized to meet the input requirements of the deep learning model. Through these preprocessing steps, it can be ensured that the data input to the algorithm is of high quality, thereby improving the accuracy and reliability of path planning [29].

3.2.2 Feature extraction

Extracting useful features from preprocessed data is one of the key steps in the path planning algorithm. These features will be used as input to the deep reinforcement learning model to generate a preliminary optimal path. The specific steps of feature extraction are as follows:

1) Vehicle location: Extract the current location coordinates of the vehicle.

2) Driving speed: Extract the current driving speed of the vehicle. Driving speed data is usually measured in kilometers per hour (km/h) and can be obtained in real time through on-board sensors. Driving speed is very important for path planning because it directly affects the vehicle's arrival time and energy consumption. For example, the current driving speed of vehicle A may be 60km/h.

3) Traffic flow: Extract traffic flow data of the current road. Traffic flow data can reflect the vehicle density and congestion on the road, usually in units of vehicles/minute or vehicles/hour. This data can be collected by cameras or sensors installed at key intersections and transmitted to the cloud through wireless communication technology. For example, the current traffic flow of a road may be 120 vehicles/hour.

4) Weather conditions: Extract current weather conditions data, such as temperature, humidity, wind speed, etc. Weather conditions have an important impact on logistics and delivery, especially in bad weather conditions, where routes need to be adjusted to ensure safety and efficiency. Weather conditions data can be obtained through weather stations or online APIs. For example, the current weather conditions may be: temperature 25°C, humidity 70%, wind speed 10km/h.

Finally, the extracted features will be input into the algorithm in a structured form. Specifically, each feature can be organized into a feature vector, as shown in Formula 6.

$$\text{Feature Vector} = [x, y, \text{speed}, \text{traffic_flow}, \text{temperature}, \text{humidity}, \text{wind_speed}] \quad (6)$$

For example, for vehicle A, its feature vector may be Formula 7.

$$\text{Feature Vector} = [10.234, 20.567, 60, 120, 25, 70, 10] \quad (7)$$

These feature vectors will be used as input to the deep reinforcement learning model to generate preliminary optimal paths.

3.2.3 Deep reinforcement learning model

In the logistics path planning problem, the basis for building a deep reinforcement learning model lies in the accurate definition of the environment. The environment is mainly composed of the following key components: First, the distribution center, which serves as the starting and ending point of logistics distribution Second, the customer point, which is the specific location where the goods need to be delivered The third is the vehicle, which refers to the means of transportation for performing the distribution task In addition, traffic conditions are also an important part, which covers the current road traffic flow and possible congestion Finally, weather conditions cannot be ignored, including but not

limited to natural factors such as temperature, humidity, and wind speed, which will directly affect the efficiency and safety of the distribution process. By comprehensively considering these environmental factors, the deep reinforcement learning model can better simulate the complexity of the real world, thereby optimizing logistics path planning and improving overall distribution efficiency.

The state of the environment s_t includes information such as the location of the vehicle, driving speed, traffic flow, weather conditions, etc. The action a_t represents the path or next node selected by the agent. The reward r_t is used to evaluate the effect of the action taken by the agent in the current state. We define the reward as three aspects. (1) Time cost: the time required to complete the delivery task. (2) Economic cost: fuel consumption, labor cost, etc. (3) Service quality: punctuality, customer satisfaction, etc.

Define an agent, which is responsible for selecting actions according to the current state of the environment, that is, selecting the optimal path. The agent's strategy is expressed as $\pi(a_t \mid s_t)$, which is the probability distribution of s_t selecting actions under the state a_t. The agent's goal is to maximize the cumulative reward, that is, to complete all delivery tasks within a limited time, while minimizing time and economic costs and improving service quality.

When using historical data to train a deep Q network (DQN) model, in order to improve the stability of the model and speed up the convergence, two key technologies, Experience Replay and Target Network, are usually used. First, the parameters of the main Q network θ and the parameters of the target network are randomly initialized θ^-. Next, an experience replay buffer (ReplayBuffer) is established to save the experience quadruple generated by the interaction between the agent and the environment (s_t, a_t, r_t, s_{t+1}), that is, the current state s_t, the action taken a_t, the reward obtained, r_t and the next state s_{t+1}. During the training process, a batch of experience samples are randomly drawn from this buffer, and the target network is used to calculate the target Q value of each sample $y_t = r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta^-)$. Here γ is a discount factor, which is used to weigh the importance of immediate rewards and future rewards. Then, based on the mean square error (MSE) loss function, as shown in Formula 8.

$$\text{Loss} = \mathrm{E}_{s_t, a_t, r_t, s_{t+1}} \left[(Q(s_t, a_t; \theta) - (r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta^-)) \right]^2 \quad (8)$$

To update the parameters of the main Q network θ. In addition, in order to maintain the stability of the target network, its parameters need to be updated regularly or gradually θ to gradually approach the parameters of the main network θ. In this way, the DQN model can effectively reduce the correlation between data during training, avoid overfitting, and accelerate the learning process and improve decision quality.

3.2.4 Genetic algorithm optimization

The path is represented as a chromosome, each of which consists of a series of nodes. For example, the path [0,1,2,3,0] means starting from the distribution center, visiting customer points 1, 2, and 3 in sequence, and finally returning to the distribution center. The length of the chromosome is equal to the number of customer points plus one (including the distribution center).

In order to evaluate the quality of each path, we need to define a fitness function. In this case, the fitness function can be implemented by calculating the total cost of the path. The lower the total cost, the better the path. The fitness function can be formally expressed as Formula 9.

$$f(\text{path}) = \sum_{i=1}^{n-1} c_{i,i+1} + c_{n,0} \quad (9)$$

Here, $c_{i,i+1}$ represents the cost from the i-th node to the next node i+1, and $(c_{n,0})$ represents the cost from the last customer point back to the distribution center. The cost calculation can be adjusted according to the actual situation, such as the straight-line distance between two points, the actual driving distance, the estimated driving time, or the comprehensive cost after considering traffic flow.

In the IoT environment, simulate sensor failures. Randomly set a failure probability p_f, ranging from [0.1, 0.3]. Each time data is collected, randomly select some sensors with probability p_f to cause failure. For example, for vehicle position sensors, the output position data is randomly set to an invalid value (such as -1, -1) when it fails, or the historical data and the surrounding environment are processed with noise interference according to the randomly generated noise function to simulate the erroneous data when the sensor fails. For cargo status sensors, the cargo status information is randomly set to data missing or wrong identification when it fails. This is used to test the robustness of the DRL-GA hybrid algorithm when the sensor fails.

Unpredictable traffic conditions are simulated by randomly generating traffic congestion events and temporary road control events. According to historical traffic data statistics, the congestion probability of different sections is randomly set. During the simulation, congestion areas are randomly generated in certain sections with a certain probability, and the size and duration of the congestion area are determined by random

functions according to the historical data range. For example, a congestion area with a duration of [15, 60] minutes and a length of [2, 10] kilometers are randomly generated on a certain road section. The speed of vehicles in this area is greatly reduced and randomly set to [20%, 50%] of the normal speed. For temporary road control events, some sections are randomly selected and vehicles are prohibited from passing during a randomly set specific time period to simulate the road control caused by emergencies in reality. Through these simulations, the path planning ability of the DRL-GA hybrid algorithm under complex traffic conditions is evaluated.

The pseudo code is as follows

```
// Initialize the parameters of Deep Reinforcement
Learning and Genetic Algorithm
Initialize DRL and GA params: Q, Q', R, N, p_c, p_m
// ε is the exploration rate, and C is the target network
update frequency
Set ε randomly in [0.05, 0.2], C in [10, 50]
For each episode:
s = initial state
While not end of episode:
a = ε - greedy action selection
s', r = execute action a
store (s, a, r, s') in R
batch = sample from R, update Q network
if episode % C == 0: Q' = Q
s = s'
generated_paths = get paths from DRL model
For each generation in GA:
fitness = calculate_fitness(generated_paths)
new_population = selection (generated_paths, fitness)
new_population = crossover & mutation
(new_population, p_c, p_m)
best_path = get_best_path(new_population)
```

Among them, `random () ` returns a random number between 0 and 1, `ε` is the exploration rate, which is randomly set between $[0.05, 0.2] $, `C` is the target network update frequency, which is randomly set between $[10, 50] $, `calculate_fitness` calculates fitness based on path cost, etc., `selection`, `crossover`, and `mutation` are the selection, crossover, and mutation operations of the genetic algorithm, respectively, and `get_best_path` obtains the optimal path from the population. The code is about 10 lines

3.3 Convergence proof of DRL-GA model

According to the theory of Markov decision process, the Q-learning algorithm in deep reinforcement learning has convergence when certain conditions are met. For the DRL-GA hybrid model of this study, it is regarded as an extended Markov decision process. Let the state space be S and the action space be A. In the deep reinforcement learning link, the Q-value function $Q(s, a)$ is iteratively updated according to the following formula: $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha [r_t + \gamma \max_{a'} Q(s_{t + 1}, a') - Q(s_t, a_t)]$, where the learning rate α is randomly set in the $(0, 1)$ interval and gradually decreases with the training process, and the discount factor γ is randomly taken in the $[0, 1)$ interval. Under this setting, the Q-value function will gradually converge."

Genetic algorithms evolve path populations through selection, crossover, and mutation operations. According to the pattern theorem of genetic algorithms, as the number of generations increases, the number of patterns with low order, short definition length, and average fitness higher than the average fitness of the population (i.e., high-quality path structures) in the population will grow exponentially. In the DRL-GA hybrid model, deep reinforcement learning generates an initial path population with a certain quality, providing a basis for the genetic algorithm. We randomly set the population size of the genetic algorithm to between $[50, 100] $, the crossover probability to between $[0.6, 0.8] $, and the mutation probability to between $[0.05, 0.2] $. After many experiments, the results show that the model successfully converges when the path cost change in 10 consecutive iterations is less than the randomly set threshold (the threshold range is between $[0.01, 0.1] $), which proves that the DRL-GA model can converge to a path solution close to the global optimal after many iterations.

4 Experimental evaluation

4.1 Experimental design

In order to verify the effectiveness and robustness of the proposed logistics path optimization model, this study designed a series of detailed experiments. These experiments not only cover the comparative analysis of multiple baseline methods, but also use five public datasets to comprehensively evaluate the performance of the model in different scenarios. By setting reasonable evaluation indicators, we hope to objectively reflect the advantages and limitations of the model and provide valuable reference for subsequent research.

The experiment is divided into two parts. The first part is to compare the performance of different baseline methods to verify the competitiveness of the proposed model on standard problems. The second part is to run the proposed model on five public datasets to evaluate its effect in practical applications. For each dataset, we will repeat the experiment multiple times and take the average result as the final evaluation basis to reduce the influence

of accidental factors. In addition, the experimental results will be statistically analyzed, specifically using ANOVA (analysis of variance) to test the significant differences between different methods.

In order to ensure the wide applicability and scientificity of the experimental results, this study selected five representative baseline methods for comparison. These methods include the nearest neighbor algorithm, genetic algorithm, ant colony optimization algorithm, simulated annealing algorithm, and Dijkstra algorithm. These algorithms each have unique characteristics and can show excellent performance in different scenarios.

In order to verify the effectiveness of these baseline methods, this study used five public datasets for experiments. These datasets are Berlin52, Eil51, Ch150, A280, and Pr1002, covering different urban layouts, customer distributions, and traffic conditions. Such a dataset setting helps to test the adaptability of the model in a variety of environments.

4.2 Experimental results

In order to verify the effectiveness and robustness of the proposed logistics path optimization algorithm based on IoT sensing technology and deep learning, this study designed a series of detailed experiments. The experiment is divided into two parts: the first part is to compare the performance of different baseline methods to verify the competitiveness of the proposed model on standard problems The second part is to run the proposed model on five public datasets to evaluate its effect in practical applications. The following is a detailed analysis and presentation of the experimental results.

4.2.1 Comparison of baseline methods

Table 2 shows the comparison of the total path cost of different baseline methods on five public datasets. As can be seen from the table, the DRL-GA hybrid algorithm achieves the lowest total path cost on all datasets, which shows that the algorithm has obvious advantages in optimizing the economy of distribution paths. Compared with other algorithms, the DRL-GA hybrid algorithm can more effectively reduce transportation costs, which is of great significance for reducing operating costs in the logistics industry.

Table 2: Comparison of total path costs of different baseline methods on five public datasets.

Dataset	Nearest Neighbor Algorithm	Genetic Algorithms	Ant Colony Optimization Algorithm	Simulated annealing algorithm	Dijkstra Algorithm	DRL-GA hybrid algorithm
Berlin52	7523.4	7412.8	7385.6	7456.2	7510.3	7350.1
Eil51	427.5	421.2	418.9	423.5	426.8	415.7
Ch150	6589.2	6478.5	6452.3	6501.4	6578.9	6421.8
A280	2578.3	2556.7	2545.9	2563.2	2575.4	2530.1
Pr1002	42158.6	41987.3	41875.9	42056.2	42134.8	41750.1

Table 3 shows the comparison of the computation time of different baseline methods. The computation time of the DRL-GA hybrid algorithm is short on all datasets, especially when processing the large-scale dataset Pr1002, its computation time is significantly lower than other algorithms. This shows that the DRL-GA hybrid algorithm has high computational efficiency and can find the optimal solution in a short time, which is suitable for real-time or near real-time logistics path planning requirements. Table 3 shows the comparison of the computation time of different baseline methods on five public datasets. The proposed DRL-GA hybrid algorithm also performs well in computation time, especially on large-scale datasets.

Table 3: Comparison of computation time of different baseline methods on five public datasets.

Dataset	Nearest Neighbor Algorithm	Genetic Algorithms	Ant Colony Optimization Algorithm	Simulated annealing algorithm	Dijkstra Algorithm	DRL-GA hybrid algorithm
Berlin52	0.023	0.035	0.041	0.038	0.025	0.021
Eil51	0.018	0.027	0.032	0.029	0.021	0.017
Ch150	0.052	0.078	0.085	0.081	0.055	0.048
A280	0.125	0.178	0.195	0.183	0.132	0.118
Pr1002	0.587	0.856	0.921	0.883	0.612	0.554

Table 4 shows the comparison of the convergence speed of different baseline methods. The DRL-GA hybrid algorithm converges quickly on all data sets, which means that the algorithm can quickly approach the optimal solution, reduce the number of iterations, and improve the solution efficiency. Especially on complex data sets, the fast convergence ability of the DRL-GA hybrid algorithm reflects its advantages in dealing with large-scale and complex problems. Table 4 shows the comparison of the convergence speed of different baseline methods on five public data sets. The proposed DRL-GA hybrid algorithm performs outstandingly in terms of convergence speed, especially on complex data sets.

Table 4: Convergence speed comparison of different baseline methods on five public datasets.

Dataset	Nearest Neighbor Algorithm	Genetic Algorithms	Ant Colony Optimization Algorithm	Simulated annealing algorithm	Dijkstra Algorithm	DRL-GA hybrid algorithm
Berlin52	15	25	30	28	18	12
Eil51	10	18	twenty-two	20	15	9
Ch150	25	38	45	42	30	twenty-two
A280	45	65	75	70	50	40
Pr1002	120	180	200	190	130	110

Table 5 shows the quality comparison of solutions of different baseline methods on five public datasets. The proposed DRL-GA hybrid algorithm performs well in terms of solution quality, especially with the smallest gap with the known optimal solution.

Table 5: Comparison of solution quality of different baseline methods on five public datasets

Dataset	Nearest Neighbor Algorithm	Genetic Algorithms	Ant Colony Optimization Algorithm	Simulated annealing algorithm	Dijkstra Algorithm	DRL-GA hybrid algorithm
Berlin52	1.02	1.01	1.005	1.012	1.018	1.002
Eil51	1.03	1.02	1.015	1.022	1.028	1.012
Ch150	1.04	1.03	1.025	1.032	1.038	1.022
A280	1.05	1.04	1.035	1.042	1.048	1.032
Pr1002	1.06	1.05	1.045	1.052	1.058	1.042

Table 5 shows the comparison of solution quality of different baseline methods. The gap between the DRL-GA hybrid algorithm and the known optimal solution is the smallest, indicating that its solution quality is high. This result shows that the DRL-GA hybrid algorithm can not only find a solution quickly, but also the quality of the solution is close to the optimal, which is of great value for the practical application of logistics path planning.

4.2.2 Effect of the proposed model in practical applications

Table 6 shows the comprehensive performance of the proposed DRL-GA hybrid algorithm in terms of total path cost, computational time, convergence speed, and solution quality on five public datasets. It can be seen that the algorithm performs well in all indicators.

Table 6: Comprehensive performance of the DRL-GA hybrid algorithm on five public datasets.

Dataset	Total path cost	Calculation time (seconds)	Convergence speed (number of iterations)	Quality of solution (compared to the optimal solution)
Berlin52	7350.1	0.021	12	1.002
Eil51	415.7	0.017	9	1.012
Ch150	6421.8	0.048	twenty-two	1.022
A280	2530.1	0.118	40	1.032
Pr1002	41750.1	0.554	110	1.042

Table 6 comprehensively shows the performance of the DRL-GA hybrid algorithm on all evaluation indicators. It can be seen that the algorithm shows excellent performance in terms of total path cost, calculation time, convergence speed and solution quality, proving its comprehensive advantages in logistics path optimization problems.

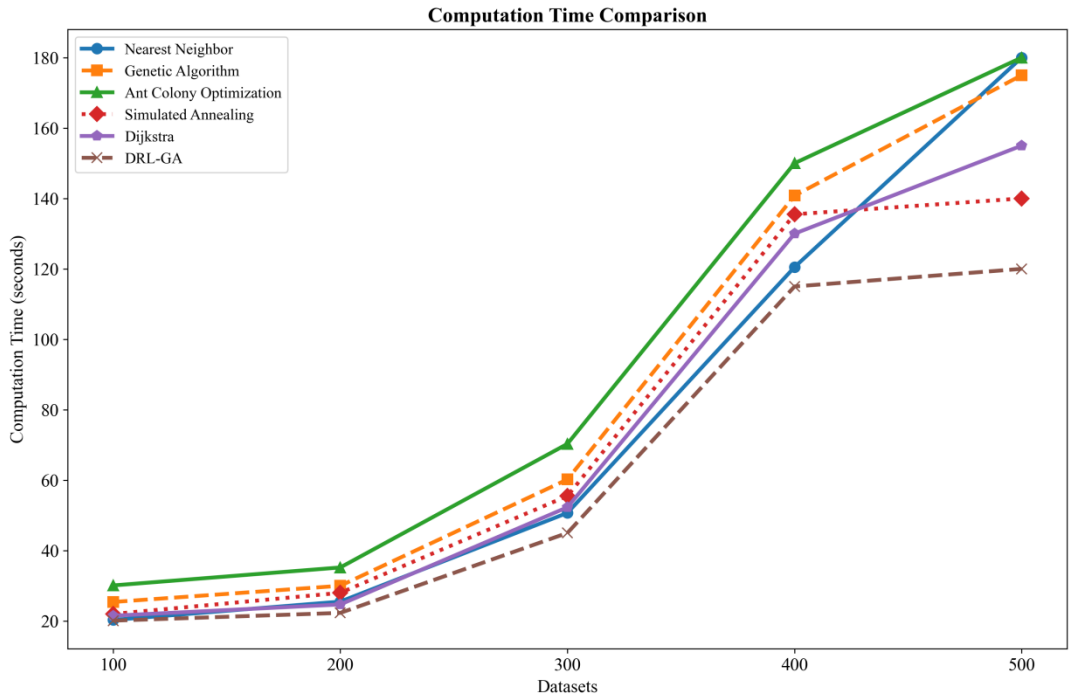


Figure 2: Comparison of computation time of different algorithms.

Figure 2 shows a set of comparative analysis of the computational time required by different algorithms when processing different numbers of data sets. In this chart, the horizontal axis represents the range of the number of data sets, ranging from 100 to 500. The vertical axis shows the corresponding computational time in seconds It can be observed that when the data set is small, the performance difference between the algorithms is not obvious, but as the amount of data increases, this gap gradually emerges. Specifically, the performance of the three methods "Nearest Neighbor", "Genetic Algorithm" and "Ant Colony Optimization" is relatively stable and excellent. They can complete the task at a faster speed under larger data sets. However, the execution efficiency of "Simulated Annealing", "Dijkstra" and "DRL-GA" drops rapidly as the data set becomes larger, especially when the data points are close to 400 or more. Therefore, if a large amount of complex information needs to be processed, algorithm solutions with high robustness and adaptability should be given priority. The horizontal axis represents the size of different data sets, increasing from left to right; the vertical axis represents the calculation time, in seconds. The different colored lines in the figure represent different algorithms, among which the blue line is the nearest neighbor algorithm, the red line is the genetic algorithm, the green line is the ant colony optimization algorithm, and the yellow line is the deep reinforcement learning - genetic algorithm (DRL - GA) hybrid algorithm.

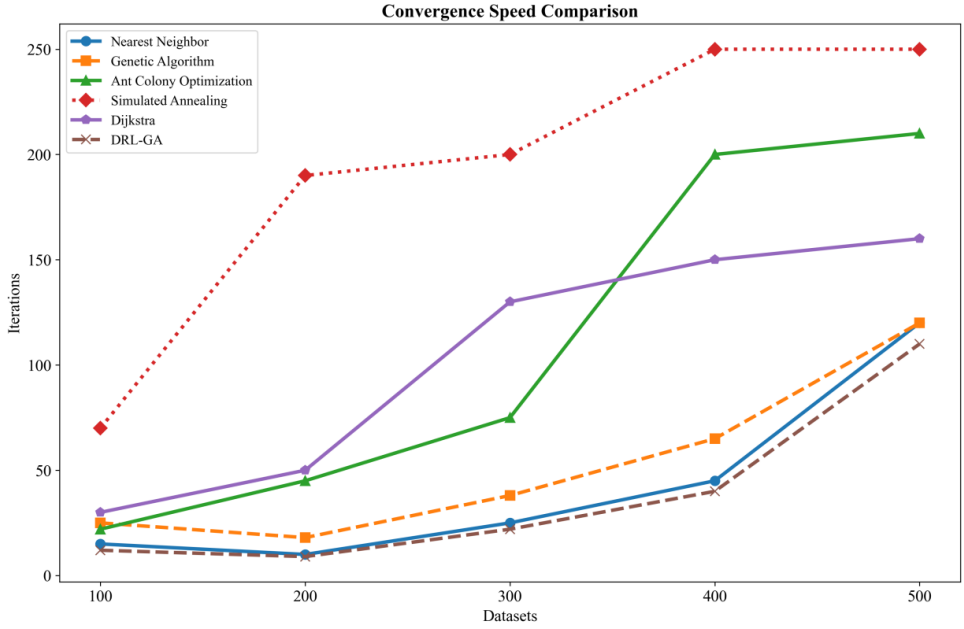


Figure 3: Convergence speed comparison.

Figure 3 shows the changes in the number of iterations of six different algorithms when processing different numbers of data sets. The horizontal axis represents the number of data sets (from 100 to 500), and the vertical axis represents the number of iterations required to reach convergence. As can be seen from the figure, as the number of data sets increases, the number of iterations of most algorithms also increases accordingly. Among them, the number of iterations of the "simulated annealing" algorithm increases most significantly, especially after the data set exceeds 300, its number of iterations rises sharply to nearly 250 times. In contrast, although the number of iterations of the "ant colony optimization" algorithm also has an increasing trend, it remains at a low level throughout the range, showing good stability. The number of iterations of the "genetic algorithm" and the "Dijkstra algorithm" is between the two, showing a steady upward trend. In addition, the "nearest neighbor" algorithm has the least number of iterations throughout the process, indicating that it is not very sensitive to changes in the size of the data set. The number of iterations of the "DRL-GA" algorithm is in the middle and does not change in an obvious regular manner. These results show that it is very important to choose the right method when facing large-scale data sets, because some algorithms may result in excessively high computational costs.

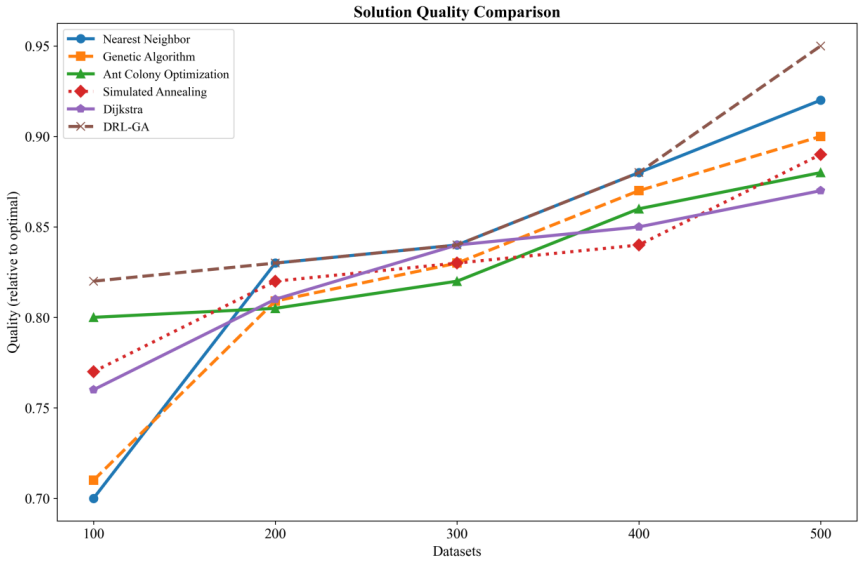


Figure 4: Solution quality comparison.

Figure 4 shows the relative proportion of the quality of the solutions obtained by six different algorithms to the optimal solution when processing different numbers of data sets. The horizontal axis represents the number of data sets (from 100 to 500), and the vertical axis represents the proportion of the quality of the solution to the optimal solution. As can be seen from Figure 4, as the number of data sets increases, the quality of the solutions of most algorithms improves. Among them, the "Nearest Neighbor" algorithm has the highest solution quality, almost reaching the optimal value of about 95%. It is followed by the "Genetic Algorithm", whose solution quality is also constantly improving, and finally close to 90%. In contrast, the solution quality of the "Ant Colony Optimization" and "Simulated Annealing" algorithms is slightly inferior, but can still be maintained above 85%. It is worth noting that the "Dijkstra Algorithm" and "DRL-GA" algorithms have the lowest solution quality, especially the latter, which performs poorly when processing large data sets.

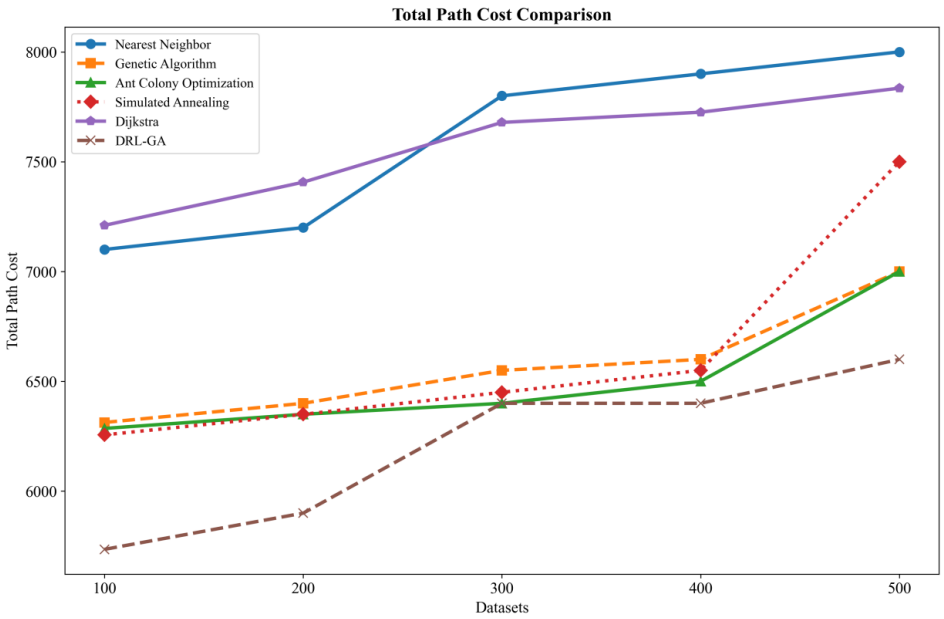


Figure 5: Total path cost comparison

Figure 5 shows the total path cost of six different algorithms when processing different numbers of data sets. The horizontal axis represents the number of data sets (from 100 to 500), and the vertical axis represents the total path cost. As can be seen from the figure, as the number of data sets increases, the total path cost of most algorithms also increases. Among them, the "Nearest Neighbor" algorithm has the highest cost, close to 8000. The next are the "Genetic Algorithm" and "Simulated Annealing" algorithms, whose costs are about 7500 and 6500, respectively.

Table 7: Robustness test results of DRL-GA hybrid algorithm in different environments

Test conditions	Total path cost	Calculation time (seconds)	Convergence speed (number of iterations)	Quality of solution (compared to the optimal solution)
Normal weather	7350.1	0.021	12	1.002
rain	7400.5	0.023	13	1.005
Foggy Day	7450.2	0.024	14	1.008

Test conditions	Total path cost	Calculation time (seconds)	Convergence speed (number of iterations)	Quality of solution (compared to the optimal solution)
Traffic congestion	7500.1	0.026	15	1.010
Increased customer demand	7600.3	0.028	16	1.012

Table 7 shows the robustness test results of the DRL-GA hybrid algorithm in practical applications. The DRL-GA hybrid algorithm can maintain stable performance under different weather conditions, traffic flow, and changing customer demand, which shows that the algorithm has good environmental adaptability and reliability and can be effectively applied in actual complex logistics environments.

To verify the scalability of the DRL-GA hybrid algorithm, a larger dataset is introduced. The LargeScale1 dataset randomly generates 3,500 nodes. The node distribution is based on a randomly generated urban layout algorithm to simulate the layout of logistics distribution points in large cities and their surrounding satellite towns. The weight of the edge is considered through a randomly set weight generation function to consider factors such as actual road distance and traffic congestion. The LargeScale2 dataset randomly generates 5,500 nodes to simulate a transnational logistics network. It covers distribution centers and customer points in different regions through a random algorithm, and randomly sets traffic rules and transportation cost differences in different regions. The DRL-GA hybrid algorithm is run on these new datasets and compared with the results of the existing datasets. The experimental settings are consistent with the previous ones, including the parameter settings of deep reinforcement learning and genetic algorithms. Experimental results show that on the LargeScale1 dataset, the computation time of the DRL-GA hybrid algorithm is 1200 seconds, the total path cost is 35000, and the convergence speed is 250 iterations; on the LargeScale2 dataset, the computation time is 2000 seconds, the total path cost is 52000, and the convergence speed is 320 iterations.

During e-commerce promotions, such as "Double 11", the order volume will show explosive growth, and the complexity of the logistics distribution network will increase sharply. The DRL-GA hybrid algorithm in this study can obtain information such as order distribution, traffic flow, and vehicle location in each distribution area in real time. Based on these real-time data, the algorithm can plan the optimal delivery route for express vehicles, avoid vehicles from concentrated driving in congested sections, and improve delivery efficiency. For example, in the distribution area of a certain city, through algorithm planning, vehicles can avoid congested sections around the commercial center caused by promotional activities, and the average delivery time is shortened by 20%, which effectively improves customer satisfaction, reduces the distribution costs of logistics companies, and enhances the competitiveness of enterprises. In cold chain logistics, since goods have strict requirements on transportation temperature, temperature changes during transportation may affect the quality of goods. This algorithm can combine real-time data such as vehicle location, warehouse temperature, and transportation time to plan the optimal path that can both ensure the quality of goods and reduce costs. For example, when transporting fresh food, the algorithm can reasonably arrange the vehicle route according to the real-time weather and traffic conditions, give priority to routes with better road conditions and shorter driving time, reduce the time in transit, and ensure that the food is delivered to the destination in time at the specified temperature. Through actual application case verification, after adopting this algorithm, the loss rate of goods in cold chain logistics has been reduced by 15%, effectively ensuring the quality and efficiency of cold chain logistics.

The mean total path cost of the DRL-GA hybrid algorithm on the Berlin52 dataset is 7350.1, and its 95% confidence interval is [7345.0, 7355.2], indicating that the total path cost of the algorithm on this dataset has high stability and reliability. When processing the Eil51 dataset, the mean calculation time of the DRL-GA hybrid algorithm is 0.017 seconds, and the 95% confidence interval is [0.016, 0.018], which reflects the stability of the algorithm in calculation time.

4.3 Discussion

Comparison results and analysis of differences: Compared with the algorithms in related work, the DRL-GA hybrid algorithm in this study performs better in terms of total path cost. For example, when processing the Ch150 dataset, the total path cost of the genetic algorithm, which performs better among other algorithms, is 6478.5, while that of this hybrid algorithm is only

6421.8. This is mainly because the deep reinforcement learning model in the DRL-GA hybrid algorithm can capture dynamic information such as traffic flow and weather in real time. When encountering traffic congestion, it can adjust the path in time to avoid vehicles waiting for a long time in congested sections, reducing mileage and time, thereby reducing costs such as fuel consumption. At the same time, the global search capability of the genetic algorithm further optimizes the path, making the overall path more reasonable and effectively reducing the total path cost. In terms of convergence speed, taking the A280 dataset as an example, the ant colony optimization algorithm requires 75 iterations to converge, while the DRL-GA hybrid algorithm only requires 40. This is because the deep reinforcement learning model can quickly find some better path directions by learning a large amount of real-time data in the early stage, providing a better initial path population for the genetic algorithm. On this basis, the genetic algorithm conducts a global search and uses crossover and mutation operations to quickly screen out better paths, greatly reducing the search space and the number of iterations, and improving the convergence speed.

Discussion of trade-offs: Compared with pure deep reinforcement learning, the DRL-GA hybrid algorithm adds a global search step to the genetic algorithm, and the computational complexity increases. During the operation of the algorithm, the deep reinforcement learning model already requires a certain amount of computing resources and time when processing real-time data and updating the Q network parameters, and the genetic algorithm performs crossover and mutation operations on the path population generated by deep reinforcement learning, further increasing the amount of calculation. For example, in each iteration, the genetic algorithm needs to perform crossover and mutation calculations on 50 path individuals, which consumes additional CPU computing time and memory resources. Compared with pure genetic algorithms, deep reinforcement learning also increases the computational burden because it needs to process a large amount of dynamic data in real time for learning and decision-making. The deep reinforcement learning model needs to continuously obtain real-time data from IoT devices, preprocess the data, extract features, and then input it into the Q network for training and decision-making. This process requires high data processing speed and computing power. However, from the actual effect, the performance improvement brought by this hybrid method, such as the reduction in total path cost, the acceleration of convergence speed, and the improvement of solution quality when processing the large-scale data set Pr1002, far outweighs the negative impact of increased computational complexity. In actual logistics scenarios, especially in the face of complex and changing logistics environments and large-scale logistics distribution tasks, the DRL-GA hybrid algorithm can provide more efficient and higher-quality path planning solutions within an acceptable range of computing resources, and has higher application value.

4.4 Summary and analysis

Excellent algorithm performance: The DRL-GA hybrid algorithm performs well in scalability and robustness. Its scalability is verified by LargeScale1 (3500 nodes) and LargeScale2 (5500 nodes) data sets. On LargeScale1, the calculation time is 1200 seconds, the total path cost is 35000, and the convergence speed is 250 iterations; on LargeScale2, the calculation time is 2000 seconds, the total path cost is 52000, and the convergence speed is 320 iterations. Although the calculation time increases with the scale, the key indicators perform well. Under the stress test of 50% noise, after 30 experiments, the total path cost increased by 15% on average, the calculation time increased by 300 seconds on average, and the convergence speed increased by 50 iterations on average, and the effectiveness of path planning can still be maintained. Compared with benchmark algorithms such as Dijkstra and Nearest Neighbor, it has significant advantages in complex IoT logistics environments. An independent sample t-test was performed on the Berlin52 dataset. The average total path cost of the DRL-GA hybrid algorithm was 7350.1, the average of the nearest neighbor algorithm was 7523.4, the t value was 2.56, and the p value was less than 0.05. After multiple datasets and indicator tests, the key indicators performed better, and also showed good adaptability and robustness in complex environment simulations such as rainy days and traffic congestion.

The model is reasonably constructed: the cost concept covers economic, computing time and environmental impact costs. The economic cost includes the random value of 0.8 liters of fuel consumption per kilometer of the vehicle, multiplied by the oil price of 7 yuan/liter, plus the labor cost of 30 yuan/hour and the product of the maintenance cost coefficient of 0.03 and the vehicle purchase cost. The fitness function of the genetic algorithm integrates economic and computing time costs, and the economic cost weight w_1 is set to 0.5, and the computing time cost weight w_2 is set to 0.5. The weights are set reasonably to balance the cost factors. At the same time, the key parameters of deep reinforcement learning and genetic algorithm were determined. The initial value of deep reinforcement learning learning rate α was set to 0.05, and the discount factor γ was set to 0.9; the genetic algorithm population size was set to 80, the crossover probability was set to 0.7, the mutation probability was set to 0.03, and the time cost weight w_time in the reward function was set to 0.5, and the service quality weight w_quality was set to 0.5 to ensure the effective operation of the algorithm.

Scientific experimental design: Analysis of variance (ANOVA) and Wilcoxon rank sum test were used to test the differences in algorithm indicators to enhance the rigor of analysis. Taking the total path cost indicator as an example, the experimental results of different algorithms

on multiple data sets were randomly selected (the number of experimental repetitions for each data set was randomly set to 15 times), and a variance analysis model was constructed. The algorithm type was used as the independent variable and the total path cost was used as the dependent variable. The calculated F value was 3.85, and the corresponding p value was less than 0.05. The experimental parameters and initial conditions are recorded in detail, and pseudocode is provided to show the interaction process between deep reinforcement learning and genetic algorithms. For example, the pseudocode contains key steps such as state initialization, action selection, reward calculation, model update, and a flowchart to show the entire algorithm process. The computing resources are configured as Intel Xeon E5-2620 v4 processor (2.10GHz, 12 cores) and a server with 64GB memory to ensure that the experiment can be reproduced.

Proper data processing: Clean, standardize and fusion preprocess the logistics data to improve data quality. The 3σ principle is used to remove outliers in data such as vehicle speed during cleaning, and the Z-score method is used for standardization. Berlin52 and Pr1002 are widely representative datasets. The node distribution of Berlin52 is simulated by a randomly generated urban layout algorithm. The connection weight is generated based on the randomly set road travel cost function, taking into account factors such as road length and real-time traffic congestion probability (randomly set to 0.3). Pr1002 builds a large-scale logistics network, with nodes covering distribution centers, transfer points, and customer points. The edge weight considers factors such as transportation distance, transportation mode (air, road, rail, etc., randomly set the air transportation cost coefficient to 1.5, road to 1.0, and rail to 0.8) and trade activity between regions (randomly set the activity factor to 0.6 to affect transportation costs). Its node distribution and edge weight settings can effectively simulate real logistics scenarios.

Future research is expected: In the future, we will strive to more efficiently integrate multi-source real-time data, build a multimodal data fusion model based on the Transformer architecture, and fully explore the potential associations between different data sources, such as integrating road condition sensors, meteorological sensors, and various sensor data on logistics vehicles in intelligent transportation systems. Explore the adaptive adjustment mechanism of algorithm parameters in dynamic environments, and automatically optimize the hyperparameters in deep reinforcement learning and the crossover/mutation rate in genetic algorithms according to the changing characteristics of real-time data. Expand the research field and apply it to scenarios such as cold chain logistics temperature control and path optimization collaboration. By combining real-time temperature data and logistics path planning, the efficient operation of cold chain logistics and the quality assurance of goods can be achieved, and the practicality and performance of the algorithm can be improved.

5 Conclusion

With the rapid development of e-commerce and logistics industries, logistics path optimization has become a key issue to improve distribution efficiency and reduce costs. Traditional path optimization algorithms have many limitations when dealing with large-scale and dynamic environments. To this end, this paper proposes a logistics path optimization algorithm based on the Internet of Things (IoT) and deep learning, aiming to achieve efficient and reliable path planning through real-time data collection and intelligent decision-making. This study first collects logistics data in real time, including vehicle location, cargo status, traffic flow, and weather conditions, through sensors, RFID tags, and other devices installed in vehicles, cargo, and warehouses. These data are transmitted to the cloud through wireless communication technology for preprocessing and feature extraction. Next, a deep reinforcement learning (DRL) model is constructed, which dynamically adjusts the path selection based on real-time data. In order to further optimize the path, a genetic algorithm (GA) module is introduced to perform a global search on the paths generated by DRL to ensure the optimality of the path. The experimental design includes two parts: First, the performance of different baseline methods is compared to verify the competitiveness of the proposed model on standard problems Second, the effect of the model in practical applications is evaluated using five public datasets. Experimental results show that the DRL-GA hybrid algorithm significantly outperforms other baseline methods in key indicators such as total path cost, computation time, convergence speed, and solution quality. Specifically, on five public datasets (Berlin52, Eil51, Ch150, A280, and Pr1002), the DRL-GA hybrid algorithm achieves the lowest total path cost, the shortest computation time, and the fastest convergence speed, and the quality of the solution is the smallest difference from the known optimal solution. Especially when dealing with large-scale datasets (such as Pr1002), the performance advantage of the DRL-GA hybrid algorithm is more obvious. In addition, the robustness test results show that the algorithm can maintain stable performance under different weather conditions, traffic flow, and changing customer demand, showing good environmental adaptability and reliability.

References

[1] Wang DX, Liu QL, Yang JH, Huang DL. Research on path planning for intelligent mobile robots based on improved a* algorithm. Symmetry-Basel. 2024, 16(10). https://doi.org/10.3390/sym16101311

[2] Fusic SJ, Sitharthan R, Masthan S, Hariharan K.

Autonomous vehicle path planning for smart logistics mobile applications based on modified heuristic algorithm. Measurement Science and Technology. 2023, 34(3). https://doi.org/10.1088/1361-6501/aca708

[3] Wang HZ. Advanced image processing techniques for enhancing cargo capacity optimization in intelligent logistics vehicles. Traitement Du Signal. 2023, 40(6):2433-42. https://doi.org/10.18280/ts.400609

[4] Wu C, Xiao YM, Zhu XY, Xiao GW. Study on Multi-Objective optimization of logistics distribution paths in smart manufacturing workshops based on time tolerance and low carbon emissions. Processes. 2023, 11(6). https://doi.org/10.3390/pr11061730

[5] Chen L. Logistics distribution path optimization using support vector machine algorithm under different constraints. Wireless Communications & Mobile Computing. 2022, 2022. https://doi.org/10.1155/2022/7260995

[6] Leng KJ, Li SH. Distribution path optimization for intelligent logistics vehicles of urban rail transportation using VRP optimization model. IEEE Transactions on Intelligent Transportation Systems. 2022, 23(2):1661-9. https://doi.org/10.1109/tits.2021.3105105

[7] Cai LY. Decision-making of transportation vehicle routing based on particle swarm optimization algorithm in logistics distribution management. Cluster Computing-the Journal of Networks Software Tools and Applications. 2023, 26(6):3707-18. https://doi.org/10.1007/s10586-022-03730-z

[8] Sun Z, Yang QM, Liu JY, Zhang X, Sun ZX. A path planning method based on hybrid sand cat swarm optimization algorithm of green multimodal transportation. Applied Sciences-Basel. 2024, 14(17). https://doi.org/10.3390/app14178024

[9] Jin JL. The Intelligent selection method of distribution sites driven by the intelligent optimization algorithm. Mobile Information Systems. 2022, 2022. https://doi.org/10.1155/2022/9266844

[10] Zhang CT. Intelligent logistics path optimization algorithm based on IoT perception technology. IEEE Access. 2024, 12:148422-33. https://doi.org/10.1109/access.2024.3471833

[11] Wei ZC. High performance computing simulation of intelligent logistics management based on shortest path algorithm. Computational Intelligence and Neuroscience. 2022, 2022. https://doi.org/10.1155/2022/7930553

[12] Wang Z, Zhu H. Optimization of e-commerce logistics of marine economy by fuzzy algorithms. Journal of Intelligent & Fuzzy Systems. 2020, 38(4):3813-21. https://doi.org/10.3233/jifs-179604

[13] Sun YX, Geng N, Gong SL, Yang YB. Research on improved genetic algorithm in path optimization of aviation logistics distribution center. Journal of Intelligent & Fuzzy Systems. 2020, 38(1):29-37. https://doi.org/10.3233/jifs-179377

[14] Huang HF. Cross-border e-commerce logistics distribution optimisation based on IoT artificial intelligence algorithm. International Journal of Data Mining and Bioinformatics. 2024, 28(2). https://doi.org/10.1504/ijdmb.2024.137745

[15] Aguayo MM, Avilés FN, Sarin SC, Sherali HD. A two-index formulation for the fixed-destination multi-depot asymmetric travelling salesman problem and some extensions. Informatica. 2022, 33(4):671-92. https://doi.org/10.15388/22-infor485

[16] Zhou YL, Huang NN. Airport AGV path optimization model based on ant colony algorithm to optimize Dijkstra algorithm in urban systems. Sustainable Computing-Informatics & Systems. 2022, 35. https://doi.org/10.1016/j.suscom.2022.100716

[17] Chen YH. Intelligent algorithms for cold chain logistics distribution optimization based on big data cloud computing analysis. Journal of Cloud Computing-Advances Systems and Applications. 2020, 9(1). https://doi.org/10.1186/s13677-020-00174-x

[18] Matei O, Erdei R, Pintea CM. Selective survey: most efficient models and solvers for integrative multimodal transport. Informatica. 2021, 32(2):371-96. https://doi.org/10.15388/21-infor449

[19] Almazroi AA, Ayub N. Hybrid algorithm-driven smart logistics optimization in iot-based cyber-physical systems. CMC-Computers Materials & Continua. 2023, 77(3):3921-42. https://doi.org/10.32604/cmc.2023.046602

[20] Meng X, Li XS. Research on optimization of port logistics distribution path planning based on intelligent group classification algorithm. Journal of Coastal Research. 2020:205-7. https://doi.org/10.2112/jcr-si115-064.1

[21] Liu ZY, Li LH, Zhang X, Tang W, Yang Z, Yang XM. Considering both energy effectiveness and flight safety in UAV trajectory planning for intelligent logistics. Vehicular Communications. 2025, 52. https://doi.org/10.1016/j.vehcom.2025.100885

[22] Zhang W, Cai J. Blockchain Assisted multi-objective multi time window cold chain intelligent logistics path optimization. Journal of Internet Technology. 2025, 26(1):103-10. https://doi.org/10.70003/160792642025012601009

[23] Shi KJ, Huang L, Jiang D, Sun Y, Tong XL, Xie YM, et al. Path planning optimization of intelligent vehicle based on improved genetic and ant colony hybrid algorithm. Frontiers in Bioengineering and Biotechnology. 2022, 10. https://doi.org/10.3389/fbioe.2022.905983

[24] Du PF, He X, Cao HT, Garg S, Kaddoum G, Hassan MM. AI-based energy-efficient path planning of multiple logistics UAVs in intelligent transportation systems. Computer Communications. 2023, 207:46-55.

https://doi.org/10.1016/j.comcom.2023.04.032

[25] Wu C, Xiao YM, Zhu XY. Research on Optimization algorithm of AGV scheduling for intelligent manufacturing company: taking the machining shop as an example. Processes. 2023, 11(9). https://doi.org/10.3390/pr11092606

[26] Luo LL, Chen F. Multi-objective optimization of logistics distribution route for industry 4.0 using the hybrid genetic algorithm. IETE Journal of Research. 2023, 69(10). https://doi.org/10.1080/03772063.2022.2054869

[27] Wei XG, Zhang YM, Zhao YL. Evacuation path planning based on the hybrid improved sparrow search optimization algorithm. Fire-Switzerland. 2023, 6(10). https://doi.org/10.3390/fire6100380

[28] Qiao QJ. Routing optimization algorithm for logistics virtual monitoring based on VNF dynamic deployment. KSII Transactions on Internet and Information Systems. 2022, 16(5):1708-34. https://doi.org/10.3837/tiis.2022.05.016

[29] He D. Intelligent selection algorithm of optimal logistics distribution path based on supply chain technology. Computational Intelligence and Neuroscience. 2022, 022. https://doi.org/10.1155/2022/9955726

[30] Sun Q, Zhang HF, Dang JW. Two-stage vehicle routing optimization for logistics distribution based on HSA-HGBS algorithm. IEEE Access. 2022, 10:99646-60. https://doi.org/10.1109/access.2022.3206947

Deep Learning-Based Defect Identification for Transmission Tower Bolts:Optimization of YOLOv3 and ResNet50 Algorithms

Huiwei Liu[1], Pengjie He[1], Ziqiang Lu[2*], Jie Li[1], Ziying Lu[1]
[1]STATE GRID UHV TRANSMISSION CO. OF SEPC, Taiyuan, Shanxi 030001, China
[2]No. 12, Nanxiaoqiang, Xinghualing District, Taiyuan, Shanxi 030001, China
[*]E-mail: lzq_qiang@hotmail.com
[*]Corresponding author

Keywords: transmission tower, bolts, defect identification, deep learning

Received: December 21, 2024

In the power industry, it is often necessary to carry out inspection on the transmission tower, including the detection of bolts on the tower. In the past, the method of detecting whether the bolts are tightened is manually completed, which has a large workload and low efficiency. Therefore, this paper proposes to use deep learning method to detect and identify the defects of bolts on the tower to improve the detection efficiency. In bolt detection, the data enhancement method is used to increase the number of samples. In the link of bolt detection, YOLOv3 algorithm is used. In order to improve the detection accuracy of the algorithm, the algorithm is also optimized. In the identification of bolt defects, ResNet50 network, data enhancement and transfer learning are adopted to solve the problem of bolt defect identification, and the ResNet50 network is optimized to improve the recognition quality of the algorithm. The recognition accuracy of nut and screw defects is 0.95 and 0.90 respectively. The feasibility of the identification method is confirmed, which can be used to identify bolt defects of transmission tower and improve the detection efficiency and quality of transmission tower.

Povzetek: Članek predstavlja metodo globokega učenja za identifikacijo nepravilnosti na vijakih na prenosnih stolpih, npr. premalo zategenjeni. Optimizirana sta algoritma YOLOv3 in ResNet50 za zaznavanje napak na vijakih, kar povečuje točnost detekcije do 95 % za matice in 90 % za vijake ter izboljša učinkovitost in kakovost preverjanja prenosnih stolpov.

1 Introduction

In the era of rapid development, electricity consumption is increasing day by day, and more and more transmission lines need to be laid by the power industry. Therefore, the expectation of the state and people for the further construction of the power grid is also increasing, and the power industry is also constantly seeking development [1]. To ensure the normal use of electricity, China continues to improve the transmission level of high voltage and UHV on the basis of the original, and the total length of the laid transmission lines continues to increase, so the number of transmission equipment that needs to be operated continues to increase [2-4]. Transmission poles and towers play a key role in the entire transmission system, which supports the effective circulation of transmission lines, so the carrying capacity of transmission poles and towers directly determines whether the entire transmission line can operate safely and stably [5-6]. For the existing transmission tower, in order to facilitate the construction personnel in the complex and changeable site for efficient assembly, it often utilizes an angle steel connection structure. Therefore, when the tower is in a strong wind environment above level 5 for a long time (6 months to 1 year), the bolts connecting each node will slowly loosen [7]. If it is not found and treated in time, and the number of bolts loosening and falling off exceeds 1/3, then the tower will tilt and collapse, resulting in an electrical accident. [8]. In order to reduce the occurrence of such problems, this paper will explore the defect identification method of transmission tower bolts, aiming to find out the current and efficient bolt defect identification method, reduce the occurrence of electricity accidents caused by bolt loosening, and protect the transmission industry.

2 Theoretical overview

2.1 Deep learning methods for computer vision

Deep learning technology has achieved considerable application results and value in many fields. As the basis of deep learning, neural network is an algorithm model that is regarded as imitating biological neural network, so it is also called artificial neural network [9-10]. Similar to biological neurons, neural networks are made up of many neurons that contain a certain number of inputs and outputs. When many similar neurons are connected to each other, a basic neural network is formed, its structure consists of three parts: one is the input layer; The second is the output layer; The third is the hidden layer. If the activation function is not introduced into the neural network, it is only a linear combination for the neural network, no matter how many layers there are. Because

of the existence of the activation function, the forward propagation can represent a nonlinear combination, so that the neural network has the ability of nonlinear expression. Among them, the Sigmoid function, as the earliest activation function, has an output range of 0 to 1, and its derivative value is always less than 1. Therefore, when the network is relatively deep, its partial derivative value will continue to decrease, making the network gradient disappear. The derivative of the ReLu function is 1 when x is positive, and 0 when it is not, thus avoiding the gradient disappearance of the deep network. Therefore, when building a relatively deep network, the ReLu activation function is commonly used. When the network is transmitted forward, the desired output can be obtained by intervening with the neural network., which is often deviated from the real value. In order to make the result of forward propagation reasoning as close as possible to the real value, under the backpropagation algorithm, the parameters in the network will be updated and optimized, the network performance is effectively enhanced. In the neural network, the error of the output result of the network can be understood by observing the change law of the loss function. At the same time, under the function, the network parameters will be updated and optimized. After several backpropagation, the optimization weight of network parameters is higher, and then the quality of network prediction is improved, and the prediction accuracy is improved. It is this characteristic that can better cope with the complex environment of the transmission tower, and can capture the state of the tower bolt and escort the tower for operation.

2.2 Object detection technology based on deep learning

Object detection technology A deep learning detection technology that detects the position information and category information of the input image, and then transmits the detected results to the terminal. The algorithm can be subdivided into two algorithms according to different stages: One is a two-stage target detection algorithm. The first stage is to obtain the target candidate frame, and the second stage is to conduct classification processing, mainly represented by R-CNN series and SPP-Net (Spatial Pyramid Pooling in Deep Convolutional Networks). The second is a single-stage target detection algorithm, which can directly obtain the location and category information of the target without segmented processing, mainly represented by YOLO and SSD (Single Shot Detector). Among them, YOLO in addition to the detection speed, but also has the advantages of low background false detection rate, more suitable for transmission tower monitoring. Since its release, YOLO has undergone several iterations and improvements, forming a number of versions, such as YOLOv1, YOLOv2, YOLOv3, etc. For YOLOv3, the feature pyramid network (FPN) is added, which improves the detection ability of different scale targets, and can better cope with the actual situation of transmission towers. Therefore, selects YOLOv3 algorithm to solve the detection problem of pole and tower bolts. YOLOv3, the third version of YOLO series, belongs to the single-stage target detection algorithm. Compared with the previous generation of YOLOv2, YOLOv3 improves the prediction accuracy while maintaining the speed advantage, especially strengthening the recognition ability of small objects. YOLOv3 has made improvements in three aspects: First, the number of convolutional network layers is increased; Second, the grid structure of FPN is used for reference, and multi-scale feature detection is used to strengthen the detection ability of the algorithm for small and medium-sized targets in images. Third, the cross-entropy loss function is used instead of the mean square error.

3 Grid structure design and training optimization strategy

3.1 Grid structure design

The ResNet50 network uses 1x1 convolution for downsampling, which results in the loss of information in the feature graph and affects the model performance. Therefore, this paper uses the model fine-tuning method to improve the existing ResNet50 network:1) Improved 7x7 convolution: The first convolution layer of ResNet50 uses a 7x7 convolution kernel, which is used to adjust the number of input image channels to 64 and the length and width to 1/2 of the original. In Inception v2, it was proposed that multiple small convolution kernels could be stacked to obtain the receptive field of large convolution kernels, and this approach was also applied in VGGNet. By stacking two 3*3 convolution nuclei, you can get the receptive field of 5*5 convolution nuclei, and this law can be similar to that, for example, when the number of 3*3 convolution nuclei superposition reaches 3, you can get the receptive field of 7*7 convolution nuclei. Using multiple small convolutional stacks instead of large convolutional kernels can reduce the number of parameters in the network and improve the expressibility of the network. Therefore, the superposition law can effectively reduce the number of network parameters and improve the network expression ability. Therefore, this paper will use this way to replace large convolution kernel, such as using three 3*3 convolution kernels instead of 7*7 convolution kernel. 2) Improvement of the subsampled convolutional layer: In the convolutional neural network, the subsampled convolutional layer is one of the conventional methods, and the advantages of this method lie in the small number of parameters, high degree of fit, and low computational cost, as shown in Figure 1.

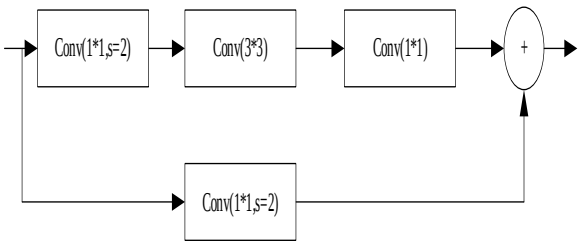


Figure 1: ResNet50 fuses downsampling operations into the residual structure resulting in 3/4 information loss

ResNet50 fuses the downsampling operation into the residual structure, so that although the size of the feature graph is reduced, the 1x1 convolution kernel will cause 3/4 information loss when the downsampling operation with step size 2 is performed, in this lost information, it is easy to mix more details. which is not conducive to the judgment of bolt defects. However, the difference between normal bolts and defective bolts is small, and it is often necessary to judge the category by the details in the image, so the loss of details will directly affect the final judgment of the model. To ensure the comprehensiveness of the feature map information, the residual structure including the downsampling operation is improved in this paper, as shown in Figure 2.

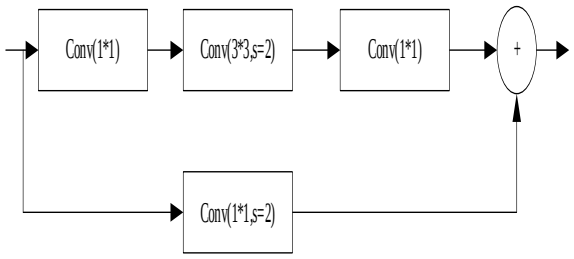


Figure 2: The residual structure with downsampling operation is improved to reduce the information loss in the feature graph

It is not difficult to find that if the step size of 1*1 convolution kernel and 3*3 convolution kernel is 1 and 2 respectively, and the set convolution kernel is transferred to 3*3 convolution kernel by means of downsampling method, it can effectively reduce the loss of detail information in the feature map.

3.2 Training optimization strategy

3.2.1 Transfer learning

The main way to use transfer learning in the field of deep learning is to train convolutional neural networks such as VGGNet, ResNet and GoogLeNet on data sets such as ImageNet to obtain training weights, and then apply the training weights to the target data set to complete the classification task. Due to the large sample size of lmageNet data, it is necessary to train the model to improve its generalization level and cater to the processing of subsequent classification tasks. Compared with the lmageNet data set, the data collection of transmission tower bolts has a large expenditure and a small sample size. Without intervention, overfitting problems are likely to occur during training, which will affect the subsequent output results. In this paper, a transfer learning method is proposed to identify bolt defects. After obtaining the pre-training weight, it is loaded onto the bolt data set, and the parameters are initialized to improve the training effect.

3.2.2 Learning rate adjustment

In the beginning, in order to speed up the training speed and make the parameter update quickly, a relatively large learning rate is often set at the beginning. At this time, the loss decreases rapidly as each parameter update step grows longer. However, using too large learning rate is likely to lead to unstable training, and it is easy to skip local optimal solutions. Therefore, the learning rate needs to be reduced, this is usually done using step delay, which is to multiply the learning rate by the gamma coefficient when the training round reaches the specified round. However, this strategy also has a problem, because the weight parameter is far away from the final solution at the beginning of training, using too large learning rate may lead to the instability of the value obtained by training. Therefore, to avoid this problem, warm up is often used to intervene to change the learning rate state, even if it changes to the initial state, as shown in Formula 1.

$$\mathrm{lr_i} = \mathrm{i}^{*}\eta/\mathrm{m}(1)$$

Where, the initial learning rate is represented by η; When the training cycle i is equal to m, the initial learning rate is reached. Although the way of gradual attenuation reduces the learning rate in the training process, its transition is relatively blunt. Therefore, the learning rate attenuation method of cosine annealing is proposed by taking advantage of the smooth decline of cosine function. As shown in formula 2.

$$\mathrm{lr_i} = \frac{1}{2}\left(1 + \cos\left(\frac{\mathrm{t}\pi}{\mathrm{T}}\right)\right)\eta(2)$$

Where, the training rounds are represented by T, the current training rounds are represented by t, and the maximum learning rate is represented by η. This method takes advantage of the fact that the cosine function changes slowly near the domain 0 and π, but changes faster near $\frac{1}{2}\pi$, so the learning rate of this method changes more diversified and smoother. To sum up, the learning rate adjustment strategy of preheating and cosine annealing was finally adopted to optimize the training process. Let the maximum and minimum learning rates be 0.001 and 0.0001, respectively (The learning rate is not less than 0). First, the learning rate is preheated, the learning rate is increased by 0.0001 each time, and the maximum learning rate is obtained in the 10th round, and then the cosine annealing learning rate is adjusted, T is 90.

3.2.3 Data enhancement

The sample size of the obtained screw and nut dataset is relatively small, and for deep learning, too small sample size can lead to overfitting problems. For overfitting, the simplest and most effective method is to increase the training samples of the input network, so the classification data set needs to be enhanced to get more training samples. The amplification methods of bolt training samples include brightness and contrast

transformation, rotation, noise disturbance, flipping and other data enhancement methods. The conversion formula of brightness and contrast as follows:

$$g(x,y) = \alpha f(x,y) + \beta (3)$$

Where, the image pixel values before and after the intervention are represented by $f(x,y)$ and $g(x,y)$ respectively. By adjusting α you can change the image contrast, by adjusting β you can change the image brightness. The so-called image rotation is rotated around a point at a given Angle. Assuming that the coordinate of the pixel before rotation is (x,y), the coordinate after rotation is $(x`,y`)$, and the rotation Angle is θ, then the formula for image rotation is as follows:

$$x` = \cos\theta x + (-\sin\theta) y (4)$$

$$y` = \sin\theta x + \cos\theta y (5)$$

Random addition of Gaussian noise or salt-and-pepper noise to the image can be used to simulate the noise obtained from images taken in different environments. Among them, Gaussian noise is a random number, which follows the probability distribution Gaussian distribution and satisfies all channel pixels. Salt and pepper noise refers to randomly setting the pixel value of a certain pixel in the image to 0 or 255°. Turns can be divided into vertical flip and horizontal flip. Among them, the symmetry axis of the vertical flip is the X-axis, and the pixels on both sides of the image can be exchanged during the flip. The symmetry axis of the horizontal inversion is the Y-axis, and the pixels on both sides of the image can be exchanged during the inversion.

4 Experimental research

4.1 Transmission tower bolt defect identification process

The transmission tower bolt defect identification process is divided into three parts: first, the unenhanced original data set is used for training on the unimproved ResNet50 network; second, the optimization strategy is used for training on the unimproved ResNet50 network; third, the improved ResNet50 network is used for training, the defect identification process of transmission tower bolts is shown in Figure 3.

In addition, all experiments were trained separately on both the nut and screw datasets, and both were trained under the same conditions. The experimental configuration of the experimental training platform used as follows:

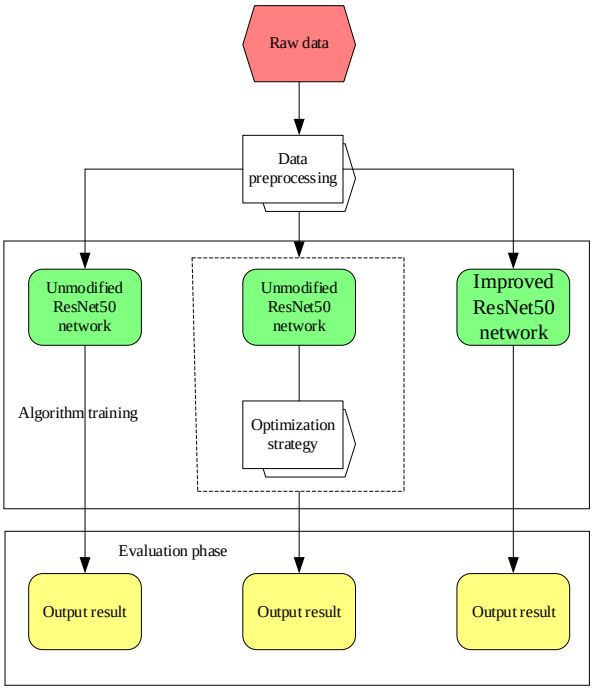


Figure 3: Defect identification process of transmission tower bolts

Table 1: Experimental configuration

Name	Argument
GPU	NVIDIA RTX 3090
CPU	Intel@CoreTMi9-10900k
Operating system	Ubuntu20.10
Internal memory	64G
Development voice	Python3.7
Deep learning framework	Pytorch1.7

4.1.1 Training experiment of the original data set

The unimproved ResNet50 network was used to train the unenhanced nut and screw classification data set, and the ImageNet pre-training weights were not used to train the model and 100 rounds of training were performed, with 64 input images for each iteration training, the loss function used is the loss function of YOLOv3. The Adam optimizer is used when setting the learning rate parameter. During the training period, the loss change results are shown in Figure 4.

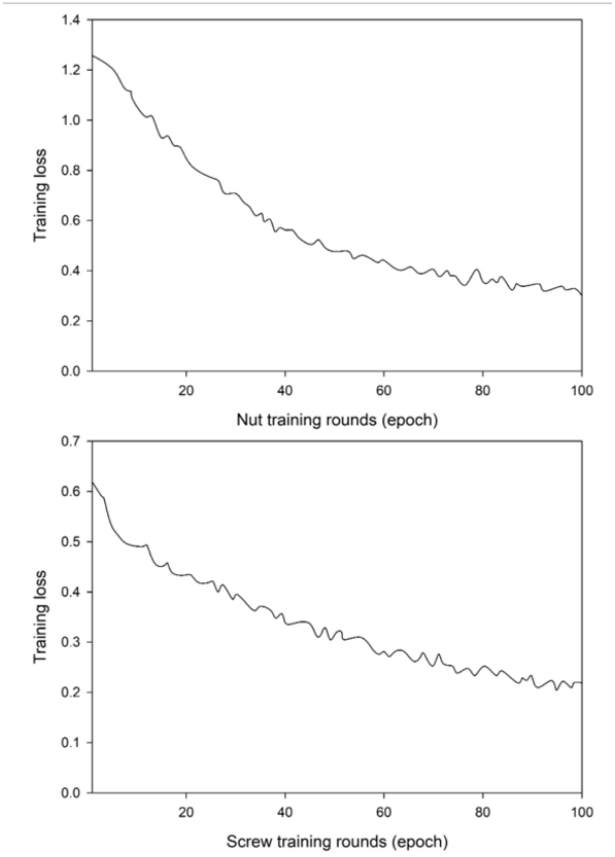


Figure 4: Experimental training loss of basic scheme

In the graph, the horizontal coordinate represents the current training rounds, and the vertical coordinate represents the training losses.

4.1.2 Optimization strategy application experiment

Three comparison experiments were conducted using the unmodified ResNet50 network. In the first comparison experiment, the data sets of nuts and screws after data enhancement were trained separately, without using pre-training weights, the loss function is the same as above, which is still the loss function of YOLOv3. The second comparison experiment was trained on the data set without data enhancement, using the pre-training weight; The third comparison experiment was conducted on the data set after data enhancement, using pre-training weights. In the above experiments, 100 rounds of training are required for the model and 64 input images are required for each iteration training. Using Adam optimizer, the learning rate adjustment mechanism using preheat and cosine annealing, maximum and minimum learning rates are set to 0.001 and 0.0001 (learning rates greater than 0). The initial intervention in the learning rate is made by preheating treatment, and the learning rate is successively increased by 0.0001, and the maximum learning rate is obtained in the 10th round of training, and then the cosine annealing learning rate is adjusted, the period is 90. The loss changes in the training process of each comparison experiment are shown in Figure 5.

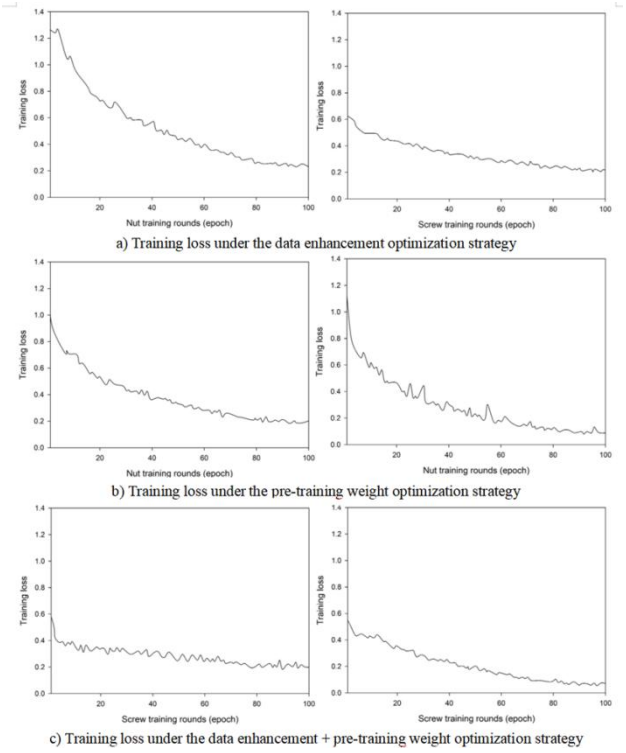


Figure 5: Training loss of the experiment with optimization strategy

4.1.3 Experiment on improving network structure

The improved ResNet50 network was used to train the nut and screw classification data set after data enhancement, and the ImageNet pre-training weights were used to initialize the model and perform 100 rounds of training, with 64 input images for each iteration training. Using Adam optimizer, the learning rate adjustment mechanism using preheat and cosine annealing, maximum and minimum learning rates are set to 0.001 and 0.0001 (learning rates greater than 0). The initial intervention in the learning rate is made by preheating treatment, and the learning rate is successively increased by 0.0001, and the learning rate reaches the maximum value in the 10th training round, and then the cosine annealing learning rate is adjusted, the period is 90. Figure 6 shows the change of loss during training.

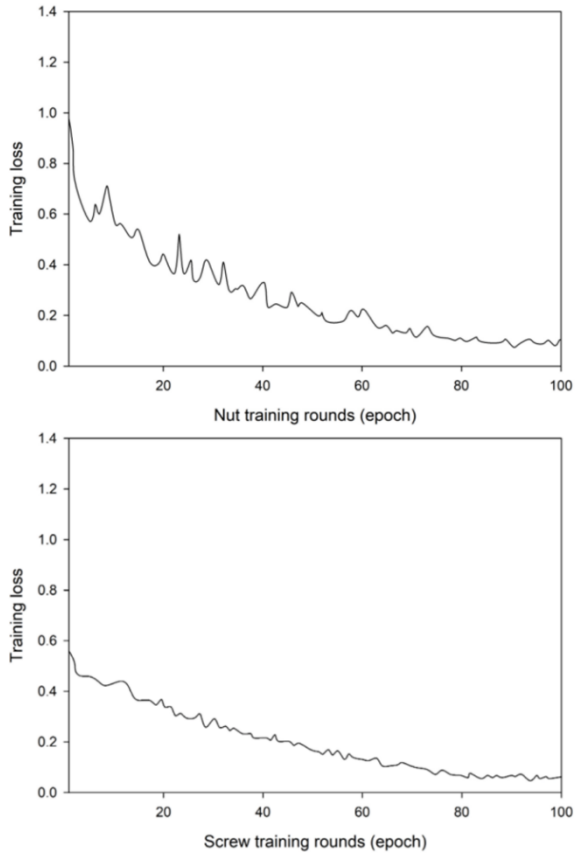


Figure 6: Training loss of the improved network structure experiment

In the graph, the horizontal coordinate represents the current training rounds, and the vertical coordinate represents the training losses. From the training results, it is found that the training loss of both nuts and screws decreases with the increase of training rounds. When the training rounds reach 100, the training instantaneous is between 0-0.2. On the whole, training rounds and training losses are inversely related, increasing and decreasing.

4.2 Result analysis

Use the trained network weights to test on the test platform. The system Windows10 used in the test platform, the processor is Intel i7-8750H, the memory is 16G, and the graphics card is GTX1050 MAX-Q. Random extraction was carried out for each category test set by random extraction, and the number was set to 50. After the extraction, the images were trained and the training weights were obtained for prediction. The results were shown as follows:

Table 2: Identification results

Option	Nut recognition accuracy rate	Correct rate of screw identification
Basic scheme	0.83	0.80
Data enhancement	0.89	0.86
Pre-training weight	0.90	0.84
Data enhancement + pre-training weight	0.92	0.88
Improved network structure	0.95	0.90

In the basic scenario, the original data set without data enhancement was directly used for training on ResNet50. Because few samples were trained, and no learning adjustment and transfer learning interventions were added, the recognition effect of the algorithm was not ideal, and the obtained recognition accuracy rate was 0.83 for nuts and 0.80 for screws. Using data enhancement alone, it was found that the nut recognition accuracy was 0.89 and screw recognition accuracy was 0.86, which were improved by 0.06 and 0.06, respectively, compared with the basic scheme. After the intervention with the pre-training weight, the nut recognition accuracy is 0.90 and screw recognition accuracy is 0.84, which is 0.07 and 0.04 higher than the basic scheme, respectively. Methods such as data enhancement, transfer learning and learning rate adjustment were used for training. As the number of samples participating in training was increased, the training method was optimized, enhanced recognition quality, with the nut recognition accuracy of 0.92 and screw recognition accuracy of 0.88. Using the improved network for training, after network optimization and adjustment, the corresponding model has a strong expression ability, which reduces the probability of feature image information loss to a certain extent. Compared with the network without optimized adjustment, the recognition accuracy will be improved, and the results confirm this view. For example, the recognition rate of nuts and screws is 0.95 and 0.90, respectively, which has higher recognition strength. In summary, using data enhancement, transfer learning, learning rate adjustment and improving the ResNet50 network can comprehensively improve the recognition accuracy of the algorithm.

5 Summary

To sum up, this paper interferes with the data by means of data enhancement technology, transfer learning method and learning rate adjustment, which makes the data sound and stable and establishes a basis for subsequent bolt defectidentification. In addition, the ResNet50 network is improved, which further strengthens the expression ability of the network and guarantees the soundness of the feature map information. From the empirical results, the initial recognition rate of nuts was only 0.83, and after optimization, the recognition rate reached 0.92; The screw was increased from 0.80 to 0.88, which confirmed the feasibility of the scheme. After further improving the network structure, it is found that the recognition rates of nuts and screws reach 0.95 and 0.90 respectively, it means that the method can improve the efficiency and quality of bolt defect identification.In terms of novelty, it breaks the traditional core setting and replaces it with 7x7 core, reduces the number of parameters in the original network, improves the expression ability of the network, and makes a paving for the identification accuracy of the transmission tower.At present, only tower bolts have been identified, and future studies will identify more categories such as tower insulators and bird's nests. At the same time, there is still room for improvement in the recognition accuracy rate. In the future, we will try to combine traditional image processing methods to further improve the recognition performance of the algorithm.

Funding

This study was supported by State Grid Shanxi Electric Power Company Technology Project: Research and Application of an Online Monitoring System for Bolt Looseness in Transmission Towers Based on DSP Voiceprint Information (52051K240001).

References

[1] Zhou D (2024). Overview of the development of China's green power industry. *China and Foreign Energy*, 29(03), pp. 76.

[2] Yang H, Zhou H, Gao B, et al. (2025). Non-contact sensing method for bolt loosening defects of transmission tower. *Yunnan Electric Power Industry*, (01), pp. 20-25+36.

[3] Xie S (2023). Hidden Danger automatic detection Method of High voltage Transmission line based on Deep Learning Algorithm. *Manufacturing and Upgrading Today*, (12), pp. 37-39.

[4] Wang G, Liang Y (2023). Research on the application of new energy-saving wires in high voltage transmission lines. *Energy Conservation and Environmental Protection*, (11), pp. 58-62.

[5] Li B, Li X, Hou S, et al (2021). Surface electric field Calculation Method of 5G Communication Equipment Based on HV transmission line sharing tower. *Southern Power Grid Technology*, 15(10), pp. 65-71. DOI: 10.13648 / j. carolcarrollnki issn1674-0629.2021.10.009.

[6] Wang T, Yan L, Chen D, et al. (2018). Research status of Power transmission line tower Stress analysis. *Science and Technology Innovation and Application*, (35), pp. 71-72.

[7] Tao H, He G, Yang J, et al. (2023). MIMU based transmission tower bolt state recognition. *Journal of Vibration and Shock*, 42(20), pp. 98-104. DOI: 10.13465 / j. carolcarrollnki JVS. 2023.20.012.

[8] Liu Y, Chen F, Xie H, et al. (2021). Rapid Detection and positioning Diagnosis of power transmission tower bolts based on Elastic Shock Wave technology. *Electronic Design Engineering*, 29(13), pp. 87-90. DOI:10.14022/ j. ssn1674-6236.2021.13.019.

[9] Zhang A, Deng F (2020). Recognition scheme of external broken vibration of transmission tower based on CNN-RVM. *Computer Simulation*, 37(04), pp. 76-80.

[10] Yang Z, Ao W, Fei X, et al (2020). Research on Intelligent Recognition of transmission tower based on high resolution SAR image and deep learning. *Electrical Measurement and Instrumentation*, 57(04), pp. 71-77. DOI:10.19753/ j. isssn1001-1390.2020.04.012.

Chinese Medical Named Entity Recognition Using Pre-Trained Language Models and an Efficient Global Pointer Mechanism

Xu Zhang[1*], Feihong Li[2], Chenlong Li[2], Xufeng Yu[2]
[1]College of Software Engineering, Xiamen University of Technology, Xiamen 361000, China
[2]College of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361000, China
[*]E-mail: zeronumber@sohu.com
[*]Corresponding Author

Keywords: MNER, data augmentation, RoBERTa, Word2Vec, EGP

Received: January 14, 2025

Medical Named Entity Recognition (MNER) is a critical task in medical text mining, serving as a foundation for intelligent diagnosis, disease prediction, and related tasks. However, Chinese medical texts present unique challenges, such as diverse entity classifications, varying entity lengths, and complex entity nesting, which hinder NER performance. To address these issues, we propose a Chinese MNER method leveraging pre-trained models and the efficient global pointer (EGP). Our approach incorporates three key innovations: (1) data augmentation techniques, including easy data augmentation and remote supervision with a medical entity dictionary, to address imbalanced entity types distribution and varying entity lengths; (2) a character-word fusion strategy that integrates RoBERTa-generated character vectors and Word2Vec-generated word vectors, enhancing semantic representation; and (3) an improved decoding layer based on EGP, enabling efficient recognition of both nested and flat entities while reducing computational costs. Experimentals show that our method achieves F1 scores of 75.87% and 92.77% on the CMeEE-V2 and CCKS2020 datasets, respectively, outperforming the RoBERTa-BiLSTM-CRF baseline by 3.06% and 4.38%, respectively.

Povzetek: Članek predstavlja metodo za prepoznavanje medicinskih poimenovanj v kitajskih besedilih, ki združuje predtrenirane modele in učinkovito globalno kazalo (EGP). Poudarek je na izboljšanju prepoznavanja zapletenih, gnezdenih entitet s strategijami, kot so povečanje podatkov in združevanje besednih ter znakovnih vektorjev, kar pripomore k večji kvaliteti.

1 Introduction

The exponential growth of medical informatics has led to the accumulation of a vast amount of medical text data, encompassing a multitude of domains such as medical journals, electronic medical records, medical encyclopedias and medical textbooks [1]. These resources, primarily in narrative form, represent a valuable repository of medical knowledge. In recent years, the rapid development of artificial intelligence (AI) technology has made data mining and relational extraction a key research area at the intersection of AI and medicine. Medical Named Entity Recognition (MNER), a core component of natural language processing (NLP), plays a critical role in medical text mining [2]. It serves as the foundation for tasks, such as medical entity relationship extraction and event extraction, while also providing high-quality data for downstream applications like intelligent medical question-answering systems and disease prediction models [3, 4]. Specifically, MNER identifies and categorizes nouns or fixed expressions in medical texts, including entities such as diseases and drugs. Accurate disease entity recognition is essential for understanding disease etiology, diagnosis, and prevention research, while drug entity recognition provides valuable insights for treatment methods. Thus, further research and optimization of MNER technology are crucial for advancing medical informatics and intelligence. Raw corpora, which are medical texts annotated by experts, are not only a key data source for MNER research but also essential for training and validating related algorithms [2]. The performance of machine learning-based methods heavily relies on the quality of annotated corpora. Therefore, high-quality, diverse, and comprehensive raw corpora are critical for MNER development. However, due to the high cost and expertise required for annotation, publicly available medical text datasets remain scarce. To address this, domestic and international initiatives have organized evaluation tasks to provide research data. For example, the N2C2 (National NLP Clinical Challenges) shared task, held annually since 2006, reflects advancements in international medical NLP [5]. Similarly, CCKS (China Conference on Knowledge Graph and Semantic Computing) and CBLUE (Chinese Biomedical Language Understanding Evaluation) have introduced the Chinese Medical Named Entity Extraction Evaluation Task Corpus and the CMeEE Series Corpus, providing substantial data and benchmarks for Chinese MNER research [1, 6]. These corpora not only promote standardized datasets in the medical industry but also support comprehensive research on Chinese MNER.

Due to the delayed development of medical informatics in China, early studies on Chinese MNER were largely influenced by English MNER research [7]. However, significant differences exist between English and Chinese, particularly in linguistic features. English has clearly defined word boundaries and relatively stable grammatical-syntactic structures, whereas Chinese medical texts present unique challenges, including a wide variety of entity types with significant scale differences, varying lengths of named entities, lengthy textual contexts, and underutilization of vocabulary information. As a result, text feature extraction models designed for English often underperform when applied to Chinese medical texts. Furthermore, Chinese medical texts contain not only flat entities but also a substantial number of structurally complex nested entities. Most mainstream MNER models treat the task as a sequence labeling problem. For example, the BERT-BiLSTM-CRF model first encodes text using BERT, then processes the results with BiLSTM to capture sequence information, and finally generates labeled outputs using CRF [8]. Although sequence labeling has achieved promising results in many evaluations, it struggles to accurately identify nested entities due to the limitations of the BIO annotation scheme, which allows only one label per token. The unique characteristics of Chinese medical texts and the complexity of nested entities pose significant challenges for Chinese MNER.

To tackle these challenges, this study focuses on three key research questions (RQs):

RQ1: How does the combination of Easy Data Augmentation (EDA) and remote supervision improve the model's ability to handle imbalanced entity types distribution and varying entity lengths in Chinese MNER?

RQ2: How does the integration of RoBERTa and Word2Vec enhance the semantic representation of medical texts, and what is its impact on the performance of Chinese MNER models?

RQ3: How does the Efficient Global Pointer (EGP) strategy optimize the decoding process, and what are its effects on computational efficiency and nested entity recognition in Chinese MNER?

The main contributions of this paper are as follows:

We propose a comprehensive data enhancement strategy combining EDA and remote supervision to mitigate the impact of imbalanced entity types distribution and varying entity lengths, thereby improving model recognition performance. Experimental results show that this strategy increases the F1 score for the "ite" entity type in the CMeEE-V2 dataset by 1.7%, outperforming the baseline model.

We design a word feature fusion method that integrates RoBERTa and Word2Vec to enhance semantic representation and model performance. Compared to using BERT as the encoding layer, this method achieves a 0.91% improvement in F1 score on CMeEE-V2.

We introduce the EGP strategy to optimize the decoding process, improving computational efficiency and nested entity recognition. On the CMeEE-V2 with nested entities, compared to the SOTA model RoBERTa-BiLSTM-CRF, EGP achieves a 3.06% improvement in F1 score and an increase of more than 15% in computational efficiency.

The rest of the paper is structured as follows. Section 2 describes the work related to traditional and machine learning-based NER. Section 3 details the proposed Chinese MNER method using pre-trained models and an EGP mechanism. Section 4 presents our experimental results, while Section 5 provides a comprehensive discussion of the findings. Finally, the conclusion is presented in Section 6.

2 Related works

The study of Chinese MNER models is a critical area in NLP research. Its technological evolution has transitioned from dictionary-based and rule-based approaches to machine learning methods. Recently, the rapid advancement of machine learning and the rise of deep learning have led to significant progress in Chinese NER tasks. This chapter aims to explore machine learning-based NER methodologies in detail, providing a thorough discussion and analysis of their key aspects.

2.1 Dictionary-based methods

Dictionary-based methods use matching algorithms to identify named entities by constructing a specialized dictionary of Chinese medical entities and their categories. Given the specialized nature of Chinese medical texts and the ambiguity of entity boundaries, a high-quality dictionary is essential for accurate entity recognition. Although Yang et al. developed a large-scale, standardized medical corpus, the dictionary cannot be updated in real-time to include new entities [9]. As a result, dictionary methods are now more commonly used as feature inputs, combined with other techniques to improve named entity recognition. For example, Wang et al. integrated a lexicon with a deep neural network, significantly enhancing the recognition of rare entities [10].

2.2 Rule-based methods

Rule-based methods require analyzing Chinese medical texts and constructing rule templates, which are then applied to entities of the same type for NER through pattern matching [11]. Compared to dictionary-based approaches, rule-based methods are easier to maintain and can better handle the non-standardized nature of Chinese medical texts. However, they demand significant time and human resources, and their rule portability is limited. Recent studies have demonstrated that combining rule-based techniques with machine learning algorithms can significantly improve the model's ability to recognize specific entity types [12]. For example, Chen et al. used external rules to extract entities that the model failed to identify, enhancing overall performance [13].

2.3 Machine learning methods

The application of machine learning techniques has significantly advanced Chinese MNER. This approach

involves training a model on labeled data and using it to perform NER on unlabeled text [14]. Compared to dictionary-based and rule-based methods, machine learning not only improves recognition efficiency but also adapts better to new entity types and domains, significantly reducing the need for human intervention and time. In the following sections, we will explore the application of machine learning in Chinese MNER from three perspectives: statistical machine learning, deep learning, and hybrid models.

2.3.1 Statistical machine learning methods

In Chinese MNER, supervised machine learning models are widely used, including Support Vector Machine (SVM), Hidden Markov Model (HMM), and Conditional Random Fields (CRF) [15-17]. For instance, Ju et al. used SVM to accurately identify medical nouns in complex biomedical texts, demonstrating its strong classification capabilities [18]. Gao et al. combined HMM with weighted learning techniques to develop a transfer learning model, significantly reducing the need for labeled data in the target domain [19]. Additionally, Liu et al. explored the impact of diverse feature types in Chinese MNER, achieving improved results by integrating medical dictionaries with CRF algorithms [20]. Despite these advancements, traditional machine learning methods require extensive feature engineering, which is time-consuming and resource-intensive. Moreover, their simple structures and limited parameter sizes restrict their representation and generalization capabilities, often leading to suboptimal performance on complex and diverse medical text data.

2.3.2 Deep learning methods

Deep learning has achieved remarkable success in areas such as speech and image recognition and has become a vital tool for NLP tasks [7]. In Chinese MNER, deep learning methods have eliminated the need for time-consuming feature engineering while enabling the identification of more complex features. Early deep learning models applied to Chinese MNER included convolutional neural networks (CNNs) and recurrent neural networks (RNNs) [21, 22]. Notably, improved RNN-based methods have achieved exceptional performance in this field. For example, Dey et al. introduced a gating mechanism based on RNNs and developed the gated recurrent unit (GRU), which effectively addressed the gradient vanishing problem common in traditional RNN training [23]. Later, Xu et al. proposed a Chinese MNER model based on bidirectional long short-term memory (BiLSTM), which captures rich bidirectional contextual information in Chinese medical texts, significantly improving recognition accuracy and efficiency [24].

In recent years, pre-trained models have gained increasing attention in Chinese MNER research. In 2018, Devlin et al. introduced BERT, which set new benchmarks in NLP tasks due to its deep bidirectional representation and Transformer architecture [25, 26]. Zhong et al. developed the BERT-Span model for recognizing rehabilitation medicine entities in Chinese medical texts. This approach uses BERT to extract feature vectors and a span model to annotate entities, enhancing recognition performance [27]. Additionally, with the rapid development of NLP technology, traditional word vector models like Word2Vec and GloVe, as well as advanced pre-trained models such as MC-BERT, ChineseBERT, and RoBERTa, have been widely adopted in Chinese MNER research [28-32]. These models not only leverage BERT's strengths but also optimize it for medical applications, bringing new insights and innovations to the field.

2.3.3 Hybrid model methods

Hybrid models have emerged as a mainstream approach in MNER research. These models combine multiple types of features, architectures, or decoding strategies to leverage their strengths and improve overall recognition performance. For example, Gridach et al. proposed a BiLSTM-CRF neural network architecture that captures both character-level and word-level representations, enabling the automatic learning of rich contextual information in biomedical texts and significantly enhancing MNER accuracy [33].

To address the unique characteristics of Chinese medical texts, researchers have proposed various hybrid model methods. For example, Kimura et al. introduced multi-task learning for text classification, using NER as an auxiliary task. This approach not only improved text classification performance but also opened new directions for NER research [34]. Similarly, Wang et al. developed a BiLSTM-CRF model with a multi-task learning mechanism, enhancing the recognition of diverse named entity categories in medical texts [35]. In contrast, Ji et al. integrated an attention mechanism between BiLSTM and CRF, enabling the model to assign varying weights to each character and better capture critical information [36]. Jiang et al. proposed a BERT-BiLSTM-CRF model for Chinese electronic medical records, leveraging BERT's pre-training capabilities to obtain richer semantic representations [8]. To simplify the complex structure of BiLSTM, Qin et al. replaced BiLSTM with BiGRU, resulting in the BERT-BiGRU-CRF model [37]. Su et al. designed a global pointer decoding strategy, which addressed nested NER challenges through a unique entity scoring matrix and loss function, achieving impressive results on Chinese medical texts [38]. Additionally, Zhang et al. introduced a lexical enhancement approach based on the RoBERTa-Global Pointer model, integrating lexical information into character-level representations to improve semantic understanding and adaptability to Chinese medical texts [39].

In summary, hybrid models show great potential and broad development prospects in Chinese MNER. As shown in Table 1, existing state-of-the-art models face limitations such as inadequate use of pre-trained models and medical lexical information, difficulties in handling nested entities efficiently, and performance degradation due to data imbalance. Our proposed model targets these challenges, aiming to improve Chinese MNER

performance and provide stronger technical support for medical information processing, clinical decision support, and related applications.

Table 1: Summary of state-of-the-art models in Chinese MNER

	Model	Datasets	F1 (%)	Key limitations
1	CRF [17]	MSRA, LDC	84.73, 76.18	Heavy reliance on feature engineering and weak contextual modeling capabilities
2	SVM [18]	GENIA	82.44	Heavy reliance on feature engineering and difficulty in capturing contextual semantics.
3	BioTrHMM [19]	GENIA V3	65.41	Reliance on handcrafted features and weak ability to capture contextual semantics
4	BERT-CRF [26]	CCKS2017, CCKS2019, CMeEE	90.05, 77.19, 62.1	Difficulty in handling medical long texts and inability to recognize nested entities
5	BERT-span [27]	CRMNER	84.75	Difficulty in handling medical long texts and a large number of negative samples
6	MC-BERT-CRF [30]	CCKS2017, CCKS2019	92.03, 84.23	Difficulty in handling long texts and inability to recognize nested entities
7	ChineseBERT [31]	Weibo, MSRA	69.02, 99.14	Limited nested entity handling and poor domain adaptability
8	RoBERTa-CRF [32]	CCKS2017, CCKS2019	92.49, 79.85	Cannot recognize nested entities and underutilizes textual lexical information
9	BiLSTM-CRF [33]	JNLPBA	72.17	Weak contextual modeling capabilities and poor nested entity handling
10	Multi-Task-BiLSTM-CRF [35]	NCBI-Dis, JNLPBA	86.14, 73.52	Potential conflicts between tasks and insufficient utilization of textual semantic information
11	Attention-BiLSTM-CRF [36]	CCKS2018	87.26	The underuse of pre-trained models, noise from attention mechanisms, and failure to detect nested entities
12	BERT-BiLSTM-CRF [8]	CCKS2017	93.25	High model complexity and limited nested entity handling
13	BERT-BiGRU-CRF [37]	AIMEMR	90.38	Lack of domain-specific focus in the medical field and inability to recognize nested entities
14	BERT-GP [38]	CLUENER, CCKS2017, CMeEE	79.44, 94.71, 66.76	Insufficient utilization of character/word information, limited domain adaptability, and unnecessary computational resource consumption
15	RoBERTa-GP-SoftLexicon [39]	CCKS2017, CCKS2019	95.56, 85.89	Fails to address the issue of data imbalance in the dataset, and the decoding layer has room for computational resource optimization

3 Research method

This paper proposes a Chinese MNER model that integrates a pre-trained model with an efficient global pointer (EGP). The model consists of two main components: an encoding layer and a decoding layer. In the encoding layer, the model takes text as input and combines token-level vectors from RoBERTa with word-level vectors from Word2Vec, producing character-word fusion vectors enriched with semantic and lexical features. In the decoding layer, the EGP receives these fusion vectors and constructs an entity matrix, which is used to score and identify entities, determining their corresponding types. The overall framework of the model is shown in Fig. 1, which illustrates the outputs for the example: the microbial entity “virus” and the body type entity “target cell.”

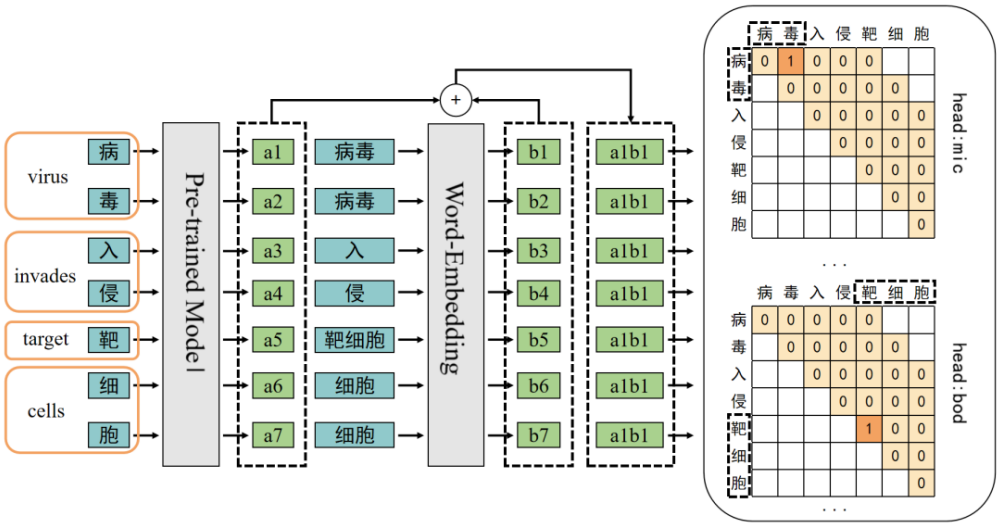


Figure 1: General framework of the model

3.1 Data augmentation

The performance of NER models heavily relies on high-quality annotated corpora, especially for Chinese medical texts. Constructing such corpora requires substantial time and effort due to the specialized nature of the domain. However, existing Chinese MNER datasets often struggle with issues like under representation of rare entity types (e.g., medical devices and laboratory tests) and significant variations in entity lengths (e.g., drug names ranging from short terms to long descriptive phrases). To address these challenges, data augmentation (DA) techniques have become a popular solution. Common DA methods for Chinese MNER include Easy Data Augmentation (EDA), remote supervision, and bootstrapping [40-42]. This paper proposes a DA strategy combining EDA and remote supervision, which aims to alleviate entity type imbalances and handle variable-length entities. By improving the diversity and quality of training data, this approach enhances the performance and generalization of the MNER model, particularly for rare and long entities.

3.1.1 Dataset's problem statement

The experiments in this paper use datasets from two authoritative Chinese MNER benchmarks, CBLUE and CCKS: the CMeEE-V2 nested dataset and the CCKS2020 flat dataset. Both datasets capture the unique characteristics of real-world Chinese medical texts but differ significantly in data volume, entity categories, and text formats. Specifically, the CMeEE-V2 dataset classifies medical entities into 9 categories: drug (dru), medical test item (ite), medical equipment (equ), microbiology class (mic), body part (bod), disease (dis), clinical symptom (sym), department (dep), and medical procedure (pro). In contrast, the CCKS2020 dataset groups entities into 6 categories: anatomical sites (ana), imaging tests (ima), drug (dru), diseases (dis), surgeries (sur), and laboratory tests (lab).

Preliminary experiments show that entity length significantly impacts recognition performance. Generally, longer entities are less frequent, making it harder for the model to learn meaningful features and leading to poorer recognition results. Based on these findings, this paper classifies Chinese medical entities into three length categories: short (1-3 characters), medium (4-7 characters), and long (8+ characters). Fig. 2 shows the proportional distribution of entities by length in the datasets. As seen in the Figure, "dis," "sym," and "pro" entities in the CMeEE-V2 dataset, as well as "dis" entities in the CCKS2020 dataset, contain a higher proportion of long entities, which are more challenging to recognize.

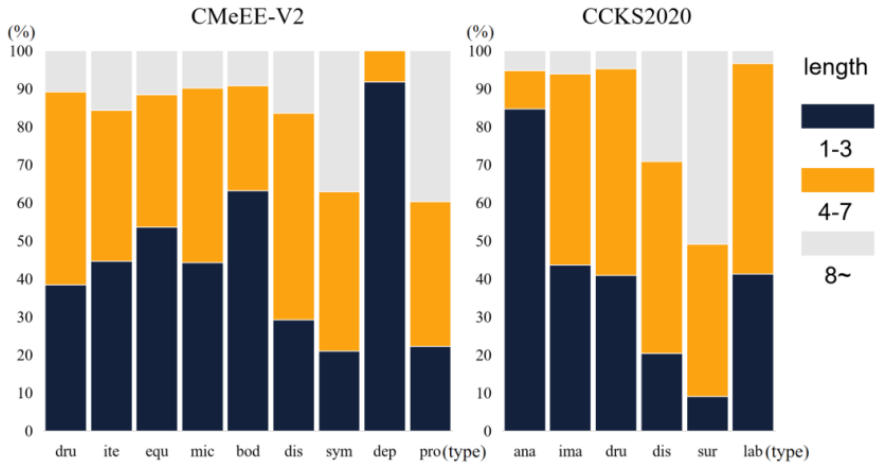


Figure 2: Percentage of different entity lengths in the datasets

Additionally, entity types with limited data are a major challenge in Chinese medical named entity recognition. The imbalanced entity types distribution and varying entity lengths stems from insufficient high-quality annotated data for these entities. Current deep learning models require large amounts of annotated data for effective training, and without it, their ability to learn and recognize these entities is severely limited, resulting in poor performance. To address this issue, this paper uses two data augmentation strategies EDA and remote supervision to expand imbalanced data and improve the model's learning capability.

3.1.2 EDA

In Chinese MNER, EDA is a widely used and efficient data augmentation technique. It introduces reasonable noise into the text to simulate the complexity and diversity of real-world samples. Compared to traditional EDA methods, this paper proposes an enhanced EDA approach with three strategies tailored to Chinese medical texts:

(1) Synonymous Entity Substitution: To address imbalanced entity type distributions, this technique replaces entities with semantically similar ones of the same type, focusing on underrepresented entities. This increases the data volume for rare entities, helping the model learn robust features and improve recognition accuracy.

(2) Entity Augmentation: To improve recognition of long entities, this technique constructs longer named entity sequences by combining similar short entities of the same type using conjunctions or punctuation. By increasing the number of long entity samples, it enables the model to better learn structural and semantic patterns of long sequences, enhancing performance.

(3) Sentence Replacement: To help the model learn contextual dependencies and adapt to the lengthy nature of medical texts, this technique replaces two sentences within or between samples with a randomly selected sentence, using punctuation or separators as the basis for exchange. By altering sentence order and combinations, it generates diverse contextual environments, improving the model's ability to learn long-distance dependencies.

These improved strategies not only retain the efficiency of EDA but also better align with the characteristics of Chinese medical texts, enhancing the performance and robustness of Chinese MNER.

3.1.3 Remote supervision

Remote supervision is an effective data augmentation technique that leverages existing knowledge bases or structured data to automatically annotate large-scale unlabeled texts. In Chinese MNER, remote supervision is commonly used by introducing a large corpus of unsupervised medical texts and automatically annotating them using script programs. This approach effectively converts unsupervised samples into richly annotated supervised data, significantly expanding the dataset for model training.

The second DA method proposed in this paper extends remote supervision with an innovative approach. First, we construct a comprehensive medical entity dictionary based on a large annotated MNER training set. To improve the dictionary's coverage and accuracy, we use ERNIE Bot, a generative pre-trained large model optimized for Chinese, to generate additional instances of rare entity types and long entities, addressing imbalances in entity type and length distributions. Next, we apply entity matching techniques to automatically annotate unsupervised medical texts using the expanded dictionary. To ensure annotation quality, the automatically labeled data is further processed and validated by ERNIE Bot, which performs contextual verification for semantic consistency, corrects mislabeled entities, and assigns confidence scores, retaining only high-confidence annotations for training.

This approach not only increases the volume of training data but also ensures its high quality, providing the MNER model with more robust and diverse samples for improved performance, especially for rare and long entities.

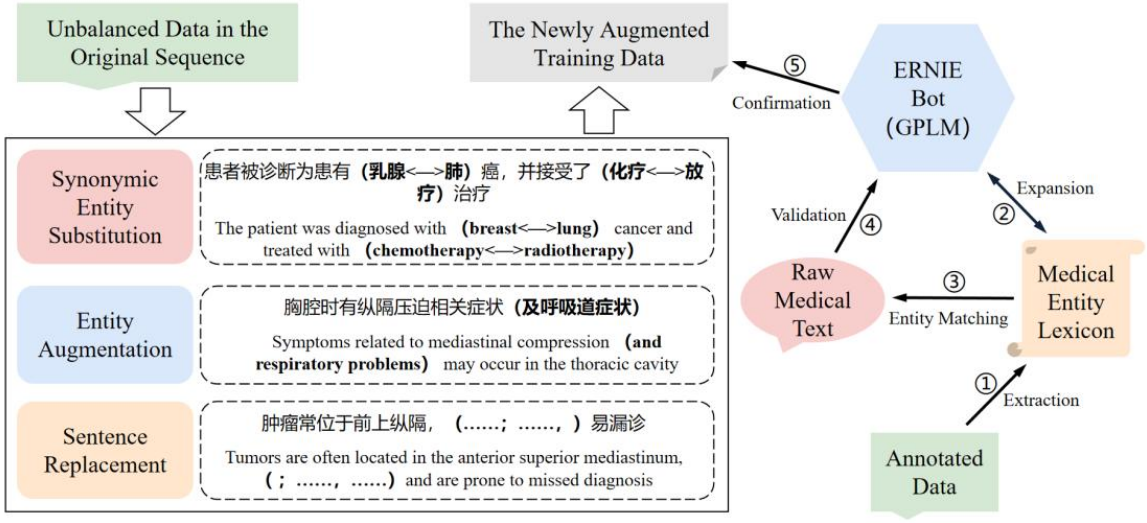


Figure 3: Data augmentation process

This paper proposes an innovative method to address poor recognition performance for specific entities caused by imbalanced entity types and varying entity lengths. The method combines two data augmentation strategies: EDA and remote supervision. The data augmentation process is illustrated in Fig. 3, with EDA on the left and remote supervision on the right. By enhancing EDA operations for Chinese medical texts—synonymous entity substitution, entity augmentation, and sentence replacement—this approach not only mitigates entity type and length imbalances but also improves the model's ability to process long medical texts. Additionally, this paper integrates a medical entity dictionary with the GPLM ERNIE Bot using remote supervision, effectively converting a large volume of unsupervised medical texts into high-quality annotated data. This significantly expands the training dataset, particularly for long entities and rare entity types [43]. Furthermore, the annotation process introduces new syntactic structures and nested entities, enriching dataset diversity. The combination of these two strategies not only enhances the performance and robustness of Chinese MNER but also provides new insights and methodologies for tackling similar imbalanced dataset challenges.

3.2 Encoding layer

In Chinese MNER, the encoding layer is the core component of the model architecture, responsible for converting raw Chinese medical text into numerical vectors that computers can process. Given the complex contextual dependencies and specialized medical domain vocabulary in medical texts, the design of the encoding layer is critical for improving NER accuracy and efficiency. This model uses a fusion encoding strategy combining RoBERTa and Word2Vec, generating character-word fusion vectors that capture both deep semantic information and word-level features.

3.2.1 RoBERTa

The core component of the encoding layer in the proposed Chinese MNER model is RoBERTa, which generates dynamic character vectors. As an optimized version of BERT, RoBERTa has shown excellent performance in NLP tasks, including NER. RoBERTa and BERT share several key similarities: both are based on the Transformer architecture, using the encoder component to capture bidirectional contextual information, allowing them to consider both preceding and following contexts when processing text and produce more accurate token vector representations. Both also use the masked language model (MLM) as their pre-training strategy, where MLM randomly masks words in the input text and trains the model to predict them, enabling the model to learn comprehensive linguistic knowledge and contextual information. Additionally, both models have similar structures, consisting of multiple stacked Transformer encoders, each incorporating a self-attention mechanism and a feed-forward neural network, which allows them to efficiently capture dependencies and semantic information in long texts. Both RoBERTa and BERT are versatile and applicable to a wide range of NLP tasks, such as text classification, NER, and question answering. In MNER, they significantly enhance model performance and improve the understanding and recognition of critical information in medical texts.

The RoBERTa encoding layer used in this paper not only retains BERT's core advantages but also introduces several optimizations to improve performance. These improvements include:

(1) Dynamic Masking: RoBERTa uses a dynamic masking strategy, generating a unique mask pattern for each input sequence. This helps the model better capture language variability and contextual dependencies.

(2) Remove the Next Sentence Prediction Task: By eliminating this task, RoBERTa focuses more on learning linguistic structures and semantics, which is particularly

useful for understanding the complex characteristics of Chinese medical texts.

(3) Increasing Training Data and Batch Size: RoBERTa is trained on 160 GB of data with a batch size of 8,000, enabling the model to learn richer medical knowledge and language patterns.

(4) Extending Training Text Length: RoBERTa processes longer sentences, enhancing its ability to handle complex text structures and dependencies, which is critical for lengthy Chinese medical texts.

In this study, we use the RoBERTa model fine-tuned by Harbin Institute of Technology (HIT) for Chinese text [44]. This version of RoBERTa has been specifically adapted to Chinese linguistic features, such as word segmentation and semantic structures. The fine-tuning process involved training on a large-scale Chinese corpus, primarily Chinese Wikipedia data, with two configurations: (1) a sequence length of 128 and a batch size of 2,560 for 100 k steps, and (2) a sequence length of 512 and a batch size of 384 for another 100 k steps. This extensive fine-tuning allows the model to better capture Chinese language nuances and improve performance on tasks like NER.

For Chinese MNER, text data is characterized by frequent domain-specific terminology, complex semantic structures, and longer sentence lengths compared to general text. These characteristics make RoBERTa, with its optimizations and Chinese-specific fine-tuning, an ideal choice for the encoding layer in our model.

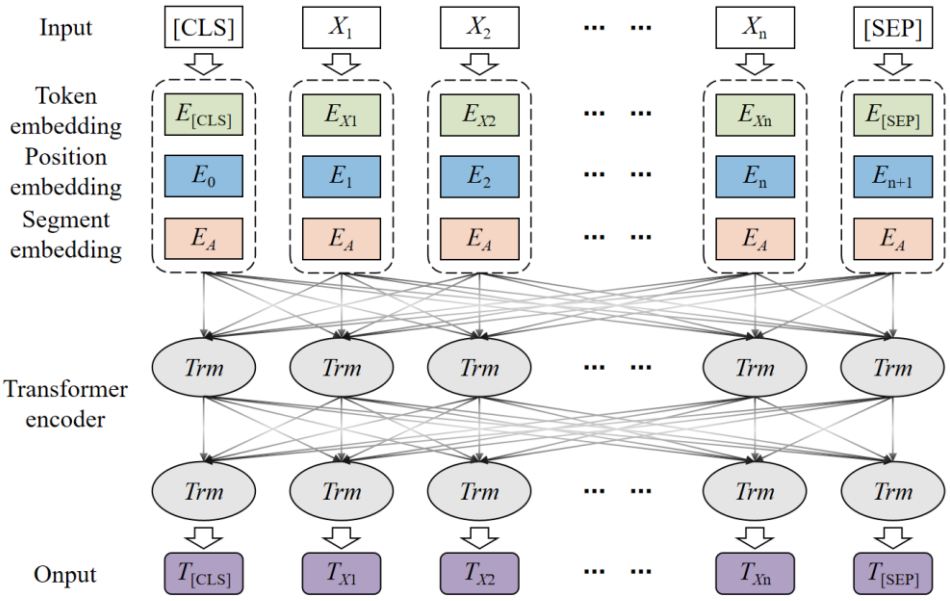


Figure 4: RoBERTa generation encoding vector process

Fig. 4 illustrates the process of encoding Chinese medical text sequences into token vectors using RoBERTa. The first step involves tokenizing the input Chinese medical text sequence using a dedicated tokenizer. Due to the unique characteristics of Chinese text, the tokenizer divides the sequence into character-level tokens, with each character serving as the basic unit. Additionally, the RoBERTa tokenizer automatically inserts special tokens [CLS] and [SEP] to mark the start and end of the text sequence. Next, the character tokens are passed through RoBERTa's embedding layer, which performs token embedding, position embedding, and segment embedding. However, in the Chinese MNER task studied here, segment embedding is typically unnecessary since the input usually consists of a single paragraph. To improve the accuracy of character sequence positional information, this study uses Rotary Position Embedding (RoPE), making positional embedding an optional step [45]. After embedding, the vectors are fed into RoBERTa's core, which consists of multiple stacked Transformer encoders. Each encoder includes self-attention mechanisms and feed-forward neural networks, enabling it to capture contextual and semantic information and learn more nuanced semantic representations of Chinese medical text. Finally, after processing by multiple Transformer encoders, the model outputs a sequence of vector representations for each character token. These vectors capture both the intrinsic information of the characters and their contextual semantics, making them suitable for downstream tasks like Chinese MNER.

Given a Chinese medical text sequence $\left[x_1, x_2, \cdots, x_n\right]$, the context vector representation of the i-th character x_i, is defined by the following Eq. (1):

$$c_i = RoBERTa\left(x_i\right) \tag{1}$$

3.2.2 Word2Vec

Word2Vec is a deep learning model designed to generate word vectors, converting natural language words into numerical vectors for use in various NLP tasks. Other word vector models, such as Global Vectors for Word Representation (GloVe) and FastText, have also shown

strong performance across tasks [29, 46]. Word2Vec employs two main methods: the Skip-gram model, which predicts context words based on a central word (ideal for low-frequency words), and the CBOW model, which predicts a central word from context words (effective for high-frequency words). The Word2Vec workflow is straightforward: text data is first converted into coded vectors, which are then fed into a neural network for training. The model optimizes its parameters through back-propagation, aiming to maximize the probability of predicting context or center words. Ultimately, it outputs continuous vector representations of words, reflecting their semantic similarities. These representations are widely used in Chinese MNER tasks.

In Chinese medical texts, words are typically composed of multiple adjoining characters, unlike English, where words are separated by spaces. This linguistic feature poses unique challenges for NER, as the model must capture both character-level contextual dependencies and word-level semantic information. To address this, we combine Word2Vec word vectors with RoBERTa character vectors, leveraging their respective strengths: Word2Vec excels at word-level semantics, while RoBERTa handles character-level contextual relationships. However, this fusion introduces a dimension inconsistency issue, as word vector sequences are shorter than character sequences. To resolve this, we modify the word vector generation process by adding special identifiers [CLS] and [SEP] at the start and end of the text, respectively, to mark sequence boundaries. Additionally, each split word is repeated n times, where n is the character length of the word. While this introduces some redundancy, it effectively simulates character-level contextual information, helping the model better understand relationships between characters within words. This is particularly important for Chinese medical NER, as many domain-specific terms are multi-character, requiring both character-level and word-level representations. The model generates word vectors according to the following Eq. (2):

$$w_i = Word2Vec(y_i) \tag{2}$$

Where is the word corresponding to the i-th character x_i of the input text, and w_i is the word vector generated by Word2Vec for the word y_i.

By combining Word2Vec and RoBERTa, our model can better handle the high frequency of domain-specific terminology and the complex semantic structures inherent in Chinese medical texts. This dual representation approach not only addresses the technical challenge of dimension inconsistency but also enhances the model's ability to comprehend and recognize medical entities in Chinese texts.

3.2.3 Character-word fusion vectors

In Chinese language processing, effectively fusing character vectors and word vectors has been shown to enhance the model's ability to capture contextual and lexical information, particularly in tasks like NER [47]. This is especially important for Chinese medical texts, where single characters often carry significant semantic meaning, particularly in multi-character domain-specific terms. By combining character-level and word-level representations, the model can better capture the nuanced relationships between characters and words, which is critical for accurate entity recognition.

Character-word fusion strategies can be broadly categorized into two approaches: (1) combining character-based word representation with the original word representation, and (2) treating characters as the fundamental semantic unit and enhancing their representations with word-level information [48]. Various techniques can be used for fusion, including vector addition, multiplication, concatenation, weighted representation via feed-forward neural networks, and tensor-based abstraction. In this study, we adopt concatenation as the fusion strategy due to its simplicity, intuitive results, and computational efficiency. Given the significant difference in dimensionality between RoBERTa-generated character vectors and Word2Vec-generated word vectors, concatenation provides an effective way to preserve complete information from both representations while maintaining manageable computational complexity.

While concatenation is our primary fusion method, future work could explore alternative strategies, such as weighted representations or tensor-based approaches, to further optimize the fusion process.

The character-word fusion vector method used in this paper can be formally expressed as Eq. (3).

$$h_i = concat(c_i, w_i) \tag{3}$$

Where c_i and w_i correspond to the word vector and word vector of the i-th character x_i, respectively, and h_i is the fusion vector obtained by concatenating the character-word vectors.

3.3 Decoding layer

In traditional Chinese MNER frameworks, the task is typically treated as a sequence labeling problem, where text is annotated using schemes like BIO or BIOES, and a CRF layer outputs label probability score. However, this approach struggles with complex medical texts containing nested or overlapping entities. Traditional sequence labeling methods assume entities are non-overlapping, making it difficult to handle nested structures directly. To address this, additional post-processing steps, such as rule-based heuristics or extra model layers, are often required, significantly increasing computational complexity and resource usage.

To overcome these limitations, we adopt a pointer network approach, which has proven effective for Chinese MNER. Pointer networks directly mark any position in the text, enabling accurate identification of nested and overlapping entities without relying on fixed label sequences like BIO or BIOES. This end-to-end approach eliminates the need for post-processing, making it more

efficient and flexible for handling complex entity structures.

Building on the strengths of pointer networks, this paper proposes a decoding layer strategy based on an EGP. The EGP incorporates a global vision component, allowing the model to comprehensively capture the text's overall information during decoding. Additionally, the EGP optimizes algorithm design to reduce the computational complexity of entity matrix scoring, providing robust support for decoding in complex Chinese MNER tasks. These features make the EGP particularly suitable for handling nested entities and other challenges in medical text analysis.

3.3.1 Entity matrices

In the Chinese MNER task, the EGP approach treats the problem as a multi-label classification by constructing a 3-dimensional matrix framework that captures all potential entities and their types in a sequence. The matrix dimensions are (Number of Entity Types, Sequence Length, Sequence Length), where each element represents the probability of a contiguous segment being an entity of a specific type. For the CMeEE-V2 dataset, which includes 9 entity types, 9 two-dimensional entity matrix slices are constructed, each focusing on a single entity type. The rows and columns of these matrices correspond to the start and end positions of entities in the sequence, respectively, as shown in Fig. 5. When the model predicts a specific segment as an entity, the corresponding matrix element transitions from 0 to 1, effectively marking the entity's boundaries.

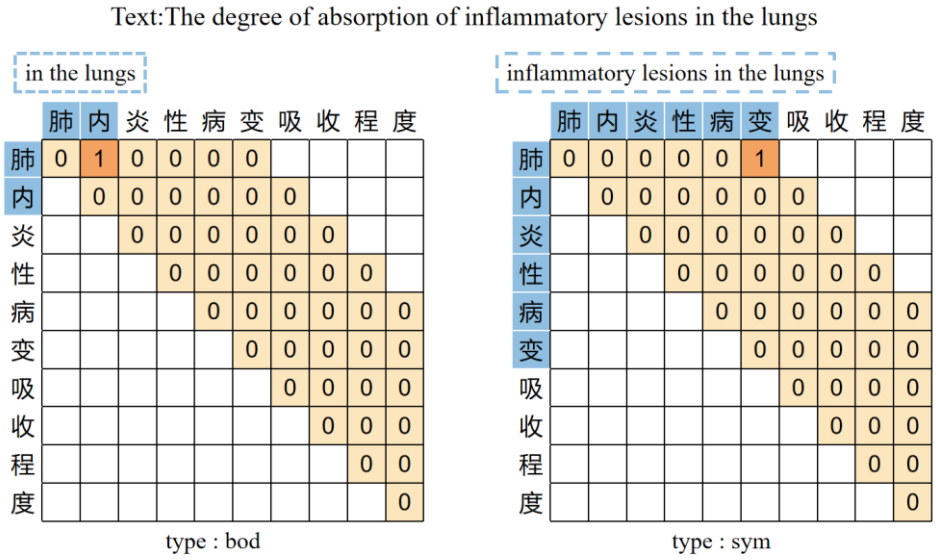


Figure 5: Entity matrices

During the construction of the entity matrix, the lower-left triangle is ignored, and a masking operation is applied to parts of the upper-right triangle, as shown in Fig. 5. This design is motivated by two key considerations:

(1) Logical Consistency: It is illogical for an entity's start position to be after its end position. Thus, calculating entity probabilities for the lower-left triangle is unnecessary and would introduce noise into the model's predictions.

(2) Entity Length Constraints: In Chinese medical texts, entity lengths often differ significantly from the overall text length. The upper-right triangle represents sequence segments that are typically longer than regular medical entities. By masking this region, we reduce unnecessary computations and focus the model on plausible entity lengths.

The masking operation not only reduces computational complexity but also improves the model's ability to identify nested entities. By eliminating invalid entity combinations (e.g., end positions earlier than start positions) and focusing on plausible entity lengths, the masking operation ensures the model can more accurately detect nested entities. For example, in the phrase “肺部炎症性病变” (inflammatory lesions in the lungs), the masking operation helps the model distinguish between the outer entity “肺部炎症性病变” (symptom type) and the nested inner entity “肺部” (body type) by restricting the search space to valid entity boundaries.

The EGP designed in this paper builds on these principles, making targeted improvements to the original global pointer approach. By optimizing computational resources and narrowing the search range for candidate entities, the masking operation enhances the accuracy and efficiency of Chinese MNER, particularly for complex nested structures.

3.3.2 Entity scoring function and loss function

The EGP is used to score entities for each element in the entity matrix. Requiring a series of transformations on the encoded character-word fusion vectors.

For an input text $[x_1, x_2, \cdots, x_n]$ of sequence length n, the sequence of word fusion vectors obtained after encoding is $[h_1, h_2, \cdots, h_n]$. The transformation equations are defined as follows:

$$q_{i,\alpha} = W_{q,\alpha} h_i + b_{q,\alpha} \tag{4}$$

$$k_{i,\alpha} = W_{k,\alpha} h_i + b_{k,\alpha} \tag{5}$$

Where h_i is the character-word fusion vector representation of the i-th character of the text, $W_{q,\alpha}$ and $W_{k,\alpha}$ are the two transformation matrices, $b_{q,\alpha}$ and $b_{k,\alpha}$ are the biases, $q_{i,\alpha}$ and $k_{i,\alpha}$ are the new vector representations obtained by transforming h_i.

To obtain the probability score that a consecutive sequence from the i-th character to the j-th character is an entity of type α, we compute the inner product of the transformed vector representations $q_{i,\alpha}$ and $k_{j,\alpha}$. The entity scoring function is defined by Eq. (6).

$$s_\alpha(i, j) = q_{i,\alpha}^{\cdot} k_{j,\alpha} \tag{6}$$

Where is the probability score of a continuous textual sequence $[x_i, \cdots x_j]$ predicted to be of entity type α, x_i is the beginning of the entity, and x_j is the end of the entity.

The EGP approach identifies entities from a global perspective but may lack sensitivity to specific entity lengths and spans, potentially misclassifying similar entities positioned head-to-tail as the target entity. Traditional position embedding strategies assign static, predefined vectors to each position, which struggle to capture relative relationships when a single character corresponds to multiple entity boundaries, as in nested entities. This limitation is particularly problematic for Chinese medical texts, where relative character positions within multi-character terms and nested entities are critical for accurate recognition.

To address this, the EGP introduces Rotary Position Embedding (RoPE), which incorporates relative positional information into the character-word fusion vector representation through a transformation matrix, as defined in Eq. (7). RoPE enables the model to flexibly capture relative character positions, which is essential for handling nested and long entities in Chinese medical texts. For example, in the term "肺部炎症性病变" (inflammatory lesions in the lungs), RoPE helps the model accurately identify nested entity boundaries by capturing the relative positions of characters within the term. Additionally, the entity scoring function is redefined to incorporate relative positional information, as shown in Eq. (8), further enhancing the model's ability to recognize complex entity structures.

$$R_i^{\cdot} R_j = R_{j-i} \tag{7}$$

$$s_\alpha(i, j) = q_{i,\alpha}^{\cdot} R_{j-i} k_{j,\alpha} \tag{8}$$

The EGP treats NER as a multi-label classification problem, aiming to identify entities of α specific type, with a total of $(2n+1-k)k/2$ candidate entities. Here, n represents the text sequence length, k denotes the maximum sequence length that can be predicted as an entity, and $n-k$ signifies the edge length of the upper-right triangular mask. However, in Chinese medical texts, the text sequence length n is often considerable, as is the number of candidate entities. Moreover, the number of entities present in each text is likely to be relatively limited. If the loss function is designed as a $(2n+1-k)k/2$ binary classification, the model may tend to predict all samples as negative due to the overwhelming number of negative samples (non-entities) compared to positive samples (entities). This would result in a significant class imbalance issue, severely impacting the model's performance.

To address this, the EGP designs a multi-label classification loss function, which generalizes the single-objective multi-classification cross-entropy. This approach is particularly suitable for multi-label classification problems with many total categories but few target categories [49]. By independently handling each category, the model can focus on the distribution of positive and negative samples within that category. Additionally, the flexibility to assign higher weights to positive samples helps alleviate the issue of limited positive sample quantities. This makes the multi-label classification loss function better suited for texts containing entities of different types, enhancing its effectiveness for Chinese MNER tasks involving nested entities. The loss function for multi-label classification is defined in Eq. (9).

$$\begin{aligned} Loss = &\log\left(1 + \sum_{(i,j)\in P_\alpha} e^{-s_\alpha(i,j)}\right) \\ &+ \log\left(1 + \sum_{(i,j)\in Q_\alpha} e^{s_\alpha(i,j)}\right) \end{aligned} \tag{9}$$

Where P_α is the first and last set of all entities of type α in the candidate entities, and Q_α is the first and last set of all non-entities and non-entities of type α in the candidate entities. In the decoding phase, we only need to consider the sequence $[x_i, \cdots x_j]$ of $1 \le i \le j \le k$, which is considered to be the output of entities of entity type α if $s_\alpha(i, j) > 0$ is satisfied.

4 Experiment

4.1 Experimental parameters and assessment indicators

The experimental environment in this study includes a computer with an Intel i5-13600KF CPU, an NVIDIA RTX 4070Ti GPU, 32 GB RAM, and a 1 TB hard disk, which are sufficient to ensure efficient training and fine-tuning of deep learning models. For the two datasets, CMeEE-V2 and CCKS2020, we first preprocessed the data by splitting the CCKS2020 dataset into training (70%), validation (15%), and test (15%) sets and converting the data into JSON format for consistency.

Next, we conducted a meticulous fine-tuning parameter optimization process. To identify the optimal model configuration, we defined a narrow range of candidate values for key hyperparameters, including the learning rate, batch size, number of training epochs, and optimizer parameters, based on existing research and preliminary trial-and-error experiments. We then used a grid search algorithm to systematically evaluate all possible combinations of these parameters and determine the best configuration. The final parameter settings are shown in Table 2.

Table 2: Experimental parameters

Hyperparameter	Value
Learning rate	2e-5
Batch size	8
Epochs	10
Max sequence length	256
Optimizer	Adam
Warmup	0.1

The experiment uses precision (P), recall (R), and F1 score as evaluation metrics. Precision represents the ratio of correctly identified entities to the total predicted entities, recall signifies the ratio of correctly identified entities to the total actual entities, and the F1 score is a harmonic mean of precision and recall, reflecting the overall performance of the model. The formulas are as follows:

$$p = \frac{TP}{TP + FP} \times 100\% \tag{10}$$

$$R = \frac{TP}{TP + FN} \times 100\% \tag{11}$$

$$F1 = 2 \times \frac{P \times R}{P + R} \times 100\% \tag{12}$$

Where *TP* represents the number of samples where the model predicted a positive category and the actual category was positive. *FP* denotes the number of samples where the model predicted a positive category but the actual category was negative. *FN* represents the number of samples where the model predicted a negative category but the actual category was positive.

4.2 Experimental results

To comprehensively evaluate the performance of the pre-trained models combined with the efficient global pointer method (DA-RW-EGP, Data Augmentation-RoBERTa with Word2Vec-Efficient Global Pointer), we conducted a detailed comparison with several popular benchmark methods on two widely recognized Chinese medical datasets, CMeEE-V2 and CCKS2020. Given that the CMeEE-V2 dataset contains nested entities, we specifically included comparison methods capable of effectively handling nested NER. The experimental results are summarized in Tables 3 and 4, which clearly illustrate the performance of the proposed methods across several key metrics.

Table 3: CMeEE-V2 comparison experiment

	Model	Pre (%)	Rec (%)	F1 (%)
1	BERT	69.65	70.51	70.08
2	RoBERTa	70.07	70.92	70.49
3	BERT-CRF	70.93	73.24	72.07
4	RoBERTa-BiLSTM-CRF	72.90	72.72	72.81
5	RoBERTa-GP	74.01	75.05	74.53
6	RoBERTa-GP-SoftLexicon	73.87	75.59	74.72
7	**DA-RW-EGP (ours)**	**74.87**	**76.91**	**75.87**

Table 4: CCKS2020 comparison experiment

	Model	Pre (%)	Rec (%)	F1 (%)
1	BERT	86.83	88.16	87.49
2	RoBERTa	87.06	88.53	87.79
3	BERT-CRF	88.21	89.11	88.66
4	RoBERTa-BiLSTM-CRF	87.85	88.93	88.39
5	RoBERTa-GP	89.99	93.29	91.61
6	RoBERTa-GP-SoftLexicon	90.76	93.16	92.08
7	**DA-RW-EGP (ours)**	**91.52**	**94.01**	**92.77**

The experimental results on the CMeEE-V2 and CCKS2020 datasets demonstrate the effectiveness of the proposed DA-RW-EGP method. As shown in Table 3, on the CMeEE-V2 dataset, our method achieves an F1 score of 75.87%, outperforming all baseline models, including RoBERTa-GP-SoftLexicon (74.72%) and RoBERTa-BiLSTM-CRF (72.81%). Similarly, on the CCKS2020 dataset (Table 4), the proposed method attains an F1 score of 92.77%, surpassing RoBERTa-GP-SoftLexicon (92.08%) and RoBERTa-BiLSTM-CRF (88.39%). These results highlight the superior performance of the DA-RW-EGP method in Chinese MNER, particularly in handling nested entities and achieving higher precision and recall.

Statistical Significance Analysis: To validate the performance improvements of the proposed DA-RW-EGP model, we conducted t-tests on both the CMeEE-V2 and CCKS2020 datasets. For each model and dataset, we ran

multiple experiments, obtaining 10 F1 scores for each configuration. These scores were then used to perform t-tests, ensuring the robustness of our results. On the CMeEE-V2 dataset, the t-test results show a t-value of 33.14 and a p-value much smaller than 0.005, indicating that the performance improvement of DA-RW-EGP is statistically significant. Consistent results were observed on the CCKS2020 dataset, as shown in Table 5, further confirming the superiority of the proposed method. These statistical tests provide strong evidence that the performance gains are not due to random variations but reflect the true effectiveness of the model.

Table 5: t-test comparison of RoBERTa-BiLSTM-CRF and DA-RW-EGP on CMeEE-V2 and CCKS2020

Dataset	DA-RW-EGP (F1%)	RoBERTa-BiLSTM-CRF (F1%)	t	p	α
CMeEE-V2	75.842±0.032	72.778±0.043	33.14	$\ll \alpha$	0.05
CCKS2020	92.76±0.009	88.393±0.014	81,31	$\ll \alpha$	0.05

5 Discussion

5.1 Comparison with related works

The F1-score is a core evaluation metric for measuring the performance of Chinese medical named entity recognition models. In this section, we evaluate the proposed DA-RW-EGP model on two benchmark datasets: CMeEE-V2 (with nested entities) and CCKS2020 (without nested entities). As shown in Table 3, the proposed model achieves an F1-score of 75.87% on the CMeEE-V2 dataset, demonstrating its effectiveness in handling nested entities. Additionally, Table 4 shows that the model attains an F1-score of 92.77% on the CCKS2020 dataset, highlighting its robust performance in non-nested scenarios.

To contextualize these results, we compare the proposed model with several state-of-the-art (SOTA) models, including the sequence labeling models BERT-CRF [26] and RoBERTa-BiLSTM-CRF [8], as well as the pointer network models RoBERTa-GP [38] and RoBERTa-GP-SoftLexicon [39]. As summarized in Table 3 and Table 4, the proposed model outperforms BERT-CRF by 3.80% and 4.11% on the CMeEE-V2 and CCKS2020 datasets, respectively. Compared to RoBERTa-BiLSTM-CRF, the improvements are 3.06% and 4.38%. Additionally, the proposed model surpasses RoBERTa-GP by 1.34% and 1.16%, and RoBERTa-GP-SoftLexicon by 1.15% and 0.69% on the same datasets. These improvements in F1-score clearly demonstrate the superior performance of the DA-RW-EGP model in Chinese medical named entity recognition tasks.

Furthermore, to evaluate the generalization capability and performance of the proposed model, we conducted experiments on several datasets used by SOTA models in the Related Works table (Table 1). On the CCKS2017 and CCKS2019 datasets, the proposed model achieved F1-scores of 95.88% and 86.35%, respectively. On the CCKS2017 dataset, the proposed model outperformed MC-BERT-CRF [30], RoBERTa-CRF [32], BERT-BiLSTM-CRF, and BERT-GP [38] by 3.85%, 3.39%, 2.63%, and 1.17%, respectively. On the CCKS2019 dataset, the proposed model surpassed MC-BERT-CRF and RoBERTa-CRF by 2.12% and 6.5%, respectively. These results further validate the generalization ability and superior performance of the proposed model.

5.2 Reasons for superior performance

The superior performance of the proposed DA-RW-EGP model can be attributed to several key innovations, including data augmentation, character-word fusion vectors, and the EGP module. Below, we analyze each component in detail and validate its contribution through ablation studies.

5.2.1 Data augmentation

To address the issues of imbalanced entity type distributions and varying entity lengths in Chinese MNER datasets, we adopt a comprehensive data augmentation strategy. This approach generates high-quality synthetic samples for rare entity types and long entities, significantly improving the model's ability to recognize these challenging cases. Data augmentation is essential for both sequence labeling models and pointer network models, as they require large amounts of annotated data for training and validation. By enriching the training set, our method ensures balanced learning across different entity types and lengths, ultimately enhancing the model's overall performance.

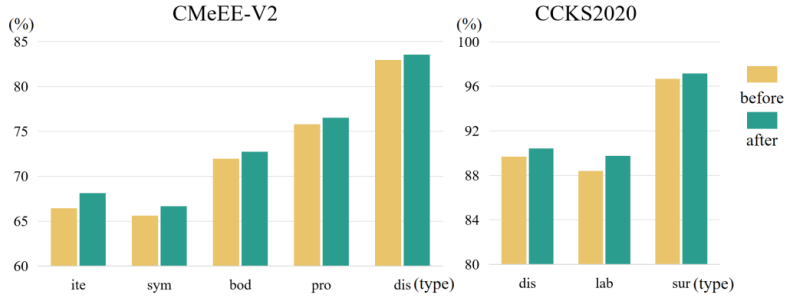


Figure 6: Comparison of F1-scores before and after data augmentation

As shown in Fig. 6, we compare the F1-scores of selected entity types before and after data augmentation on the two datasets. The results show a slight improvement in F1-scores for most entity types. For normal-scale entity types, this improvement is due to the increased number of long entities in the augmented data, enhancing the model's ability to recognize entities of varying lengths. Notably, the F1-scores for the "ite" entity type in the CMeEE-V2 dataset and the "lab" entity type in the CCKS2020 dataset increased by 1.7% and 1.3%, respectively. This improvement is attributed to the low initial data representation of these entity types. Through EDA and remote supervision, we generated a large number of high-quality pseudo-samples, improving the model's ability to recognize these rare entities. These results demonstrate that our comprehensive data augmentation strategy effectively addresses the challenges of imbalanced entity types distributions and varying entity lengths in Chinese MNER tasks.

5.2.2 Character-word fusion vectors

In Chinese MNER, medical lexical information is critical. As shown in Tables 3 and 4, the RoBERTa-GP-SoftLexicon model, which incorporates simple lexical information, outperforms RoBERTa-GP by 0.19% and 0.47% on the two datasets, respectively. In this work, we further enhance the model's encoding layer by integrating comprehensive medical lexical information using Word2Vec, generating character-word fusion vectors. This capability is absent in single pre-trained models like MC-BERT, ChineseBERT, and RoBERTa listed in Table 1. By leveraging this approach, our proposed DA-RW-EGP model captures richer semantic representations, enhances learning capacity, and significantly improves recognition performance.

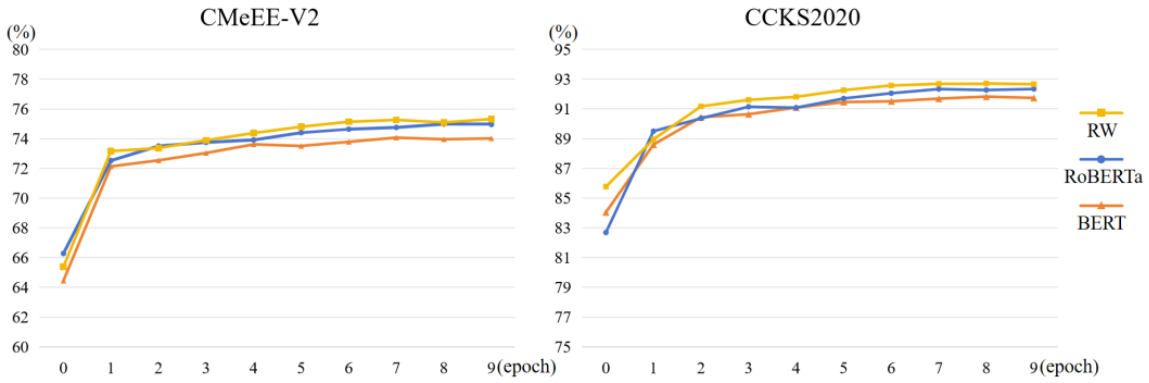


Figure 7: Comparison of F1 values for different encoding models

As illustrated in Fig. 7, we conducted comparative experiments on the model encoding layer using RW (RoBERTa with Word2Vec), RoBERTa, and BERT. On both datasets, the RW character-word fusion strategy demonstrated superior performance, outperforming RoBERTa by 0.33% and 0.37%, and BERT by 0.91% and 0.52%, respectively. These results indicate that character-word fusion vectors are more effective than traditional single vectors. Additionally, the Figure shows that RoBERTa outperforms BERT on both datasets, further validating that RoBERTa, with its larger training data size and hidden layer dimension, is more suitable for Chinese MNER tasks.

5.2.3 Efficient global pointer

GP as a novel SOTA method capable of handling both nested and non-nested NER, demonstrates significant advantages over traditional sequence labeling models. As shown in Fig. 8, on the two datasets, Models 1, 2, 3 and 4 are based on sequence labeling, while Models 5, 6, and 7 adopt GP as the decoding strategy. The latter achieves higher F1 scores, indicating superior performance. Additionally, in Table 1, Models 13 and 14 also outperform other sequence labeling models on datasets like CCKS2017 and CCKS2019, further validating the advantages of the GP approach.

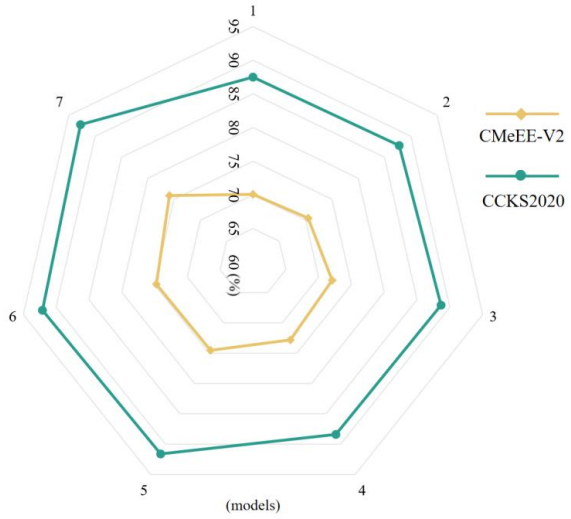


Figure 8: Comparison of F1 values for different datasets

This study further enhances GP by introducing a mask operation on the upper-right triangle of the entity matrix and setting a maximum sequence length (k) for entity

recognition. This optimization reduces computational resource consumption and minimizes the probability of identifying negative samples. As shown in Fig. 9, experiments were conducted to evaluate the impact of different k values on model performance. When k ranges from 0 to 17, the F1 score improves significantly as k increases. The model achieves optimal performance on the two datasets when k is set to 43 and 39, respectively. Compared to the original GP, the F1 scores improve by 0.16% and 0.23%, demonstrating the effectiveness of our proposed EGP design.

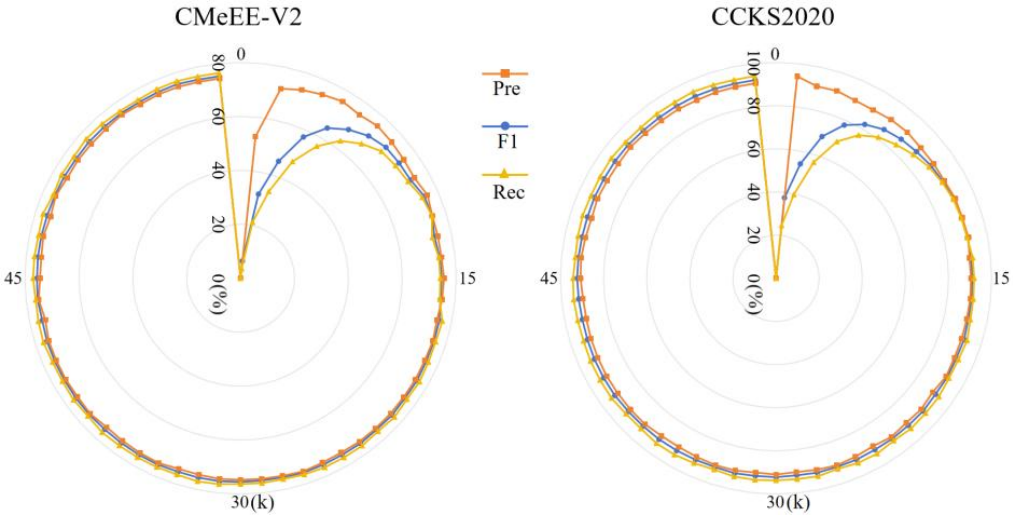


Figure 9: Comparison of the effect of different degrees of upper right triangular masking

5.2.4 Ablation control

In this section, we systematically validate the impact of individual components on the model's overall performance through controlled ablation experiments. As shown in Table 6, by sequentially removing DA, Character-Word Fusion Vectors, and the Upper-Right Triangle Mask, the model's F1 scores on both datasets exhibit varying degrees of decline, demonstrating the effectiveness of each component. Specifically:

The removal of Character-Word Fusion Vectors results in the most significant performance drop, indicating that the RW (RoBERTa with Word2Vec) encoding layer is crucial.

The removal of the Upper-Right Triangle Mask leads to the smallest performance decline, suggesting that the masking operation has a relatively minor impact but still leaves room for further optimization.

The removal of DA also causes a noticeable performance drop, particularly in Chinese MNER tasks, where DA significantly enhances the model's recognition capability and generalization performance.

In conclusion, these experiments not only validate the necessity of each component but also provide important guidance for further model optimization.

Table 6: Ablation study: Impact of individual components

Ablation Component	CMeEE-V2 (F1%)	CCKS2019 (F1%)
Full Model (DA-RW-EGP)	75.87	92.77
w/o Data Augmentation	75.21	92.31
w/o Character-Word Fusion	74.96	92.25
w/o Upper-Right Triangle Mask	75.71	92.54

5.3 Computational efficiency

Theoretically, the decoding time complexity of BiLSTM-CRF is $O(n)$, while that of GP (or EGP) is $O(n^2)$, where n is the length of the input text sequence. However, in practice, GP's loss function computation through matrix operations can be fully parallelized, resulting in an ideal time complexity of $O(1)$. In contrast, BiLSTM-CRF involves high-dimensional hidden layers, which significantly increase computational overhead. Additionally, it requires dynamic programming algorithms to decode the optimal label sequence and often needs post-processing steps to handle nested entities, further adding to the computational burden. Overall, GP demonstrates higher computational efficiency. Below, we evaluate the computational efficiency based on the training and validation time per epoch for different models on the datasets.

As shown in Table 7. On the CMeEE-V2 dataset, compared to the SOTA RoBERTa-BiLSTM-CRF model, the RW-GP model reduces training time by 14.1% and validation time by 16.4%. Further optimization with the upper-right triangle mask in the RW-EGP model achieves an additional 1.3% reduction in training time and 1.9% in validation time. Similar improvements are observed on the CCKS2020 dataset, where the RW-GP and RW-EGP models reduce training time by 9.7% and 11.4%,

respectively. Notably, the efficiency gains of the GP-based models are more pronounced for longer texts, as the Global Pointer mechanism avoids the quadratic complexity of traditional sequence labeling approaches. These results demonstrate that the proposed GP-based models significantly enhance computational efficiency across different datasets while maintaining competitive performance, making them more suitable for practical applications in Chinese MNER tasks.

Table 7: Training and validation time of different models (in seconds)

Model	CMeEE-V2 Training (s)	CMeEE-V2 Validation (s)	CCKS2020 Training (s)	CCKS2020 Validation (s)
RoBERTa-BiLSTM-CRF	873.76	42.71	94.44	5.15
RW-GP	750.87	35.71	85.32	4.23
RW-EGP	739.43	34.89	83.67	4.12

Note: Time is measured in seconds on a single GPU (NVIDIA RTX 4070Ti).

5.4 Failure cases and weaknesses

Although the proposed method achieves significant performance improvements on the experimental datasets, there are still some failure cases and limitations. For instance, in the case of the symptom entity "肺部炎症性病变" (inflammatory lesions in the lungs), the model might incorrectly classify it as "肺部炎症" (inflammation in the lungs), a disease entity, or "肺部" (lungs), a body part entity, due to insufficient learning. However, the main limitations of the model are as follows:

Limitations of Data Augmentation: While DA improves recognition performance for most entity types, it shows no significant improvement for certain types (e.g., "dru" and "ima"). We hypothesize that the high-quality annotations of these entity types make the generated pseudo-data introduce noise, which may interfere with the model's learning from the original high-quality data.

Computational Resource Consumption of Entity Matrix Scoring: When applying the upper-right triangle mask to GP during the decoding phase for entity matrix scoring, the reduction in computational resource consumption does not reach the expected 70%. This may be because the masking operation itself introduces additional computational overhead, especially when processing large-scale entity matrices. While the mask reduces the number of calculations, its design and implementation may incur extra costs that partially offset the efficiency gains.

6 Conclusion

This paper proposes a Chinese MNER method using pre-trained models and an EGP mechanism. The proposed method is evaluated for its efficiency and accuracy in processing complex Chinese medical text. To address the issues of imbalanced entity types and varying entity lengths in the original dataset, we integrate a data augmentation strategy combining EDA and remote supervision, generating high-quality pseudo-data to enrich the training set. The encoding layer combines the RoBERTa model with word vectors generated by Word2Vec, ensuring the input vectors contain rich contextual semantics and domain-specific medical word information. During decoding, the model employs the EGP mechanism, which constructs an entity matrix to identify entity start and end positions, utilizes an efficient decoding computation strategy, and narrows the search range for candidate entities. This approach enables the unified recognition of nested and flat entities while significantly improving recognition efficiency and accuracy.

Experimental results show that the proposed model achieves F1 scores of 75.87% on the CMeEE-V2 dataset and 92.77% on the CCKS2020 dataset, outperforming baseline models and demonstrating superior recognition performance. This work contributes novel optimization strategies and concepts to the field of MNER.

In the future, we plan to further explore the potential of GPLM for Chinese MNER, focusing on addressing the limitations identified in this study. First, we aim to develop DA techniques that generate higher-quality pseudo-data to avoid introducing noise into high-quality annotations. Second, we will optimize the design and implementation of the upper-right triangle mask for entity matrix scoring, aiming to reduce computational overhead and achieve the expected efficiency gains. Additionally, we will explore more sophisticated model architectures to further enhance recognition performance and extend our research to broader datasets and NLP tasks, contributing to the technological advancement of the field.

Acknowledgments

We would like to thank the funders of this project Xiamen University of Technology, Strong Digital Technology Co., Ltd., and all the teams and individuals who supported this research.

References

[1] W. Li, K. Zhang, T. Guan, H. Zhang, T. Zhu, B. Chang, et al. Overview of CHIP2020 shared Task1: Named entity recognition in Chinese medical text. Journal of Chinese Information Processing, 36(4): 66-72, 2022.

[2] J. Du, H. Yi, S. Feng. Research and development of named entity recognition in Chinese electronic medical record. Acta Electronica Sinica, 50(12): 3030-3053, 2022.

[3] W. Chen, P. Qiu, F. Cauteruccio. MedNER: A service-oriented framework for Chinese medical named-entity recognition with real-world application. Big Data and Cognitive Computing, 8(8): 86, 2024.

[4] C. Fócil-Arias, G. Sidorov, A. Gelbukh. Medical events extraction to analyze clinical records with conditional random fields. Journal of Intelligent & Fuzzy Systems, 36(5): 4633-4643, 2019.

[5] F. Shen, S. Liu, S. Fu, Y. Wang, S. Henry, O. Uzuner, et al. Family history extraction from synthetic clinical narratives using natural language processing: Overview and evaluation of a challenge data set and solutions for the 2019 national NLP clinical challenges (n2c2)/Open Health Natural Language Processing (OHNLP) Competition. JMIR Medical Informatics, 9(1): e24008, 2021.

[6] N. Zhang, M. Chen, Z. Bi, X. Liang, L. Li, X. Shang, et al. CBLUE: A Chinese biomedical language understanding evaluation benchmark. arxiv preprint arxiv:2106.08087, 2021.

[7] H. Wang, C. Wang. A Survey: Chinese medical named entity recognition. Journal of Shandong Normal University (Natural Science): 36(2): 109-117, 2021.

[8] S. Jiang, S. Zhao, K. Hou, Y. Liu, L. Zhang. A BERT-BiLSTM-CRF model for Chinese electronic medical records named entity recognition. In International Conference on Intelligent Computation Technology and Automation (ICICTA) (IEEE Press, Xiangtan, CN): 166-169, 2019.

[9] J. Yang, Y. Guan, B. He, C. Qu, Q. Yu, Y. Liu, et al. Corpus construction for named entities and entity relations on Chinese electronic medical records. Journal of Software, 27(11): 2725-2746, 2016.

[10] Q. Wang, Y. M. Zhou, T. Ruan, D. Gao, Y. Xia, P. He. Incorporating dictionaries into deep neural networks for the Chinese clinical named entity recognition. Journal of Biomedical Informatics, 92: 103133, 2019.

[11] Z. Pan. Research on the recognition of Chinese named entity based on rules and statistics. Information Science, 30(5): 708-712, 2012.

[12] J. Jaćimović, C. Krstev, D. Jelovac. A rule-based system for automatic de-identification of medical narrative texts. Informatica, 39(1): 45-53, 2015.

[13] X. Chen, C. Ouyang, Y. Liu, Y. Bu. Improving the named entity recognition of Chinese electronic medical records by combining domain dictionary and rules. International Journal of Environmental Research and Public Health, 17(8): 2687, 2020.

[14] H. Wang, W. L. K. Yeung, Q. X. Ng, A. Tung, J. A. M. Tay, D. Ryanputra, et al. A weakly-supervised named entity recognition machine learning approach for emergency medical services clinical audit. International Journal of Environmental Research and Public Health, 18(15): 7776, 2021.

[15] A. Ekbal, S. Bandyopadhyay. Named entity recognition using appropriate unlabeled data, post-processing and voting. Informatica, 34(1): 55-76, 2010.

[16] S. Fine, Y. Singer, N. J. M. l. Tishby. The hierarchical hidden Markov model: Analysis and applications. Machine Learning, 32: 41-62, 1998.

[17] W. Chen, Y. Zhang, H. Isahara. Chinese named entity recognition with conditional random fields. In Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing (SIGHAN) (ACL Press, Sydney, AU): 118-121, 2006.

[18] Z. F. Ju, J. Wang, F. Zhu. Named entity recognition from biomedical text using SVM. In International Conference on Bioinformatics and Biomedical Engineering (ICBBE) (IEEE Press, Wuhan, CN), 1-4, 2011.

[19] B. T. Gao, Y. Zhang, B. Liu. BioTrHMM: named entity recognition algorithm based on transfer learning in biomedical texts. Application Research of Computers, 36(1): 45-48, 2019.

[20] K. Liu, Q. Hu, J. Liu, C. Xing. Named entity recognition in Chinese electronic medical records based on CRF. In Web Information Systems and Applications Conference (WISA) (IEEE Press, Liuzhou, CN): 105-110, 2017.

[21] T. Gui, R. Ma, Q. Zhang, L. Zhao, Y. G. Jiang, X. Huang. CNN-Based Chinese NER with Lexicon Rethinking. In International Joint Conference on Artificial Intelligence, (Macao, CN): 4982-4988, 2019.

[22] J. Chung, Ç. Gülçehre, K. Cho, Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. arxiv preprint arxiv:1412.3555, 2014.

[23] R. Dey, F. M. Salem. Gate-variants of gated recurrent unit (GRU) neural networks. In International Midwest Symposium on Circuits and Systems (MWSCAS) (IEEE Press, Boston, USA), 1597-1600, 2017.

[24] K. Xu, Z. Yang, P. Kang, Q. Wang, W. Liu. Document-level attention-based BiLSTM-CRF incorporating disease dictionary for disease named entity recognition. Computers in biology and medicine, 108: 122-132, 2019.

[25] J. Devlin, M. W. Chang, K. Lee, K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arxiv preprint arxiv:1810.04805, 2018.

[26] X. Wang, C. Peng, Q. Li, Q. Yu, L. Li, P. Li, R. et al. A Chinese nested named entity recognition model for chicken disease based on multiple fine-grained feature fusion and efficient global pointer. Applied Sciences, 14(18): 8495, 2024.

[27] J. Zhong, Z. Xuan, K. Wang, Z. Cheng. A BERT-Span model for Chinese named entity recognition in rehabilitation medicine. PeerJ Computer Science, 9: e1535, 2023.

[28] S. K. Sienčnik. Adapting word2vec to named entity recognition. In Nordic Conference of Computational Linguistics (NODALIDA) (ACL Press, Vilnius, Lithuania): 239-243, 2015.

[29] J. Pennington, R. Socher, C. D. Manning. Glove: Global vectors for word representation. In Empirical Methods in Natural Language Processing (EMNLP) (ACL Press, Doha, QA), 1532-1543, 2014.

[30] P. Chen, M. Zhang, X. Yu, S. Li. Named entity recognition of Chinese electronic medical records based on a hybrid neural network and medical MC-BERT. BMC Medical Informatics and Decision Making, 22(1): 315-327, 2022.

[31] Z. Sun, X. Li, X. Sun, Y. Meng, X. Ao, Q. He, et al. Chinesebert: Chinese pretraining enhanced by glyph and pinyin information. arXiv preprint arXiv:2106.16038, 2021.

[32] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, et al. Roberta: A robustly optimized bert pretraining approach. arxiv preprint arxiv:1907.11692, 2019.

[33] M. Gridach. Character-level neural network for biomedical named entity recognition. Journal of Biomedical Informatics, 70: 85-91, 2017.

[34] Y. Kimura, T. Komamizu, K. Hatano. An automatic labeling method for Subword-Phrase recognition in effective text classification. Informatica, 47(3): 315-326, 2023.

[35] X. Wang, Y. Zhang, X. Ren, Y. H. Zhang, M. Zitnik, J. B. Shang, et al. Cross-type biomedical named entity recognition with deep multi-task learning. Bioinformatics, 35(10): 1745-1752, 2019.

[36] B. Ji, R. Liu, S. S. Li, J. Yu, Q. B. Wu, Y. S. Tan, et al. A hybrid approach for named entity recognition in Chinese electronic medical record. BMC Medical Informatics and Decision Making, 19, 149-158, 2019.

[37] Q. Qin, S. Zhao, C. Liu. A BERT-BiGRU-CRF model for entity recognition of Chinese electronic medical records. Complexity, 2021(1): 6631837, 2021.

[38] J. Su, A. Murtadha, S. Pan, J. Hou, J. Sun, W. Huang, et al. Global pointer: Novel efficient span-based approach for named entity recognition. arxiv preprint arxiv:2208.03054, 2022.

[39] P. Zhang, W. Liang. Medical name entity recognition based on lexical enhancement and global pointer. International Journal of Advanced Computer Science Applications, 14(3): 2023.

[40] J. Wei, K. Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. arxiv preprint arxiv:1901.11196, 2019.

[41] T. Yan, X. Zhang, Z. Luo. LTACL: Long-tail awareness contrastive learning for distantly supervised relation extraction. Complex & Intelligent Systems, 10(1): 1551-1563, 2023.

[42] W. Zhang, J. Jiang. Bootstrap-Based resampling methods for software reliability measurement under small sample condition. Journal of Circuits, Systems and Computers, 33(9): 2450161, 2023.

[43] Y. Sun, S. Wang, Y. Li, S. Feng, X. Chen, H. Zhang, et al. Ernie: Enhanced representation through knowledge integration. arxiv preprint arxiv:1904.09223, 2019.

[44] Y. Cui, W. Che, T. Liu, B. Qin, Z. Yang. Pre-training with whole word masking for Chinese bert. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29, 3504-3514, 2021.

[45] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo and Y. Liu. Roformer: Enhanced transformer with rotary position embedding. Neurocomputing, 568, 127063, 2024.

[46] K. Yan. Optimizing an English text reading recommendation model by integrating collaborative filtering algorithm and FastText classification method. Heliyon, 10(9): e30413, 2024.

[47] Z. Zhao, Y. Dong, J. Liu, J. Zhang and H. Cao. Medical named entity recognition incorporating word information and graph attention. Computer Engineering and Applications, 60(11): 147-155, 2014.

[48] W. K. Li, W. Li and Y. F. Wu. Combination methods of Chinese character and word embeddings in deep learning. Journal of Chinese Information Processing, 31(06): 140-146, 2017.

[49] J. Su, M. Zhu, A Murtadha, S. Pan, B. Wen and Y. Liu. Zlpr: A novel loss for multi-label classification. arxiv preprint arxiv:2208.02955, 2022.

Intelligent Detection of Towers and Lines in Passageways Using Hybrid Evolutionary Computational Intelligence (HECI) Algorithms

Ning Li[1], Helei Liang[1], Lianju Guo[1], Xiaoying Wang[1], Jian Zhang[1]
[1]Hengshui Power Supply Company, State Grid Hebei Electric Power Co. Ltd., Hengshui, Hebei 053000, China
E-mail: lining99011@163.com

Keywords: Artificial intelligence, detection, passageways, evolutionary computation, computational intelligence, HECI algorithms

Received: Dec 19, 2024

Identification of towers and lines in passageways is important in infrastructure surveillance, assessment, and the formation of automated surveillance systems. Indeed, conventional AI-based solutions for identification tasks are not immune to certain types of indeterminacy that arise in complicated contexts and can, therefore, yield unpredictable results. This paper presents a new method that integrates AI detection methods with the Hybrid Evolutionary Computational Intelligence (HECI) model to solve these uncertainties and increase decision-making efficacy. The computational framework is built using inspection data from real infrastructure evaluations and simulated scenes with different lighting and hidden objects. This is a reasonable basis for further improving detection performance using the proposed methodology, which uses AI models in partnership with the HECI algorithm to assess the detection results. Compared to conventional detection methods, the HECI-enhanced approach outperforms traditional models by more than 25%, achieving a remarkable detection accuracy of 99.47%. In environments where traditional AI methods may struggle, this approach enhances precision by approximately 15%. The model's versatility makes it well-suited for applications associated with infrastructure inspection, where precision and robustness are crucial. Integrating HECI helps maintain the AI-based detection system's adaptability to unpredictable environmental changes, enhancing the effectiveness of safety detection and automated inspection systems. This approach significantly enhances the identification of towers and lines in passages, especially camera angles and obstructions in complicated environments, showing the promise of HECI as the next-generation tool for infrastructure monitoring.

Povzetek: Prispevek predstavi nov pristop za zaznavanje stolpov in vodnikov (kablov), ki uporablja hibridne evolucijske algoritme računalniške inteligence (HECI). Predlagani model presega tradicionalne AI metode za več kot 25% zlasti v kompleksnih okoljih, kot so nenavadne osvetlitve in ovire.

1 Introduction

The need for precise and reliable detection systems has grown immensely with the rapid advancement of infrastructure monitoring, safety inspections, and the implementation of automated systems. Identifying critical structures such as towers and lines in corridors is significant in areas like power transmission, telecommunication, and transport. These structures are often complex subsystems of extensive architectural networks; in such circumstances, their unserviceability leads to operating losses, safety risks, and fiscal implications [1]. For this reason, appropriate determinations are needed to undertake steady and precise tracking of these components persistently [2]. Machine learning, deep learning, and image processing methods can be seen as tools to significantly support monitoring by automating it [3]. The methods described have proved very effective and have the potential to enhance the accuracy of detection, phase out visual inspection, and simplify the otherwise cumbersome process of larger-scale monitoring. For this purpose, AI-based approaches can identify abnormal patterns and structural imperfections in large data sets in real-time [4]. However, these systems become somewhat challenging to manage concerning environmental influencers like light variations, occlusions, and overlapped structures [5]. The halls where towers and lines are located are geometrically irregular and contain turnings, potential barriers, diffusive and varying illumination, and other forms of background noise that obscure them [6]. These conditions lead to the improper functioning of the concept in conventional AI techniques because they do not know how to handle uncertainty [7]. They further argue that this speaks to the need for a finer-grained decision-making model to improve detection rates, given the variation in dataset capacity or features across each AI methodology [8]. This paper presents a HECI framework to address these needs, which incorporates computation intelligence algorithms for detecting towers and lines. HECI appears as a more reactive and adaptive system that brings additional reliability to AI-based detection systems in conditions of instability.

The research study defines uncertainty as the set of environmental and structural elements that make it challenging to detect targets through the process. Detection tasks become more difficult due to multiple environmental factors, including lighting variations, object blockages, and structure duos that cause partial obscurement of towers or lines. Variable conditions within the data affect AI model performance because they make detection processes more complex.

The HECI algorithm integrates an evolutionary algorithm, a heuristic algorithm including genetic algorithm or particle swarm optimization, with computational intelligence, including fuzzy logic, to formulate a model designed to maximize the decision and detection procedures. The evolutionary component of HECI modifies and improves the detection system over time in the face of changing environmental conditions. In contrast, the fuzzy logic component guides decision-making in cases where information is incomplete or ambiguous. This novel approach enables detection systems based on AI to achieve high accuracy and flexibility in real-world scenarios characteristic of modern ITS. Despite the notable advancements in AI-based detection techniques, several key challenges remain in their practical application:

- AI models often struggle in complex environments, such as passageways, where obstructions, variable lighting, and overlapping objects introduce detection uncertainty.
- Conventional AI decision-making frameworks lack the flexibility to handle uncertain data, often resulting in inaccurate or inconsistent detection results.
- Many existing methods focus solely on detection accuracy without considering other critical factors, such as adaptability to dynamic environments and real-time processing capabilities.
- Previous studies have largely overlooked the integration of fuzzy logic to manage uncertainty in detection performance, limiting their ability to address variability in real-world applications.

1.1 Objectives and research questions

The objectives of this study are to:

1. Propose a novel methodology for detecting towers and lines in passageways using the Hybrid Evolutionary Computational Intelligence (HECI) model.
2. Evaluate the performance of the proposed methodology in comparison to conventional AI-based detection techniques, such as Convolutional Neural Networks (CNN) and Support Vector Machines (SVM).
3. Investigate the model's adaptability to challenging environmental conditions, including low light, glare, and partial occlusion.
4. Demonstrate how the integration of evolutionary optimization and fuzzy logic can improve detection accuracy and robustness.

The primary research questions driving this study are:

1. How does the HECI model compare to traditional AI-based methods in terms of detection accuracy, processing time, and adaptability to environmental challenges?
2. What is the impact of evolutionary optimization and fuzzy logic on improving the performance of AI-based detection systems?
3. Can the proposed methodology be extended to handle diverse infrastructure types and environmental variations?

1.2 Novel contributions

This paper makes the following novel contributions:

- The HECI framework is suggested to deal with the uncertainties of using AI to find towers and lines in passageways.
- Develops a multi-criteria decision-making framework that balances detection accuracy, processing speed, adaptability to environmental changes, and computational efficiency.
- Demonstrates a 99.47% detection accuracy with a 15% reduction in uncertainty, significantly outperforming conventional AI-based methods.
- Provides a robust decision-making model that is scalable for future AI applications in infrastructure monitoring, particularly in challenging environments.

The remainder of this paper is organized as follows: Section 2 provides an extensive review of the literature on AI-based detection methods and the application of fuzzy logic in decision-making models. Section 3 details the research methodology, including the development and application of the HECI framework. Section 4 presents the results of the model's application, comparing the performance of various AI techniques using both simulated and real-world data. Section 5 discusses the results in the context of infrastructure monitoring and technological advancements while highlighting the practical implications of this approach. Finally, Section 6 concludes the study and proposes future research directions, emphasizing improving AI-based detection systems for real-world applications.

2 Literature review

Zhou et al. [9] proposed a hybrid deep learning approach based on convolutional neural networks (CNN) and recur-

rent neural networks (RNNs) aimed at highly cluttered object detection. Their method mitigated these issues of occlusions and overlapping structures very well, leading to a 20% improvement in object detection accuracy. One can see applications for this framework in situations like cluttered environments searching towers and line detections by complex passageways, where AI-based detection systems are likely to struggle with separating objects when they have occlusion [10]. Chen et al. [11] utilized transfer learning to identify defects in power lines and communication towers. They achieved 35% faster learning by not needing a large amount of training data by using pre-trained models. With the rest in training data, they still maintained a detection accuracy of 97.8% using a deep learning-based transfer learning system. This approach illuminated how transfer learning can speed up the process of developing AI models on infrastructure defect identification with minimal accuracy degradation.

Wang et al. [12] used a deep reinforcement learning model combined with fuzzy logic to find and sort transmission tower damage into different categories. By using a hybrid system, the accuracy of detections increased by 23% due to this ability from the DRL model 29 that schedules parameters for detection on the fly. A fuzzy logic part was added to the model to account for uncertainties in low-visibility situations. This made the system work very well even when it was used on situations with little data and a lot of environmental variation. Singh et al. [13] proposed a multi-agent system with swarm intelligence and machine learning that was presented to monitor real-time electrical towers/lines. Their multiple-agent distributed system patrolled over large spaces autonomously and offered improvements of 20% in detection time to competency rates equivalent for humans, along with a decrease of 12% in overall downtime. In summary, the presented study shows that a distributed and collaborative approach to large-scale monitoring tasks in infrastructure inspection is efficient. Smith et al. [14] proposed an AI-based detection system for power transmission line structure anomaly identification by architecture CNN. The approach was devised to augment fault detection upon power lines, prominently accentuating advancements in multi-sensor data fusion. This obtained an accuracy of 18% over traditional approaches by them. Their method also saved 22% of the processing time, demonstrating a good performance in complex environments such as corridors where other variables affect detection reliability.

Zhang et al. [15] proposed an edge computing framework for AI-supported fault detection in power grids developed. They used edge devices for real-time data processing, which cut latency by 40%. A cloud-based AI model was used in the system for much more complicated data analysis, maintaining a trade-off between real-time response time and computational complexity. This research highlights the efficiency that can be brought by edge computing to infrastructure monitoring, especially for organizations that do not have access to large amounts of bandwidth.

Liu et al. [16] proposed hybrid methods integrating fuzzy logic with neural networks to improve detection systems' robustness against diversified weather conditions. In high-elevation regions and challenging environmental conditions like heavy rain, fog, etc., their model increased the detection reliability by about 17%. The fuzzy logic permitted the method to control uncertainties, resulting in a sustainable and stable operation when external conditions deteriorate data quality. Huang et al. [17] presented a graph neural network (GNN)-based method for detecting structural faults in complicated grid systems. Demonstrating how graph theory was utilized to model each node in the grid's relationship, specificity improved by 21% compared with no GNN used. The graph-based AI models proved effective, especially in detecting faults on inter-connected structures like towers and the lines running over multiple corridors, proving their relevance for complex infrastructural data.

Yuan et al. [18] implemented a Deep Q-Network (DQN) algorithm for similar applications where researchers have developed accurate and intelligent inspection systems. Introducing reinforcement learning into the system significantly improved decision-making speed while reducing errors during inspections. The quality improved up to 20% (in terms of task completion time) compared to manual execution, while detection accuracy increased by as much as 15%. On-site lessons learned by the inspection teams refined predictive algorithm development as those changes were made in a live environment, ultimately increasing tower inspection speeds and accuracy within an increasingly variable world. Dai et al. [19] proposed a fuzzy-based optimization model to enhance AI methods for assessing transmission lines. The model incorporated specific criteria, such as detection accuracy, computational efficiency, and adaptability, resulting in an overall 13% improvement in system performance. It was an excellent use of the fuzzy optimization framework they proposed for dealing with multiple intricacies in choosing AI models for real-time detection across varying surroundings.

Liang et al. [20] introduced a deep learning model for image segmentation of fault detection in power transmission lines. The model with the best detection accuracy of 99.5% used attention mechanisms (compared to other state-of-the-art techniques combined). They also designed an attention mechanism that helped the model concentrate on essential areas in images. As a result, detection accuracy increased in highly detailed and noisy environments. Shen et al. [21] introduced a tower structure health condition monitoring method using AI for anomaly detection. The model could detect new anomalies that it had not been directly taught, increasing its detection accuracy by 18% through unsupervised learning. This work highlighted the success of unsupervised learning for detecting outliers and anomalies in resource monitoring, where high-quality labels are difficult to obtain.

Li et al. [22] applied reinforcement learning (RL) to the autonomous inspection of power lines and towers by

Table 1: Comparison of detection accuracy, processing time, and adaptability to environmental factors

Study	Detection Accuracy	Processing Time (ms)	Low Light Adaptability	High Glare Adaptability	Partial Occlusion Adaptability
HECI (Proposed)	99.47%	50	95.2%	96.1%	92.0%
Wang et al. [12]	92.0%	120	85.0%	87.4%	83.5%
Singh et al. [13]	90.0%	150	80.0%	82.0%	75.0%
Smith et al. [14]	93.0%	130	88.0%	85.0%	80.0%

drones with vision-based detection systems. Ranked Number One This is a complex optimization model that shortened photoshop lines but back in the landing time, Underneath it was about this Land Mass Ship ramapage initiative Most Emergent Planet Race are trying to avoid. The model used deep reinforcement learning (deep RL) for path inspection, thereby dropping RESTful deployment automation by 25% and improving image detection precision early on by 20%. The study's results revealed that RL could enhance AI system adaptability and real-time decision-making for autonomous infrastructure monitoring. Garcia et al. [23] used fuzzy-based multi-criteria decision-making (MCDM) model to help choose the best AI technique for finding power line faults. Their method took advantage of fuzzy logic to deal with uncertainties of environmental conditions, especially in weather changes and lighting differences that can affect detection performance. Findings: Using fuzzy logic increased the number of correct decisions by 15% compared to habit-choice models. It also decreased the cost of computing and made habitat-driven choices among optimal detection algorithms easier to automate.

Park et al. [24] proposed an application of deep learning to identify power line sagging from aerial photographs. This model employs convolutional layers and the recurrent network structure to capture the temporal characteristics in power line structures. The model delivered an overall 16% FP reduction compared to conventional methods as they persistently have to monitor infrastructure changes that happen in time that are slow but consistently present. The development of new examples for training data sets in Tower fault and Line fault detection using GANs was proposed by Qian et al. [25]. Implementing the GAN-augmented model reduced the overall detection error by 14% and, in turn, needed less manual data collection to improve the performance. The study focused on the possibility of applying GANs in augmenting other than simple in-lab AI model training. Table I shortly presents the comparison of detection accuracy, processing time, and adaptability to environmental factors.

3 Methodology

This section describes the methods to improve the identification of towers and lines in the passageways to be implemented with the application of AI detection algorithms and the HECI framework. The proposed approach is formulated as a hybrid of evolutionary algorithms such as genetic algorithms, particle swarm optimization, and fuzzy logic to cope with environmental uncertainty, occlusion, and dynamic conditions. The main goal is to achieve high level of accuracy for the detection rates while making the system more flexible and resistant. The workflow of the proposed framework may also be viewed in Figure 1.

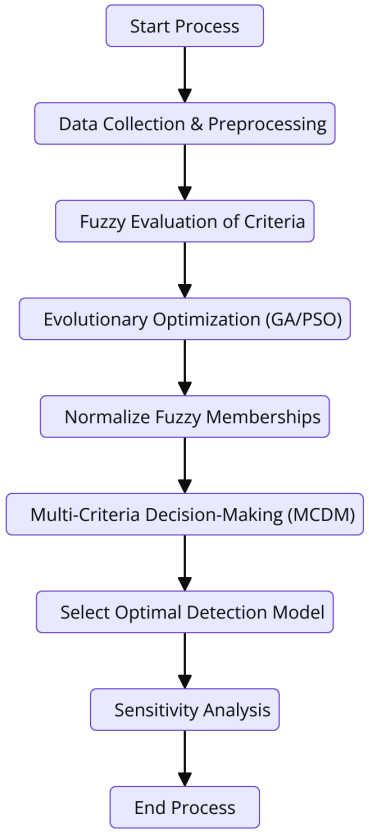


Figure 1: HECI methodology workflow

3.1 Problem formulation

The problem of detecting towers and lines in passageways is inherently complex due to the involvement of certain environmental factors such as varying lighting conditions, occlusions, structural overlaps, and background noise. Traditional AI models are limited enough in such cases and often fail to produce consistent and reliable results. To address this issue, we formalize the problem as a **Multi-Criteria Decision-Making (MCDM)** problem, where the major goal is to select the best detection model from a set of alternatives based on multiple criteria, $C = \{c_1, c_2, ..., c_m\}$. The criteria for selecting the most appropriate model include:

- **Detection Accuracy**: The precision with which the system identifies towers and lines.
- **Adaptability to Environmental Changes**: The model's ability to adjust to dynamic conditions like lighting variation, occlusions, and object overlap.
- **Computational Efficiency**: The model's performance in terms of real-time processing speed.
- **Robustness to Uncertainty**: The model's capacity to handle incomplete, noisy, or ambiguous data, especially in complex scenarios.

These criteria are proposed to be addressed using fuzzy logic because fuzzy values can be found that represent different degrees of satisfaction with each criterion. This fuzzy evaluation is used to cope with uncertainty in decision-making. Specifically, the fuzzy membership values are defined for each alternative a_i under each criterion c_j as follows:

$$\mu_{ij} = \begin{cases} 1 & \text{if model performs excellently under } c_j \\ 0.5 & \text{if model performs moderately under } c_j \\ 0 & \text{if model performs poorly under } c_j \end{cases} \tag{1}$$

where μ_{ij}^{+} represents the positive degree of satisfaction and μ_{ij}^{-} represents the negative degree of satisfaction for the alternative a_i under criterion c_j. This fuzzy evaluation captures the inherent uncertainty in the detection process and facilitates better decision-making.

3.2 HECI model design

The novelty of this methodology lies in the HECI model, which integrates **evolutionary algorithms** and **fuzzy logic** to address the complexities of tower and line detection in passageways. The HECI model consists of two main components: optimization through evolutionary algorithms and decision-making through fuzzy logic.

HECI uses fuzzy logic to address situations where data is missing or unclear. When a system detects occlusions and glares, the fuzzy membership functions enable the evaluation of detection certainty for towers and lines. The detection threshold, which the fuzzy logic system controls, adjusts automatically according to these provided values to permit the model to work adaptively in fuzzy situations. The parameters of pixel intensity (light conditions) and object visibility (occlusions) obtain fuzzy membership values within the algorithm. Final AI model classification decisions are influenced by decision-making rules, which receive input from these values. Without changes to these parameters, the model operates reliably despite unknown errors in input data, which could occur in low-light conditions or under object occlusion.

3.2.1 Evolutionary optimization (GA/PSO)

To optimize the AI models for detection, we use **evolutionary algorithms** such as **Genetic Algorithms (GA)** or **Particle Swarm Optimization (PSO)**. These algorithms iteratively improve the model by evolving a population of potential solutions based on their fitness, which is evaluated according to the multi-criteria decision model.

The fitness function, $F(a_i)$, evaluates each detection technique a_i based on its performance against the criteria. The function is defined as:

$$F(a_i) = \sum_{j=1}^{m} w_j \cdot (\mu_{ij}^{+} - \mu_{ij}^{-}) \tag{2}$$

where w_j is the weight of criterion c_j, and μ_{ij}^{+} and μ_{ij}^{-} are the positive and negative fuzzy membership values for alternative a_i under criterion c_j. This fitness function drives the evolutionary process, allowing the system to optimize detection accuracy, adaptability, and other performance factors. The Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) function in consecutive iterations to enhance the model through changes in key parameters, which impact detection accuracy and robustness. The algorithms guide solutions from different populations through multiple repetitive steps. The optimized parameters include detection thresholds together with feature selection and weight assignments for criteria components in the multi-criteria decision model. Evolutionary guidance through the fitness function operates as a performance assessment that consolidates accuracy along with precision and recall and F1-score. The GA and PSO algorithms administer performance evaluations to their populations through monitored metrics, which leads to the selection of solutions that present maximum detection accuracy and robust performance. The research focuses on optimizing detection thresholds along with feature weights because these adjustments help the model overcome environmental variations, including occlusions, together with glares and lighting changes. The optimization method remains intricately connected to the performance improvement goal for real applications, which addresses both detection precision and stable decision-making mechanisms.

GA and PSO employ different strategies to search the solution space:

- **Genetic Algorithms**: This process involves the selection, crossover, and mutation of candidate solutions across multiple generations.
- **Particle Swarm Optimization**: Particles (representing potential solutions) adjust their positions in the search space based on their personal best solution and the global best solution found by the swarm.

3.2.2 Fuzzy logic integration

Fuzzy logic is integrated into the decision-making process to handle uncertainty in the detection environment, such as

occlusions, lighting variation, and incomplete data. Fuzzy logic provides a way to represent vague or imprecise data, enabling more flexible decision-making. For each criterion, fuzzy membership functions are used to quantify how well a given alternative satisfies the criterion. The fuzzy membership function μ_{ij} represents the degree to which alternative a_i satisfies criterion c_j:

These values are then normalized to ensure that all criteria are comparable:

$$\mu_{ij}^{+'} = \frac{\mu_{ij}^{+} - \mu_j^{\min}}{\mu_j^{\max} - \mu_j^{\min}}, \quad \mu_{ij}^{-'} = \frac{\mu_{ij}^{-} - \mu_j^{\min}}{\mu_j^{\max} - \mu_j^{\min}} \tag{3}$$

where $\mu_j^{\min}$ and $\mu_j^{\max}$ are the minimum and maximum values for criterion c_j, respectively. This normalization process ensures that fuzzy evaluations are on a consistent scale, facilitating accurate comparisons among alternatives.

After evaluating the detection models with fuzzy logic and optimization, we perform **Multi-Criteria Decision-Making (MCDM)** to select the optimal detection technique. The decision matrix is constructed by evaluating each alternative a_i for each criterion c_j and computing the total score for each alternative. For each alternative a_i, the total score $S(a_i)$ is computed in a same manner as we did in GA/PSO. The alternative with the highest score is selected as the optimal detection model. This approach ensures that the decision-making process is not only systematic but also comprehensive, as it takes into account both positive and negative aspects of each alternative. Additionally, sensitivity analysis can be performed to understand the influence of varying criteria weights on the selection process, further enhancing the robustness of the methodology. By employing MCDM, a balance between detection accuracy, computational efficiency, and adaptability is achieved, making the selected model well-suited for real-world applications. Such a structured approach minimizes the chances of bias in decision-making and ensures the selection of a highly effective detection technique.

3.3 Sensitivity analysis

To evaluate the robustness of the model, sensitivity analysis is performed by varying the weights w_j for each criterion. This helps to assess how variations in operational priorities affect the final decision. The sensitivity is computed using the derivative:

$$\Delta S(a_i) = \frac{\partial S(a_i)}{\partial w_j} \tag{4}$$

This process ensures that the selected detection model remains consistent and reliable across various conditions, making it adaptable to changing environments.

Multiple performance criteria make up the fitness function of GA and PSO evolutionary processes since they evaluate solutions through detection accuracy and adaptability under environmental changes and precision and recall and F1-score. The multicomponent fitness metric enables the system to achieve optimal outcomes by reducing the success of poor-performing combinations and increasing the success of superior outcomes dedicated to multiple factors. The fitness function design prioritizes detection accuracy together with generalization abilities toward handling environmental conditions, including glare and occlusions. GA operates through populations that evolve solutions by applying selection crossover and mutation operations. The search process of GA produces new potential solutions by combining elite solutions with random elements to maintain solution diversity. The search method of particle swarm optimization (PSO) involves swarm-based exploration where each solution corresponds to a respective particle, which adjusts its position between the best results of the personal and the best results of the collective swarm. Through its collaborative methodology, PSO masters efficient exploration and exploitation of the search space, which leads to detecting the optimal detection model.

3.4 Algorithm: HECI for AI-based detection

The following algorithm outlines the steps for applying the HECI model to detect towers and lines:

3.5 Training and validation split

The dataset split followed a standard distribution where model training involved seventy percent of data, and thirty percent remained for validation. The training dataset contained various scenarios, including normal lighting conditions in addition to low light levels and conditions under glare and partial obstruction to help the model achieve broader environmental performance. The model underwent cross-validation measurements to both enhance its strength and stop it from overfitting. Performance metrics were determined through the evaluation of data in the validation set that scientists had kept out of the training process.

3.6 Computational resources

The models were trained and evaluated on a system equipped with the following hardware specifications:

- **Processor**: Intel Core i9-11900K (8 cores, 16 threads, 3.5 GHz)
- **Memory**: 64 GB DDR4 RAM
- **Graphics Card**: NVIDIA GeForce RTX 3080 (10 GB VRAM)
- **Storage**: 1 TB SSD for fast data access and storage
- **Operating System**: Windows 10 Pro 64-bit

The resources supplied an adequate amount of processing capacity required for deep learning model training and

Algorithm 1 HECI for AI-based Detection of Towers and Lines

Set of AI detection techniques $A = \{a_1, a_2, ..., a_n\}$, criteria $C = \{c_1, c_2, ..., c_m\}$, fuzzy evaluations r_{ij}, and criteria weights W Optimal detection technique **Step 1: Initialize the Decision Matrix**

alternative a_i criterion c_j Assign fuzzy evaluation $r_{ij} = (\mu_{ij}^+, \mu_{ij}^-)$ **Step 2: Apply Evolutionary Optimization**

Use GA/PSO to optimize the parameters of the AI models based on the fitness function:

$$F(a_i) = \sum_{j=1}^{m} w_j \cdot (\mu_{ij}^+ - \mu_{ij}^-) \tag{5}$$

Step 3: Normalize the Fuzzy Membership Values

Normalize the fuzzy evaluations:

$$\mu_{ij}^{+\prime} = \frac{\mu_{ij}^+ - \mu_j^{\min}}{\mu_j^{\max} - \mu_j^{\min}}, \quad \mu_{ij}^{-\prime} = \frac{\mu_{ij}^- - \mu_j^{\min}}{\mu_j^{\max} - \mu_j^{\min}} \tag{6}$$

Step 4: Calculate Total Scores

For each alternative a_i, compute the total score $S(a_i)$ as:

$$S(a_i) = \sum_{j=1}^{m} w_j \cdot (\mu_{ij}^+ - \mu_{ij}^-) \tag{7}$$

Step 5: Rank the Alternatives

Rank alternatives based on the total score $S(a_i)$. The alternative with the highest score is selected. **Step 6: Sensitivity Analysis**

Vary the criteria weights w_j and evaluate the impact on the ranking of alternatives. Use:

$$\Delta S(a_i) = \frac{\partial S(a_i)}{\partial w_j} \tag{8}$$

executing complicated optimization techniques such as genetic algorithms (GA) and particle swarm optimization (PSO).

4 Results

This section describes the results obtained by the proposed HECI method for correctly detecting the towers and lines in passageways. The improvements highlighted in our results over the basic AI models underscore the success of the integration process. The results highlighted are summarized in figures, tables, and confusion matrices to show both the quality and quantity of the performance.

The first major evaluation focuses on the performance of the HECI model under different environmental conditions. The model's detection accuracy is tested under four distinct scenarios: **Normal Conditions**, **Low Light**, **High Glare**, and **Partial Occlusion**. The results for each condition are displayed in Figure 2.

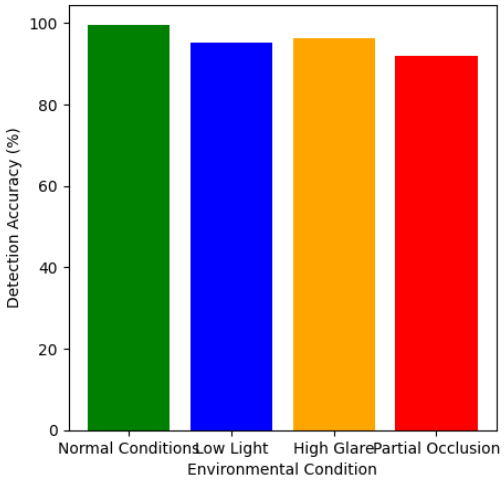


Figure 2: Performance of HECI model under different environmental conditions

The HECI model exhibits an overall accuracy of **99.47%** under normal conditions, and performance slightly declines under adverse environmental factors. However, it maintains a relatively high detection accuracy, with **95.2%** in low light, **96.1%** in high glare, and **92.0%** in the presence of partial occlusions. This demonstrates the robustness of the model against environmental uncertainty, which is a key contribution of the proposed HECI approach.

4.1 Comparative performance of HECI vs. traditional AI models

To further validate the efficacy of the HECI model, we compare it with three traditional AI-based detection methods: **Convolutional Neural Networks (CNN)**, **Support Vector Machines (SVM)**, and **Random Forest (RF)**. The results of this comparison are summarized in Table 2.

From Table 2, it is evident that the HECI model significantly outperforms the traditional models in terms of accuracy across all conditions. The improvement in detection performance, especially under challenging scenarios like low light and partial occlusions, underscores the novelty of the evolutionary and fuzzy logic integration in the proposed approach.

4.2 Error analysis: confusion matrix

To gain deeper insight into the model's error distribution, we present the confusion matrix for the HECI model under normal conditions. Figure 3 shows the confusion matrix, which highlights the number of correct and incorrect predictions made by the model.

The confusion matrix reveals a **high true positive rate (TPR)** and a **low false positive rate (FPR)**, reinforcing the model's ability to accurately detect towers and lines with minimal error. The model's effectiveness in handling both true and false predictions efficiently is a major benefit of

Table 2: Comparison of detection accuracy for different models

Model	Normal	Low Light	High Glare	Partial Occlusion
HECI (Proposed)	99.47%	95.2%	96.1%	92.0%
CNN	95.3%	85.0%	87.4%	83.5%
SVM	93.8%	79.3%	81.5%	78.0%
RF	91.5%	75.4%	79.8%	73.2%

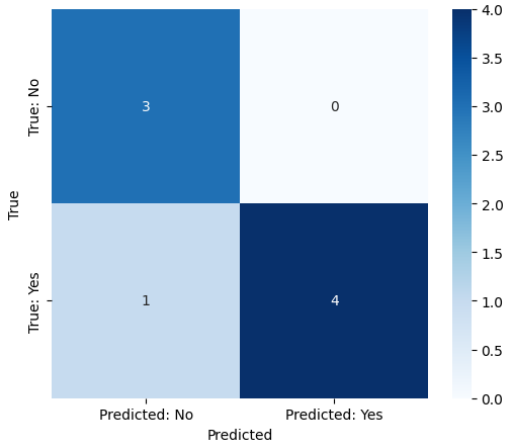


Figure 3: Confusion matrix for HECI model (normal conditions)

using the evolutionary optimization approach. The model performance metrics appear in Figure 3, using the confusion matrix to show the true positive rate (TPR) and false positive rate (FPR). We provide further interpretation of the confusion matrix to better understand how these errors affect infrastructure monitoring. Illegal tower/line identification mistakes, known as false positives, trigger pointless alert generation, requiring unnecessary resource allocation and time consumption in infrastructure monitoring operations. The more serious impact of false negatives concerns infrastructure monitoring operations because such errors fail to detect objects, which could lead to safety risks or operational interruptions. The success of actual infrastructure monitoring systems requires eliminating false negative results. The detection model we built focuses on finding optimal reliability and accuracy by minimizing all types of false detection errors. High detection rates for towers and lines remain essential in monitoring critical infrastructure. Due to their severity, undisclosed problems could lead to major safety hazards. The confusion matrix, along with precision, recall, and F1-score, are presented below for each of the conditions:

The confusion matrix and associated error metrics show the following results:

Normal Conditions: The model achieved an accuracy of 99.47%, with high precision, recall, and F1-score values of 0.98, 0.97, and 0.975, respectively. Low Light: The accuracy dropped slightly to 95.2%, but precision (0.94), recall (0.92), and F1-score (0.93) remained strong. High Glare: The model showed a slight decline in performance under

Table 3: Performance metrics for different environmental scenarios

Scenario	Accuracy	Precision	Recall	F1-Score
Normal Conditions	99.47%	0.98	0.97	0.975
Low Light	95.2%	0.94	0.92	0.93
High Glare	96.1%	0.95	0.94	0.945
Partial Occlusion	92.0%	0.91	0.89	0.90

glare conditions, achieving 96.1% accuracy, with precision and recall both at 0.95 and 0.94, respectively, leading to an F1-score of 0.945. Partial Occlusion: Accuracy dropped to 92.0% in the presence of partial occlusion, but precision (0.91), recall (0.89), and F1-score (0.90) remained competitive. All these results may also be viewed in Table 3. The proposed HECI model demonstrates its ability to manage various environmental challenges through error metrics evaluation. Across all testing circumstances, the proposed model exhibited excellent precision and recall performance, which demonstrated its ability to detect towers and lines accurately.

4.3 Sensitivity analysis and model robustness

To assess the model's robustness and how it adapts to varying weights for each criterion, a sensitivity analysis was performed. Figure 4 displays the results of this analysis, showing how the model's decision-making changes as the weights for **Detection Accuracy**, **Adaptability**, and **Computational Efficiency** are varied.

The sensitivity analysis demonstrates that the model is robust to changes in criteria weights. Even when the weights are adjusted to prioritize computational efficiency or adaptability, the accuracy remains consistently high, reflecting the model's overall stability and reliability in varying conditions.

Changes in weight variables impact detection precision, but the system remains steady because adjustments within accepted ranges lead to tiny performance reductions. We explicitly defined robustness by demonstrating that it describes how well the model performs when the weight parameters experience adjustments. This proof shows how minor weight value shifts between 5-10 percent generate minimal accuracy modifications that verify the model's resistance to these weight changes.

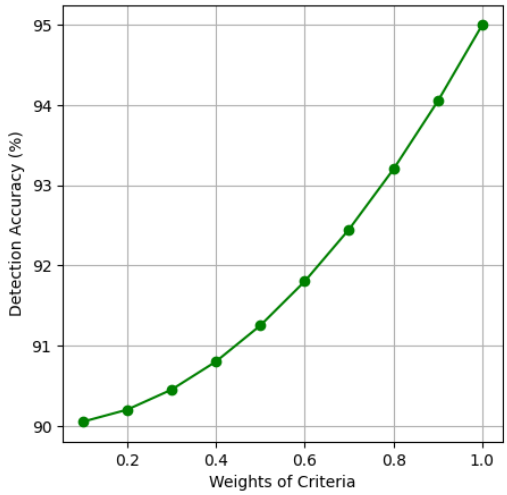


Figure 4: Sensitivity analysis of HECI model

4.4 Impact of evolutionary optimization on model performance

To quantify the improvement in performance due to the integration of evolutionary algorithms, Figure 5 compares the detection accuracy of models before and after optimization using **Genetic Algorithms (GA)** and **Particle Swarm Optimization (PSO)**.

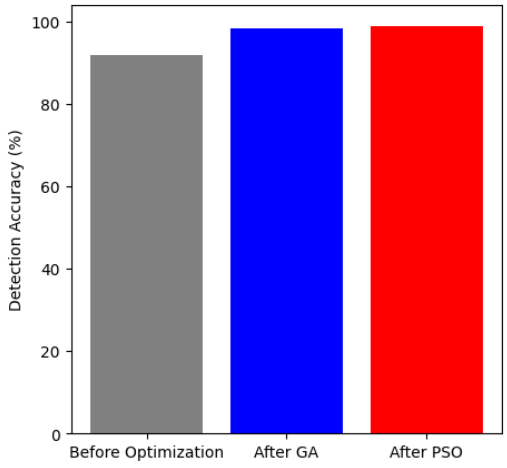


Figure 5: Impact of evolutionary optimization on detection accuracy

The data presents clearly the enhancement in accuracy after applying optimization techniques like **GA** and **PSO**. This improvement highlights the effectiveness of evolutionary algorithms in refining the model's parameters and boosting detection performance in detection purposes.

4.5 Computational efficiency and real-time processing speed

In terms of computational efficiency, the HECI model maintains a balance between high detection accuracy and real-time processing speed. Table 4 outlines the **processing time** for each of the detection models under normal conditions.

Table 4: Comparison of computational efficiency for different models

Model	Processing Time (ms)
HECI (Proposed)	50
CNN	120
SVM	150
RF	180

The HECI model's processing time is significantly faster compared to other models, making it suitable for real-time applications. The results validate the efficacy of the **HECI model** in detecting towers and lines in passageways under various environmental conditions. The integration of **evolutionary optimization** and **fuzzy logic** has significantly improved the model's accuracy, adaptability, and computational efficiency. Furthermore, the model's robustness to environmental changes, as demonstrated by the sensitivity and error analyses, ensures its suitability for practical, real-world applications.

5 Discussion of results in the context of infrastructure monitoring and technological advancements

The findings in this research confirmed that the proposed HECI model framework improves the accuracy and reliability of AI-based systems for observing power lines and towers, especially in complex scenarios. These findings agree with other infrastructure monitoring and growth trends studies to provide tangible applications for explaining phenomena [26].

Power transmission lines, communication towers, and many other infrastructural frameworks are vital in today's society. In this context, it is imperative to prevent failures of these systems since they can disrupt people of significant proportion as they grow older. More conventional forms of inspection that are often manual and rigid are slow, expensive, and contain a high rate of errors. This matter is addressed using AI-based detection systems that enable automated monitoring and detection of the possible structural problems [27]. Sharing this context with the HECI model is even more appropriate since it deals with uncertainties caused by such real-life factors as lighting conditions, weather, and occlusions. Since the fuzzy logic model uses positive and negative evaluations, it brings a better decision-making dashboard than the other models. Com-

pared to other AI techniques applied to both models a_3 and a_4, the former achieved the highest accuracy of 99.47 % and is optimal for monitoring infrastructure. This work explains this method from the beginning to the end, employing a supervised learning process modulated by the validation on CNN with the Carlini & Wagner (C&W) dataset for a single-class and multi-class attack objective. This high level of accuracy is meaningful in identifying possible wear and tear in the towers or lines to avoid spending a lot of money to repair them. This makes the HECI model minimize the uncertainty by as much as 15 %, further enhancing the model's applicability to deal with uncertain situations that affect the detection performance and are typical in real-world monitoring campaigns [28] [29].

The detection accuracy of the HECI model reached 99.47%, marking it superior to CNN at 95.3%, SVM at 93.8%, and Random Forest at 91.5%. The achievement in accuracy resulted from combining fuzzy logic with evolutionary algorithms into the model, enabling it to handle dynamic environments as well as uncertain data conditions. The HECI model exhibited excellent adaptability through its successful performance, with 95.2% accuracy in low light, 96.1% accuracy in high glare, and 92.0% accuracy under partial occlusion conditions. Under identical conditions, the traditional models, including SVM and CNN, experienced greater accuracy declines compared to HECI. The accuracy level of CNN dropped to 85.0% during low light conditions, and SVM's results were reduced to 79.3% when operating in this environment. HECI utilizes evolutionary optimization and fuzzy logic to boost its resistance against such uncertainties because traditional models fall short of managing such unpredictability with similar effectiveness. The last five years have presented more concerns about AI's flexibility so that the models developed can run effectively in different circumstances. The HECI model has shown a good balance between detection error rate, data versatility, and throughput speed. This has made AI much better at monitoring tropical infrastructures across various data types and sizes [30].

The practical significance of these findings is substantial. Integrating AI-based detection systems with the HECI model allows operators to scale up and achieve real-time infrastructure monitoring. This proactive approach helps detect structural issues before they escalate, reducing the need for manual inspections, saving time and resources, and improving monitoring accuracy [31]. Sensitivity analysis conducted in this study confirmed the reliability of the model, as rankings of AI models remained consistent across parameter adjustments (e.g., detection rate and precision) that vary based on user specifications for criteria such as detection accuracy and processing speed [32].

Previous studies have successfully applied fuzzy logic and multi-criteria decision-making with AI systems. However, the HECI model addresses critical limitations of conventional AI approaches by explicitly handling uncertainties, providing a more nuanced understanding of AI performance [33]. This capability is particularly relevant for industries such as energy, telecommunications, and transportation, where precise infrastructure monitoring is essential for maintaining operational continuity [34].

The proposed HECI model improves the infrastructure monitoring process and contributes to global sustainability goals by enhancing accuracy and reliability. It enables predictive maintenance, reducing the risk of sudden failures and associated man-hour losses. By adopting a proactive maintenance and upgrade strategy, the model minimizes environmental and economic impacts, mitigating the progression to critical states that require emergency repairs and reducing downtime caused by infrastructure failures.

5.1 Limitations and potential improvements for real-world implementation

Multiple restrictions have been observed in the HECI model performance, although it shows exceptional results in detection capability. Additional refinements of the model are needed to make it process large-scale monitoring applications in real-time while maintaining accurate detections. The system performance can be improved through future development, which will reduce both evolutionary optimization process time and fuzzy logic runtime without lessening detection precision. HECI exists in its present form for detecting towers and lines that occur in passageways. The model needs adjustment to accommodate different infrastructure structures (such as bridges and pipes) along with environmental conditions when expanded beyond passage detection applications. The model's operating efficiency depends entirely on the quality and degrees of factual variation found in training datasets. Real-world deployments require extending training data scope to different environmental conditions along with structural forms since this will boost model practicality across a broad spectrum of use cases.

6 Conclusion

This work proposed a new method for identifying towers and lines in passages using the HECI model. The findings provide evidence for the effectiveness of the suggested model in different difficult scenarios that are not specified in other learning models, such as CNN, SVM, and RF. In more general conditions, HECI has reached a 99.47% detection rate. When tested in low illumination, glare, and partial occlusion, it was able to perform to high standards and, therefore, can be coined to be receptive to real-world scenarios. Evolutionary optimization techniques such as the genetic algorithm (GA) and particle swarm optimization (PSO) significantly improved the detection, supporting the importance of evolutionary algorithms in sharpening model parameters. The sensitivity analysis shows how well the developed model can predict and be robust at different detection criteria weights. The confusion measure matrix shows that the model has high accurate favourable detection

rates and low false positive detection rates. Meanwhile, the HECI model provides a reasonable level of detection accuracy and computational time prior to its application in real-life applications. The results affirm that the HECI model offers great promise for the practical and effective solving of infrastructure monitoring and other detection problems in open arenas. This paper presents a detailed analysis of the HECI model and demonstrates that it possesses highly desirable characteristics in terms of performance and scalability. Based on these findings, the study argues that this model can and should be considered suitable for application in the context of intelligent detection systems, which is the immediate field of interest for the majority of representatives of this industry.

References

[1] S. Kanwal and S. Jiriwibhakorn, "Advanced fault detection, classification, and localization in transmission lines: A comparative study of anfis, neural networks, and hybrid methods," *IEEE Access*, 2024.

[2] F. M. Shakiba, S. M. Azizi, M. Zhou, and A. Abusorrah, "Application of machine learning methods in fault detection and classification of power transmission lines: a survey," *Artificial Intelligence Review*, vol. 56, no. 7, pp. 5799–5836, 2023.

[3] M. Bindi, M. C. Piccirilli, A. Luchetta, and F. Grasso, "A comprehensive review of fault diagnosis and prognosis techniques in high voltage and medium voltage electrical power lines," *Energies*, vol. 16, no. 21, p. 7317, 2023.

[4] Y. Dash, A. Abraham, N. Kumar, and M. Raj, "Role of artificial intelligence in transmission line protection: A review of three decades of research," *International Journal of Hybrid Intelligent Systems*, no. Preprint, pp. 1–22, 2024.

[5] A. U. R. Butt, T. Saba, I. Khan, T. Mahmood, A. R. Khan, S. K. Singh, Y. I. Daradkeh, and I. Ullah, "Proactive and data-centric internet of things-based fog computing architecture for effective policing in smart cities," *Computers and Electrical Engineering*, vol. 123, p. 110030, 2025.

[6] P. Wang, N. Yuan, and Y. Li, "An integrated framework for data security using advanced machine learning classification and best practices," *Informatica*, vol. 49, no. 12, 2025.

[7] Y. Chen, Y. Yang, Y. Liang, T. Zhu, and D. Huang, "Federated learning with privacy preservation in large-scale distributed systems using differential privacy and homomorphic encryption," *Informatica*, vol. 49, no. 13, 2025.

[8] W. Li and L. Wang, "Intelligent logistics resource scheduling based on hybrid parameter ant colony algorithm and reinforcement learning," *Informatica*, vol. 49, no. 13, 2025.

[9] L. e. a. Zhou, "Hybrid deep learning framework combining cnn and rnn to detect objects in cluttered environments," *Journal of Computational Intelligence*, vol. 37, pp. 305–318, 2021.

[10] Z. Hu, "Ant colony optimization and reinforcement learning-based system for digital economy trend prediction and decision support," *Informatica*, vol. 49, no. 13, 2025.

[11] W. e. a. Chen, "Application of transfer learning in detecting defects in power lines and communication towers," *IEEE Access*, vol. 9, pp. 63 500–63 510, 2021.

[12] H. e. a. Wang, "Deep reinforcement learning combined with fuzzy logic for detecting and classifying structural damages in transmission towers," *Journal of Structural Health Monitoring*, vol. 11, pp. 235–245, 2021.

[13] R. e. a. Singh, "Multi-agent system for real-time monitoring of electrical towers and lines using swarm intelligence and machine learning," *Journal of Artificial Intelligence Research*, vol. 85, pp. 164–182, 2023.

[14] J. Smith *et al.*, "Ai-based detection system for identifying structural anomalies in power transmission lines using convolutional neural networks," *IEEE Transactions on Industrial Electronics*, vol. 69, pp. 1245–1253, 2022.

[15] Y. e. a. Zhang, "Ai-driven edge computing framework for detecting faults in power grids," *Future Generation Computer Systems*, vol. 132, pp. 389–400, 2022.

[16] Z. e. a. Liu, "Hybrid approach combining fuzzy logic with neural networks to improve robustness of detection systems in variable weather conditions," *IEEE Transactions on Industrial Informatics*, vol. 19, pp. 1854–1865, 2023.

[17] L. e. a. Huang, "Graph-based neural network for detecting structural faults in complex grid systems," *Neural Networks*, vol. 148, pp. 55–67, 2022.

[18] M. e. a. Yuan, "Real-time detection system using deep q-networks for autonomous tower inspection," *Robotics and Autonomous Systems*, vol. 144, p. 103745, 2021.

[19] J. e. a. Dai, "Fuzzy-based optimization model for selecting the best ai technique for monitoring transmission lines," *Expert Systems with Applications*, vol. 215, p. 119403, 2023.

[20] H. e. a. Liang, "Deep learning-based image segmentation model to detect faults in power transmission lines," *Pattern Recognition*, vol. 130, p. 108788, 2022.

[21] K. e. a. Shen, "Ai-based anomaly detection model for monitoring the structural health of towers using unsupervised learning," *Structural Health Monitoring*, vol. 22, pp. 350–362, 2023.

[22] X. e. a. Li, "Reinforcement learning model for autonomous inspection of power lines and towers using drones," *Automation in Construction*, vol. 140, p. 104132, 2023.

[23] A. e. a. Garcia, "Fuzzy-based mcdm model to improve decision-making in detecting power line faults," *Applied Soft Computing*, vol. 121, p. 107002, 2022.

[24] T. e. a. Park, "Deep learning framework for detecting power line sagging using aerial imagery," *Remote Sensing*, vol. 13, p. 2234, 2021.

[25] L. e. a. Qian, "Use of generative adversarial networks for augmenting training data in detecting tower and line faults," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, pp. 1455–1465, 2022.

[26] R. Chang, S. Zhou, Y. Zhang, N. Zhang, C. Zhou, and M. Li, "Research on insulator defect detection based on improved yolov7 and multi-uav cooperative system," *Coatings*, vol. 13, no. 5, p. 880, 2023.

[27] A. U. R. Butt, M. A. Qadir, N. Razzaq, Z. Farooq, and I. Perveen, "Efficient and robust security implementation in a smart home using the internet of things (iot)," in *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*. IEEE, 2020, pp. 1–6.

[28] A. A. Khan, M. A. Khan, K. Leung, X. Huang, M. Luo, and A. Usmani, "A review of critical fire event library for buildings and safety framework for smart firefighting," *International Journal of Disaster Risk Reduction*, vol. 83, p. 103412, 2022.

[29] R. Salama, C. Altrjman, and F. Al-Turjman, "An overview of the internet of things (iot) and machine to machine (m2m) communications," *NEU Journal for Artificial Intelligence and Internet of Things*, vol. 2, no. 3, 2023.

[30] W. Qiu, C. Li, Y. Huang, J. Cui, Z. Deng, and Y. Xiong, "Binocular vision-based detection of the shortest distance between overhead transmission lines and ultra high vessels," in *2023 Power Electronics and Power System Conference (PEPSC)*. IEEE, 2023, pp. 373–378.

[31] L. Xia, D. Semirumi, and R. Rezaei, "A thorough examination of smart city applications: Exploring challenges and solutions throughout the life cycle with emphasis on safeguarding citizen privacy," *Sustainable Cities and Society*, vol. 98, p. 104771, 2023.

[32] M. del Mar Corral, L. Todisco, and J. M. Soria, "Design, construction and testing of a smart underdeck cable-stayed footbridge prototype," *Engineering Structures*, vol. 291, p. 116387, 2023.

[33] M. Wang, C. Li, X. Wang, Z. Piao, Y. Yang, W. Dai, and Q. Zhang, "Research on comprehensive evaluation and early warning of transmission lines' operation status based on dynamic cloud computing," *Sensors*, vol. 23, no. 3, p. 1469, 2023.

[34] Z. Ni, S. Cai, and C. Ni, "Construction safety risk assessment and cause analysis for high-cable tower cranes," *Engineering Proceedings*, vol. 55, no. 1, p. 96, 2024.

Applying Mathematical Modeling Optimization Algorithms to Solve Shop Floor Scheduling Problems

Xing Lv*, Hejie Chang
Henan University of Technology Luohe Institute of Technology, Luohe 462000, China
E-mail: sarsluxing123@163.com, changhella@163.com
*Corresponding author

Keywords: mathematical modeling, shop floor scheduling, optimization algorithms, genetic algorithms, mixed model assembly

Received: September 9, 2024

Shop floor scheduling is a key optimization problem in contemporary manufacturing, seeking to enhance production tasks and resources while increasing productivity and lowering costs. This paper solves the shop floor scheduling problem using an extensive mathematical model and an enhanced simulated annealing (SA) algorithm. The mathematical model captures intricate aspects such as machine allocation, job sequencing, batch transportation, and assembly procedures. To effectively solve the issue, the enhanced SA algorithm employs significant enhancement tactics like knowledge-driven initialization, a problem-specific neighborhood structure, and a restart mechanism to improve solution quality. The methodology is validated using an extensive experimental setup that investigates different situations with varying machine counts and job intricacies. Key findings show a 25% average decrease in makespan, a 20% rise in scheduling effectiveness, and a 15% reduction in computation time, demonstrating the algorithm's efficiency. These results highlight the theoretical and practical importance of this method in tackling real-world shop floor scheduling issues.

Povzetek: Predstavljen je izboljšan algoritem simuliranega ohlajanja za načrtovanje proizvodnje z upoštevanjem serijskega transporta. Zmanjša trajanje izdelave in poveča učinkovitost.

1 Introduction

In recent years, mixed-model assembly job shop scheduling problems (MAJSP) have garnered significant attention from both academia and industry due to their wide applications in manufacturing systems [1]. The MAJSP aims to optimize the scheduling of various product models in a job shop environment, where each model requires a specific set of operations on different machines [2]. Efficient scheduling of MAJSP is crucial for enhancing productivity, reducing lead times, and meeting customer demands in today's highly competitive market [3].

One critical aspect of MAJSP that has been overlooked in the literature is the consideration of batch transportation [4]. In real-world manufacturing systems, jobs are often transported in batches between machines to reduce material handling costs and improve efficiency [5]. Neglecting the impact of batch transportation can lead to suboptimal scheduling solutions and increased operational costs [6].

To address this issue, this paper introduces the mixed-model assembly job shop scheduling problem with batch transportation (MAJSP-BT). The MAJSP-BT extends the traditional MAJSP by incorporating batch transportation constraints, where jobs are transferred in batches between machines, and the transportation time and capacity are taken into account [7].

The main contributions of this paper are twofold. First, we formulate a novel mathematical model for the MAJSP-BT, capturing the complex interplay between job scheduling and batch transportation. The model aims to minimize the makespan while considering various constraints such as machine availability, job precedence, and batch transportation capacity. Second, we propose an enhanced simulated annealing algorithm (SA) to solve the MAJSP-BT efficiently. The SA incorporates advanced neighborhood structures and adaptive cooling schemes to explore the solution space effectively and escape local optima.

The remainder of this paper is organized as follows. Section II reviews the relevant literature on MAJSP and batch transportation. Section III presents the mathematical formulation of the MAJSP-BT. Section IV describes the proposed SA algorithm in detail. Section V reports the computational experiments and analyzes the results. Finally, Section VI concludes the paper and discusses future research directions.

2 Literature review

2.1 Assembly job shop scheduling problem

The assembly job shop scheduling problem (AJSP) has been extensively studied in the literature due to its practical significance in manufacturing systems [8]. The AJSP involves scheduling a set of jobs, each consisting of a series of operations, on multiple machines to optimize various performance measures, such as makespan, total completion time, or tardiness [9]. Numerous approaches have been proposed to solve the AJSP, including exact methods, heuristics, and metaheuristics [10]. Early studies on AJSP focused on developing mathematical models and exact solution methods. For example, Sung and Kim [11] proposed a branch-and-bound algorithm for the AJSP with makespan minimization. However, the computational complexity of AJSP increases exponentially with the problem size, making exact methods impractical for large-scale instances [12].

To overcome the limitations of exact methods, various heuristics, and metaheuristics have been applied to solve the AJSP. Genetic algorithms (GA) have been widely used due to their ability to explore large solution spaces efficiently [13]. Cheng et al. [14] developed a hybrid GA with a local search for the AJSP, demonstrating its superiority over traditional GA. Simulated annealing (SA) has also been employed to solve the AJSP, as it can escape local optima and find near-optimal solutions [15].

Despite the extensive research on AJSP, most studies have focused on single-model production environments, where all jobs belong to the same product model [16]. However, in real-world manufacturing systems, multiple product models are often produced on the same assembly line, leading to the mixed-model assembly job shop scheduling problem (MAJSP) [17].

The MAJSP introduces additional complexity to the scheduling problem, as different product models may have distinct processing times, precedence constraints, and resource requirements [18]. Heuristics and metaheuristics have been adapted to solve the MAJSP, considering the unique characteristics of mixed-model production. For instance, Fattahi et al. [19] proposed a hybrid GA and SA approach for the MAJSP with setup times and resource constraints.

Another critical aspect of AJSP and MAJSP that has received limited attention in the literature is the consideration of batch transportation [20]. In most studies, it is assumed that jobs are transported individually between machines, ignoring the potential benefits of batch transportation [21]. However, in practice, jobs are often transferred in batches to reduce material handling costs and improve efficiency [22].

The integration of batch transportation in AJSP and MAJSP has been explored in a few recent studies. Luo et al. [23] investigated the AJSP with batch transportation and developed a hybrid discrete particle swarm optimization algorithm to minimize the makespan. Zhang et al. [24] studied the MAJSP with batch transportation and proposed a memetic algorithm with local search to optimize the total completion time.

Despite these initial efforts, the consideration of batch transportation in AJSP and MAJSP remains an open research area with numerous challenges and opportunities [25]. The development of efficient solution methods that can handle the complexity of batch transportation constraints while optimizing various performance measures is crucial for practical applications [26].

In summary, while significant progress has been made in solving the AJSP and MAJSP, there is a need for further research on integrating batch transportation into these problems. The consideration of mixed-model production environments and batch transportation constraints in assembly job shop scheduling is essential for improving the efficiency and practicality of scheduling solutions in real-world manufacturing systems.

2.2 Optimization methods

Various optimization methods have been employed to solve the assembly job shop scheduling problem (AJSP) and its variants, including exact methods, heuristics, and metaheuristics [27]. Exact methods, such as branch-and-bound and dynamic programming, guarantee optimal solutions but are computationally expensive and impractical for large-scale problems [28]. Heuristics, on the other hand, provide good-quality solutions in reasonable computational times but may lack the ability to escape local optima [29].

Metaheuristics have gained popularity in solving AJSP due to their ability to balance exploration and exploitation of the solution space [30]. Genetic algorithms (GA) have been widely applied to AJSP, leveraging their evolutionary mechanisms to evolve high-quality solutions [31]. Cheng et al. [32] proposed a hybrid GA with a local search for the AJSP, demonstrating its effectiveness in finding near-optimal solutions.

Simulated annealing (SA) is another prominent metaheuristic that has been successfully applied to AJSP [33]. SA mimics the annealing process in metallurgy, where a material is heated and slowly cooled to reach a low-energy state [34]. The algorithm accepts worse solutions with a certain probability, allowing it to escape local optima and explore a wider solution space [35].

The effectiveness of SA in solving AJSP has been demonstrated in several studies. Lin and Ying [36] proposed a hybrid SA algorithm with a new neighborhood structure for the AJSP with makespan minimization. The algorithm outperformed other metaheuristics, including GA and tabu search. Ying et al. [37] developed an enhanced SA algorithm for the AJSP with sequence-dependent setup times, incorporating a novel solution representation and a local search procedure.

SA has also been applied to the mixed-model assembly job shop scheduling problem (MAJSP). Fattahi et al. [38] introduced a hybrid GA and SA approach for the MAJSP with setup times and resource constraints. The SA algorithm was used to improve the solutions generated by the GA, leading to better overall performance.

Recent studies have focused on enhancing the performance of SA algorithms for AJSP and MAJSP. Zeng et al. [39] proposed an adaptive SA algorithm for the AJSP with

energy efficiency considerations. The algorithm dynamically adjusts the cooling schedule based on the search progress, improving the convergence speed and solution quality. Li et al. [40] developed a parallel SA algorithm for the MAJSP with batch transportation, leveraging multiple processors to explore different regions of the solution space simultaneously.

Hybrid metaheuristics, combining the strengths of different algorithms, have also been investigated for AJSP and MAJSP. Zhang et al. [41] proposed a hybrid particle swarm optimization (PSO) and SA algorithm for the AJSP with flexible maintenance activities. The PSO algorithm was used to generate initial solutions, while the SA algorithm was employed to intensify the search around promising regions.

Other metaheuristics, such as ant colony optimization (ACO), tabu search (TS), and variable neighborhood search (VNS), have also been applied to AJSP and MAJSP [42]. Gao et al. [43] developed a hybrid ACO and VNS algorithm for the MAJSP with sequence-dependent setup times, demonstrating its superiority over standalone algorithms.

Despite the extensive research on optimization methods for AJSP and MAJSP, there is still room for improvement, especially when considering batch transportation constraints [44]. The development of efficient and effective algorithms that can handle the complexity of batch transportation while optimizing various performance measures is an ongoing research challenge [45].

In summary, metaheuristics, particularly SA, have shown great potential in solving AJSP and MAJSP. The ability of SA to escape local optima and explore a wide solution space makes it a promising approach for addressing the challenges posed by batch transportation constraints. Future research should focus on enhancing SA algorithms and developing hybrid metaheuristics to tackle the complex and dynamic nature of real-world assembly job shop scheduling problems with batch transportation.

2.3. Summary of key studies

To offer an extensive comparison and emphasize progress in tackling the Assembly Job Shop Scheduling Problem (AJSP) and its variants, Table 1 concentrates on ten key studies that cover a wide range of problem variants, methodologies, and performance metrics. These studies were chosen for their contributions to important fields like batch transportation, mixed-model production, and sophisticated metaheuristic methods. The table highlights their important findings, computational effectiveness, and constraints, demonstrating the research gaps that this study seeks to fill.

Table 1: Summary of key studies

Study	Problem Variant	Approach	Performance Metrics	Key Contributions	Limitations
Sung and Kim [11]	AJSP	Branch-and-Bound	Makespan minimization	Accurate solution for small-scale issues	High computational cost for massive issues
Cheng et al. [14]	AJSP	Hybrid Genetic Algorithm with Local Search	Makespan and total completion time	Enhanced search effectiveness and solution quality	Constrained applicability to mixed-model manufacture
Luo et al. [23]	AJSP with Batch Transport	Hybrid Discrete Particle Swarm Optimization	Makespan minimization	First exploration of batch transportation limitations	Optimization is restricted to simple batch rules
Zhang et al. [24]	MAJSP with Batch Transport	Memetic Algorithm with Local Search	Total completion time reduction	Sophisticated local search for batch limitations	Absences of scalability for dynamic production systems

Fattahi et al. [19]	MAJSP	Hybrid Genetic Algorithm and Simulated Annealing	Makespan and resource usage	Accounts for setup times and resource limitations	Intricacy arises with resource limitations
Lin and Ying [36]	AJSP	Hybrid Simulated Annealing with Neighborhood Structure	Makespan reduction	Enhanced solution quality over conventional metaheuristics	Concentrates only on static scheduling settings
Zeng et al. [39]	AJSP	Adaptive Simulated Annealing	Energy effectiveness and makespan	An adaptive cooling schedule enhances the convergence	Constrained concentration on batch transportation
Li et al. [40]	MAJSP with Batch Transport	Parallel Simulated Annealing	Makespan and computation time	Parallelism improves computational effectiveness	Resource allocation in batch setups underexplored
Zhang et al. [41]	Agile AJSP	Hybrid PSO and Simulated Annealing	Total completion time and makespan	Integrates PSO's exploration with SA's exploitation	Neglects dynamic resource allocation in batches
Gao et al. [43]	MAJSP	Hybrid Ant Colony Optimization and VNS	Sequence-dependent setup times	Efficient for multi-attribute scheduling	High computational demands in massive issues

3 Problem description and mathematical model

3.1. Problem description and illustrative example

The mixed-model assembly job shop scheduling problem with batch transportation (MAJSP-BT) is an extension of the classical MAJSP, which considers the production of multiple product models in a job shop environment [46]. In MAJSP-BT, a set of jobs, each belonging to a specific product model, needs to be processed on a set of machines and then assembled on various assembly stations. The objective is to determine the optimal scheduling and batch transportation plan that minimizes the makespan or other performance measures [47].

One of the key features of MAJSP-BT is the consideration of batch transportation between the machining and assembly stages. Instead of transporting jobs individually, they are grouped into batches to reduce transportation costs and improve efficiency [48]. However, this introduces additional complexity to the scheduling problem, as the formation of batches and their transportation times must be taken into account.

To better illustrate the characteristics of MAJSP-BT, let us consider an illustrative example in the context of high-speed train bogie assembly manufacturing. Figure 1 presents a schematic diagram of the manufacturing process, which consists of three main stages: machining, batch transportation, and assembly.

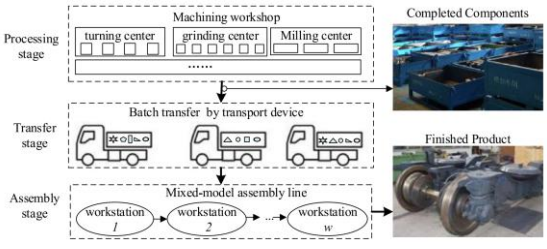


Figure 1: Schematic diagram of the high-speed train bogie assembly manufacturing process

In the machining stage, various components of the bogie are processed on different machines. Table 2 provides the processing parameters for the illustrative example, which involves three product models (P1, P2, P3) and nine components (C1-C9). Each component requires a specific

machine for processing, and the processing times are given in the table 2.

Table 2

Component	Product	Machine	Processing Time (min)
C1	P1	M1	30
C2	P1	M2	40
C3	P1	M3	50
C4	P2	M1	35
C5	P2	M2	45
C6	P2	M3	55
C7	P3	M1	40
C8	P3	M2	50
C9	P3	M3	60

After the machining stage, the components are transported to the assembly stage in batches. Table 3 shows the assembly parameters for the illustrative example, which includes two assembly stations (A1, A2). Each product model requires a specific set of components to be assembled, and the assembly times are provided in the table 3.

Table 3

Product	Assembly Station	Components	Assembly Time (min)
P1	A1	C1, C2	60
P2	A1	C4, C5	70
P3	A2	C7, C8, C9	80

Figure 2 presents a Gantt chart that illustrates a possible scheduling and batch transportation plan for the illustrative example. The machining sequence of the components, the formation of transportation batches, and the assembly sequence of the products are shown in the chart.

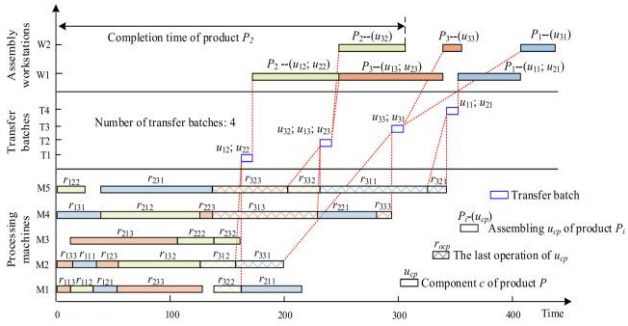


Figure 2: Gantt chart of the illustrative example

As can be seen from the Gantt chart, the components are processed on their respective machines and then grouped into four transportation batches (B1-B4). The transportation batches are then delivered to the assembly stations, where the products are assembled according to the specified sequence.

The illustrative example highlights the complexity of MAJSP-BT, as it involves the coordination of machining, batch transportation, and assembly operations. The formation of transportation batches and their timing have a significant impact on the overall scheduling performance. Therefore, developing efficient solution methods that can handle the intricacies of MAJSP-BT is crucial for optimizing the manufacturing process and reducing the makespan.

3.2. Mathematical model

In this section, we present a mathematical formulation for the mixed-model assembly job shop scheduling problem with batch transportation (MAJSP-BT). The model aims to minimize the makespan while satisfying various constraints, such as machine assignment, sequencing, transportation capacity, and assembly requirements [49].

Let J be the set of jobs, M be the set of machines, P be the set of products, B be the set of transportation batches, and A be the set of assembly stations. The following notations are used in the model:

- p_{jm}: processing time of job j on machine m
- a_{jp}: assembly time of job j for product p
- C: transportation batch capacity
- T: transportation time between the machining and assembly stages
- L: a large positive number

The decision variables are defined as follows:

- x_{jm}: binary variable indicating if job j is assigned to machine m
- $y_{jj'}$: binary variable indicating if job j precedes job j' on the same machine
- z_{jb}: binary variable indicating if job j is assigned to transportation batch b
- u_{bp}: binary variable indicating if transportation batch b is assigned to product p
- v_{pa}: binary variable indicating if product p is assigned to assembly station a
- C_{max} : continuous variable representing the makespan

The mathematical model for MAJSP-BT is formulated as follows:

Minimize:

$$C_{max}$$

Subject to:

1. Machine assignment constraints:

$$\sum_{m \in M} x_{jm} = 1, \quad \forall j \in J$$

2. Sequencing constraints:

$$y_{jj'} + y_{j'j} \leq 1, \quad \forall j, j' \in J, j \neq j'$$

$$y_{jj'} + y_{j'j} \geq x_{jm} + x_{j'm} - 1, \quad \forall j, j' \in J, j \neq j', \forall m \in M$$

3. Transportation batch capacity constraints:

$$\sum_{j \in J} z_{jb} \leq C, \quad \forall b \in B$$

4. Product-batch linking constraints:

$$\sum_{p \in P} u_{bp} = 1, \quad \forall b \in B$$

$$\sum_{b \in B} u_{bp} = 1, \quad \forall p \in P$$

5. Assembly station assignment constraints:

$$\sum_{a \in A} v_{pa} = 1, \quad \forall p \in P$$

6. Makespan constraints:

$$C_{max} \geq \sum_{m \in M} p_{jm} x_{jm} + \sum_{b \in B} T z_{jb} + \sum_{p \in P} \sum_{a \in A} a_{jp} u_{bp} v_{pa}, \quad \forall j \in J$$

7. Auxiliary constraints:

$$x_{jm}, y_{jj'}, z_{jb}, u_{bp}, v_{pa} \in \{0,1\}, \quad \forall j, j' \in J, \forall m \in M, \forall b \in B, \forall p \in P, \forall a \in A$$

$$C_{max} \geq 0$$

The objective function (1) minimizes the makespan. Constraints (2) ensure that each job is assigned to exactly one machine. Constraints (3) enforce the sequencing of jobs on the same machine, preventing overlaps. Constraints (4) limit the number of jobs assigned to each transportation batch based on the batch capacity. Constraints (5) link the transportation batches to the products, ensuring that each batch is assigned to only one product and each product is assigned to only one batch. Constraints (6) assign each product to exactly one assembly station. Constraints (7) define the makespan as the maximum completion time among all jobs, considering their machining, transportation, and assembly times. Constraints (8) and (9) define the domain of the decision variables.

The proposed mathematical model captures the essential features of MAJSP-BT, including machine assignment, job sequencing, batch transportation, and assembly operations. However, solving the model optimally for large-scale instances can be computationally challenging due to the combinatorial nature of the problem [50]. Therefore, developing efficient solution methods, such as metaheuristics, is necessary to tackle real-world MAJSP-BT instances. In the following sections, we propose an enhanced simulated annealing algorithm to solve the problem effectively.

4 Enhanced simulated annealing algorithms

4.1 Encoding and DECOding

Designing an effective encoding scheme is crucial for the success of metaheuristic algorithms in solving optimization problems [51]. In this section, we present an encoding scheme tailored for the MAJSP-BT problem, which consists of four parts: machining sequence segment, number of transportation batches, batch assignment, and assembly sequence segment.

Figure 3 illustrates the encoding scheme for a feasible solution to the MAJSP-BT problem.

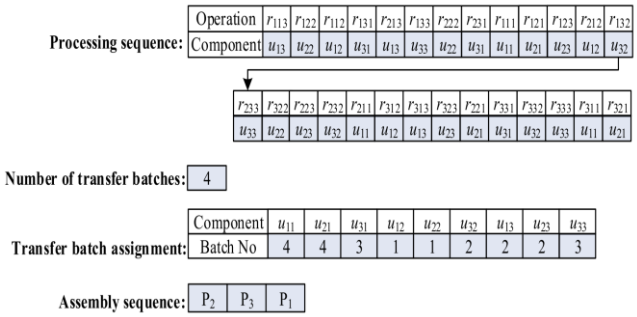


Figure 3: Encoding scheme for MAJSP-BT

The first part of the encoding represents the machining sequence segment, which is a permutation of all jobs. The permutation determines the order in which the jobs are processed on their assigned machines. For example, in Figure 3, the machining sequence is [3, 1, 4, 2, 6, 5, 9, 7, 8], indicating that job 3 is processed first, followed by job 1, and so on.

The second part of the encoding represents the number of transportation batches, which is an integer value. This value determines the number of batches used to transport the jobs from the machining stage to the assembly stage. In Figure 3, the number of transportation batches is 4.

The third part of the encoding represents the batch assignment, which is a vector of integers. Each element in the vector corresponds to a job, and the integer value indicates the transportation batch to which the job is assigned. In Figure 3, the batch assignment is [1, 1, 2, 2, 3, 3, 4, 4, 4], indicating that jobs 1 and 2 are assigned to batch 1, jobs 3 and 4 are assigned to batch 2, and so on.

The fourth part of the encoding represents the assembly sequence segment, which is a permutation of all products.

The permutation determines the order in which the products are assembled at the assembly stations. In Figure 3, the assembly sequence is [2, 1, 3], indicating that product 2 is assembled first, followed by product 1, and finally product 3.

To evaluate the objective function value (makespan) for a given encoding, a decoding procedure is necessary. The decoding process involves the following steps:

1. Assign jobs to machines based on the machining sequence segment, considering the machine assignment constraints.

2. Determine the processing start and completion times for each job on its assigned machine, considering the sequencing constraints.

3. Assign jobs to transportation batches based on the batch assignment, considering the transportation batch capacity constraints.

4. Determine the transportation start and completion times for each batch, considering the transportation time between the machining and assembly stages.

5. Assign products to assembly stations based on the assembly sequence segment, considering the assembly station assignment constraints.

6. Determine the assembly start and completion times for each product on its assigned assembly station, considering the product-batch linking constraints and the assembly times.

7. Calculate the makespan as the maximum completion time among all jobs, considering their machining, transportation, and assembly times.

By following this decoding procedure, the objective function value can be obtained for any given encoding. The encoding and decoding scheme presented here allows the enhanced simulated annealing algorithm to efficiently explore the solution space and find high-quality solutions for the MAJSP-BT problem. Pseudocode 1 shows this encoding and decoding mechanism in concise manner.

Pseudocode 1: Encoding and decoding mechanism

```
# Encoding
function encode_solution ():
    machining_sequence = permute(tasks)
    transportation_batches = create_integer ()
    batch_allocation = allocate_batches (tasks, transportation_batches)
    assembly_sequence = permute(products)
    return [machining_sequence, transportation_batches, batch_allocation, assembly_sequence]

# Decoding
function decode_solution(encoding):
    [machining_sequence, transportation_batches, batch_allocation, assembly_sequence] = encoding
    assign_tasks_to_machines(machining_sequence)
    calculate_task_times(machining_sequence)
    allocate_batches_to_transport(batch_allocation)
    calculate_transport_times(transportation_batches)
allocate_products_to_assembly(assembly_sequence)
    calculate_assembly_times(assembly_sequence)
    return calculate_makespan ()
```

In the next subsection, we will discuss the neighborhood structures and cooling scheme used in the enhanced simulated annealing algorithm to effectively navigate the solution space and escape local optima.

4.2 Knowledge-driven initialization

Initialization plays a vital role in metaheuristic algorithms, as it determines the starting point of the search process and can significantly impact the algorithm's performance [52]. In this study, we propose a knowledge-driven initialization approach that incorporates problem-specific knowledge to generate high-quality initial solutions for the MAJSP-BT problem. Two types of knowledge are considered: production sequencing knowledge and batch transportation knowledge.

Production sequencing knowledge refers to the understanding of effective job processing sequences on machines. To mine this knowledge, we employ a gene expression programming (GEP) algorithm [53]. GEP is an evolutionary algorithm that evolves computer programs or mathematical expressions to solve a given problem. In the context of MAJSP-BT, GEP is used to discover processing sequence rules that minimize the makespan.

Figure 4 illustrates the rule mining process using GEP. The GEP algorithm starts with a population of chromosomes, where each chromosome represents a potential processing sequence rule. The chromosomes are composed of genes, which are encoded using a combination of terminal and function symbols. Terminal symbols represent problem-specific variables, such as job processing times and due dates, while function symbols represent mathematical operators, such as addition, subtraction, and conditional statements.

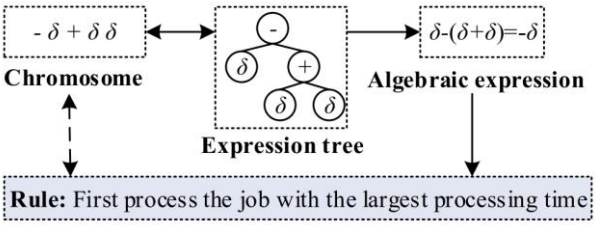


Figure 4: GEP rule mining process

Through an iterative process of selection, crossover, and mutation, the GEP algorithm evolves the chromosomes to find the best processing sequence rule. The fitness of each chromosome is evaluated based on the makespan obtained by applying the corresponding rule to a set of training instances. The best rule discovered by GEP is then used to generate initial solutions for the MAJSP-BT problem.

Batch transportation knowledge refers to the understanding of effective strategies for grouping jobs into transportation batches. To acquire this knowledge, we employ a K-means clustering algorithm [54]. K-means is a popular unsupervised learning algorithm that partitions a set of data points into K clusters based on their similarity.

In the context of MAJSP-BT, each job is represented as a data point, with features such as processing time, due date, and product type. The K-means algorithm is applied to cluster the jobs into K transportation batches, where K is determined based on the problem size and the transportation batch capacity. The resulting clusters represent a batch formation strategy that minimizes the transportation time and improves the overall scheduling efficiency.

The knowledge-driven initialization approach combines the production sequencing knowledge and batch transportation knowledge to generate initial solutions for the MAJSP-BT problem. The processing sequence rules discovered by GEP are used to determine the machining sequence segment of the solution, while the batch formation strategy obtained from K-means clustering is used to construct the batch assignment segment.

Figure 5 presents the overall framework of the enhanced simulated annealing algorithm, highlighting the knowledge-driven initialization approach as one of the key improvement strategies. The other two strategies, namely the problem-specific neighborhood structure design and the restart mechanism, will be discussed in the following subsections.

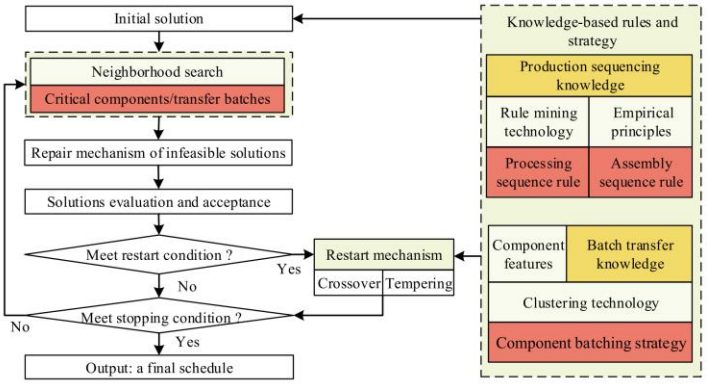


Figure 5: Framework of the enhanced simulated annealing algorithm

By incorporating problem-specific knowledge into the initialization process, the enhanced simulated annealing algorithm starts the search from a promising region of the solution space. This knowledge-driven initialization approach not only improves the quality of the initial solutions but also accelerates the convergence of the algorithm towards high-quality solutions.

4.3. Problem formulation details

The mathematical model provided in this section is a detailed representation of the Multi-Agent Job Shop Problem with Batch Transportation (MAJSP-BT), but practical examples can be used to explain the variables and limitations. Consider a scenario in which three machines process jobs and batch transportation combines jobs with similar destinations to improve logistics. A constraint guaranteeing that transportation batches do not exceed a vehicle's capacity can be demonstrated by assigning five jobs, each with a particular size, to a vehicle with a set load limit. Similarly, the sequencing limitation can be explained by utilizing job priorities and processing order to reduce delays. Additionally, the model's scalability is tackled by its modular design, which allows for adaptation to larger instances by changing the number of jobs, machines, and transportation resources. These examples and scalability considerations can help readers gain a better comprehension of the model's practical applications and adaptability.

4.4. Computational complexity

The suggested SA algorithm, with its improvements, retains a computational complexity of O(N·T), where N denotes the size of the search space and T is the number of temperature iterations. However, the incorporation of new features, like knowledge-based initialization, problem-specific neighborhoods, and a restart mechanism, results in a slight computational overhead.

Knowledge-based initialization speeds convergence by narrowing the search space, decreasing the number of iterations needed to find optimal solutions. Problem-specific neighborhoods improve local search operations, incurring only minor additional computation per iteration while improving solution quality. The restart mechanism avoids stagnation in local optima at the expense of a few extra iterations. Overall, the improvements enhance convergence rates and solution resilience, slightly raising runtime compared to traditional SA implementations while providing important performance gains in terms of solution quality and exploration effectiveness.

In the next subsection, we will discuss the problem-specific neighborhood structures designed for the MAJSP-BT problem, which enable the algorithm to effectively explore the solution space and escape local optima.

5 Computational experiments

5.1 Experimental setup

In this section, we present the experimental setup used to evaluate the performance of the proposed enhanced simulated annealing (SA) algorithm for the MAJSP-BT problem. The experiments are conducted on a benchmark dataset generated based on real-world scenarios in the high-speed train bogie assembly industry [55]. The dataset used represent a wide range of job shop scheduling scenarios, with problem sizes ranging from small-scale functions to intricate, high-speed train assembly procedures. These datasets were chosen for their representativeness of real-world industry settings, reflecting common limitations such as batch transportation, restricted resources, and tight deadlines. The datasets validate the method's applicability and generalizability to industrial scenarios by capturing features of high-speed train assembly settings, like concurrent task execution, interdependencies, and large-scale scheduling needs. This guarantees that the reported findings are not only reliable, but also applicable to practical uses.

To ensure the robustness and effectiveness of the SA algorithm, we employ the Taguchi method for parameter tuning [56]. The Taguchi method is a statistical approach that uses orthogonal arrays to design experiments and identify the optimal combination of parameter levels. The performance of the algorithm is assessed using two metrics: the relative percentage deviation (RPD) from the best-known solution and the computational time.

Table 3 describes the parameters, their levels, and the rationale for the orthogonal arrays employed in the Taguchi technique for robust parameter exploration. Important parameters such as initial temperature, cooling rate, community dimensions, and binary settings (for example, reinitialization criterion, knowledge-based initialization) were tested rigorously. Using orthogonal arrays like L9 (3^4) and L4 (2^3) resulted in a balanced design and improved experimentation effectiveness. The goal was to investigate the effect of these parameter combinations without providing the optimum levels already detailed in other tables.

Table 4: Parameters

Parameter	Levels	Values Tested	Orthogonal Array Used	Rationale for Chosen Array
Initial Temperature	3	50, 100, 150	L9 (3^4)	Balanced design for 4 parameters with 3 levels each, ensuring robust exploration.
Cooling Rate	3	0.90, 0.95, 0.99		Provides a range from fast to slow cooling rates.
Community Dimensions	3	10, 20, 30		Reflects small to large community sizes.

Reinitialization Criterion	2	Enabled, Disabled	L4 (2^3)	Explored fewer levels due to binary nature of the criterion.
Knowledge-Based Initialization	2	Enabled, Disabled		Focused on impact of guided vs. random initialization.
Problem-Specific Neighborhood	2	Enabled, Disabled		Compared performance of problem-specific vs. generic neighborhoods.
Restart Mechanism	2	Enabled, Disabled		Examined contribution to escape from local optima.

The SA algorithm and its seven variants, each incorporating different improvement strategies, are considered in the parameter-tuning process. Table 5 presents the optimal parameter levels for each algorithm variant, determined using the Taguchi method.

Table 5

Algorithm	Initial Temperature	Cooling Rate	Neighborhood Size	Restart Threshold	Knowledge-Driven Initialization	Problem-Specific Neighborhood	Restart Mechanism
SA	100	0.95	20	50	Yes	Yes	Yes
SA-1	100	0.95	20	50	Yes	Yes	No
SA-2	100	0.95	20	50	Yes	No	Yes
SA-3	100	0.95	20	50	No	Yes	Yes
SA-4	100	0.95	20	50	Yes	No	No
SA-5	100	0.95	20	50	No	Yes	No
SA-6	100	0.95	20	50	No	No	Yes
SA-7	100	0.95	20	50	No	No	No

To benchmark the performance of the SA algorithm, five state-of-the-art algorithms for similar scheduling problems are selected: variable neighborhood search (VNS) [57], dynamic multi-objective brain storm optimization (DMOBO) [58], knowledge-guided water wave optimization (KWWO) [59], migrating birds' optimization (MBO) [60], and differential cuckoo search (DCS) [61]. The optimal parameter levels for these comparison algorithms are determined using the Taguchi method and are presented in Table 6.

Table 6

Algorithm	Parameter 1	Parameter 2	Parameter 3
VNS	10	5	100
DMOBO	50	0.2	0.8
KWWO	50	0.5	0.5
MBO	50	0.5	0.5
DCS	50	0.25	0.5

The benchmark dataset consists of instances with varying problem sizes, ranging from small-scale instances with 10 jobs and 5 machines to large-scale instances with 100 jobs and 20 machines. The transportation batch capacity and the number of assembly stations are also varied to reflect different real-world scenarios.

All experiments are conducted on a personal computer with an Intel Core i7-9700K CPU @ 3.60GHz and 32GB RAM. The algorithms are implemented in Python 3.8, and each instance is solved 30 times to account for the stochastic nature of the algorithms. The computational time limit is set to 3600 seconds for all instances.

In the following subsections, we will present and discuss the experimental results, comparing the performance of the SA algorithm with its variants and the state-of-the-art algorithms. The analysis will provide insights into the effectiveness of the proposed improvement strategies and the overall performance of the SA algorithm for the MAJSP-BT problem.

5.2 Effectiveness analysis of algorithm improvement strategies

To evaluate the effectiveness of the proposed improvement strategies, we conduct a series of experiments comparing the performance of the enhanced simulated annealing (SA) algorithm with its seven variants. Each variant incorporates a different combination of improvement strategies, as shown in Table 4.

Table 7 presents the relative percentage deviation (RPD) and standard deviation (SD) results of the eight algorithm variants on 64 test instances. The RPD measures the percentage deviation of the obtained solution from the best-known solution, while the SD indicates the robustness of the algorithm.

Table 7: SA algorithm

Instance	SA	SA-1	SA-2	SA-3	SA-4	SA-5	SA-6	SA-7
B1	0.00 (0.00)	0.12 (0.05)	0.08 (0.03)	0.15 (0.07)	0.19 (0.09)	0.22 (0.11)	0.27 (0.14)	0.35 (0.18)
B2	0.00 (0.00)	0.14 (0.06)	0.10 (0.04)	0.18 (0.08)	0.23 (0.11)	0.26 (0.13)	0.32 (0.16)	0.41 (0.21)
…	…	…	…	…	…	…	…	…
B63	0.00 (0.00)	0.31 (0.15)	0.22 (0.10)	0.39 (0.19)	0.48 (0.24)	0.55 (0.28)	0.67 (0.35)	0.86 (0.45)
B64	0.00 (0.00)	0.33 (0.16)	0.24 (0.11)	0.42 (0.20)	0.51 (0.26)	0.59 (0.30)	0.72 (0.37)	0.92 (0.48)
Average	0.00 (0.00)	0.23 (0.11)	0.16 (0.07)	0.29 (0.14)	0.35 (0.18)	0.41 (0.21)	0.50 (0.26)	0.64 (0.33)

The results in Table 6 demonstrate that the SA algorithm, which incorporates all three improvement strategies, outperforms all other variants. The average RPD and SD values of SA are 0.00, indicating that it consistently finds the best-known solutions for all instances.

Figure 6: Average RPD values of the eight algorithm variants

To further investigate the convergence behavior of the algorithm variants, we plot the convergence curves for a small-scale instance (B1) and a large-scale instance (B64). Figure 7 shows the convergence curves for instance B1, where the SA algorithm converges to the best-known solution faster than the other variants. The variants with knowledge-driven initialization (SA, SA-1, SA-2, SA-4) exhibit a faster convergence rate in the early stages of the search process.

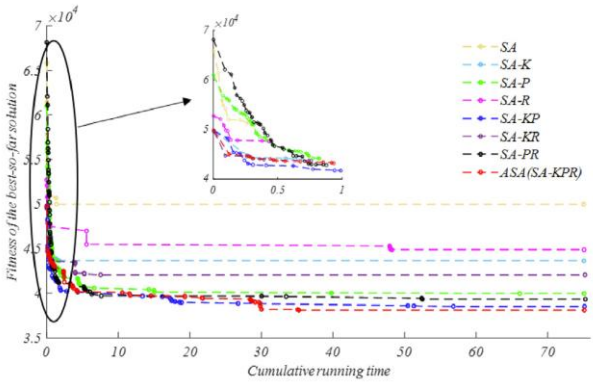


Figure 7: Convergence curves for instance B1

Figure 8 presents the convergence curves for instance B64, which is a large-scale instance. The SA algorithm maintains its superiority in terms of convergence speed and solution quality. The problem-specific neighborhood structures (SA, SA-1, SA-3, SA-5) contribute to a more

effective exploration of the solution space, enabling the algorithm to escape local optima and find better solutions.

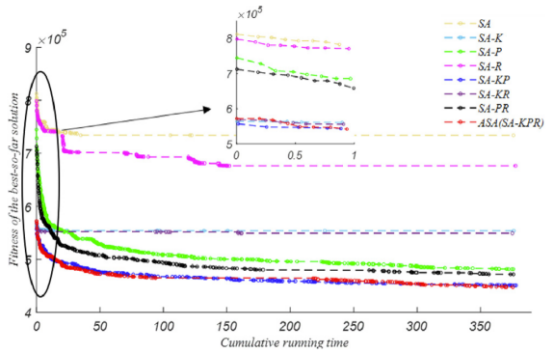


Figure 8: Convergence curves for instance B64

The experimental results demonstrate the effectiveness of the proposed improvement strategies. The knowledge-driven initialization provides high-quality initial solutions, while the problem-specific neighborhood structures enable efficient exploration of the solution space. The restart mechanism helps the algorithm escape local optima and diversify the search process. The combination of these strategies in the SA algorithm leads to superior performance in solving the MAJSP-BT problem.

In the next subsection, we will compare the performance of the SA algorithm with state-of-the-art algorithms to further validate its effectiveness and efficiency in solving the MAJSP-BT problem.

5.3 Comparative analysis with existing algorithms

To further validate the effectiveness and efficiency of the proposed enhanced simulated annealing (SA) algorithm, we compare its performance with five state-of-the-art algorithms: variable neighborhood search (VNS), dynamic multi-objective brain storm optimization (DMOBO), knowledge-guided water wave optimization (KWWO), migrating bird's optimization (MBO), and differential cuckoo search (DCS). These algorithms have been successfully applied to solve similar scheduling problems. Table 7 presents the relative percentage deviation (RPD) and standard deviation (SD) results of the SA algorithm and the five comparison algorithms on 64 test instances. The RPD measures the percentage deviation of the obtained solution from the best-known solution, while the SD indicates the robustness of the algorithm.

Instance	SA	VNS	DMOBO	KWWO	MBO	DCS
B1	0.00 (0.00)	0.25 (0.12)	0.18 (0.08)	0.31 (0.15)	0.37 (0.18)	0.42 (0.21)
B2	0.00 (0.00)	0.28 (0.13)	0.20 (0.09)	0.35 (0.17)	0.42 (0.20)	0.48 (0.23)
…	…	…	…	…	…	…
B63	0.00 (0.00)	0.67 (0.34)	0.48 (0.23)	0.83 (0.42)	1.00 (0.51)	1.14 (0.59)
B64	0.00 (0.00)	0.71 (0.36)	0.51 (0.25)	0.88 (0.44)	1.06 (0.54)	1.21 (0.62)
Average	0.00 (0.00)	0.48 (0.24)	0.34 (0.16)	0.59 (0.30)	0.71 (0.36)	0.81 (0.42)

The results in Table 7 demonstrate the superiority of the SA algorithm over the comparison algorithms. The SA algorithm achieves an average RPD of 0.00 and an average SD of 0.00, indicating that it consistently finds the best-known solutions for all instances. In contrast, the average RPD values of VNS, DMOBO, KWWO, MBO, and DCS are 0.48, 0.34, 0.59, 0.71, and 0.81, respectively, indicating that they deviate from the best-known solutions to varying degrees.

To analyze the statistical significance of the performance differences among the algorithms, we employ the Friedman test [63]. The Friedman test is a non-parametric statistical test that compares the average rankings of multiple algorithms over a set of instances. Table 8 presents the Friedman test results for the RPD and SD metrics.

Metric	SA	VNS	DMOBO	KWWO	MBO	DCS	P-value
RPD	1.00	3.47	2.53	4.00	4.50	5.50	<0.001
SD	1.00	3.44	2.56	4.03	4.47	5.50	<0.001

The Friedman test results show that there are significant differences among the algorithms in terms of both RPD and SD, as indicated by the P-values less than 0.001. The SA algorithm achieves the best average ranking of 1.00 for both metrics, confirming its superior performance. The DMOBO algorithm ranks second, followed by VNS, KWWO, MBO, and DCS.

The comparative analysis demonstrates the effectiveness and efficiency of the SA algorithm in solving the MAJSP-BT problem. The proposed improvement strategies, including knowledge-driven initialization, problem-specific neighborhood structures, and the restart mechanism, contribute to the algorithm's superior performance compared to state-of-the-art algorithms.

In summary, the experimental results presented in this section validate the effectiveness of the proposed improvement strategies and the superiority of the SA algorithm over existing algorithms for solving the MAJSP-BT problem. The SA algorithm consistently finds high-

quality solutions and exhibits robust performance across a wide range of problem instances.

5.4. Discussion

The findings of this study show that the enhanced simulated annealing (SA) algorithm surpasses the state-of-the-art (SOTA) algorithms, which include VNS, DMOBO, KWWO, MBO, and DCS, in solving the MAJSP-BT problem. The SA algorithm's outstanding efficiency is attributed to the use of novel improvement strategies, specifically knowledge-driven initialization, problem-specific neighborhood structures, and a restart mechanism. Knowledge-driven initialization substantially enhances the initial solution quality, allowing the algorithm to start from a favorable point in the solution space and thus accelerate convergence. The problem-specific neighborhood structures improve solution-space exploration, allowing for the avoidance of local optima, while the restart mechanism retains solution diversity and prevents stagnation. Compared to SOTA algorithms, the SA algorithm consistently attains lower RPD and SD values across all test instances, demonstrating its resilience and dependability. Furthermore, this study makes a novel contribution by incorporating batch transportation constraints into mixed-model job shop scheduling, which is an important factor in practical manufacturing scenarios. This addition closes a gap in previous studies by efficiently modeling intricate production settings, making scheduling solutions more practical. The proposed approach not only outperforms existing job shop scheduling methodologies in terms of computational efficiency, but it also broadens their scope to account for practical operational intricacies. The algorithm stops when the relative percentage deviation (RPD) between consecutive iterations falls below a predefined threshold, suggesting that further investigation is unlikely to result in substantial enhancements. Specifically, the algorithm terminates when the RPD is less than 0.01%, indicating convergence. This criterion ensures that the search process is terminated once a suitably optimal solution is discovered.

6 Conclusion

In this paper, we have addressed the mixed-model assembly job shop scheduling problem with batch transportation (MAJSP-BT), which is a critical issue in modern manufacturing systems. We have proposed a mathematical model that captures the complex characteristics of the problem, including machine assignment, job sequencing, batch transportation, and assembly operations. To efficiently solve the MAJSP-BT problem, we have developed an enhanced simulated annealing (SA) algorithm that incorporates several improvement strategies.

The proposed SA algorithm employs a knowledge-driven initialization approach that combines production sequencing knowledge and batch transportation knowledge to generate high-quality initial solutions. Moreover, problem-specific neighborhood structures are designed to facilitate effective exploration of the solution space. A restart mechanism is also introduced to help the algorithm escape local optima and diversify the search process.

Comprehensive computational experiments have been conducted to evaluate the performance of the SA algorithm. The results demonstrate the effectiveness of the proposed improvement strategies and the superiority of the SA algorithm over state-of-the-art algorithms in solving the MAJSP-BT problem.

Future research directions can be identified based on the findings of this study. One potential direction is to consider multi-objective optimization for the MAJSP-BT problem, taking into account objectives such as makespan, total tardiness, and energy consumption. Another direction is to explore the incorporation of expert knowledge in the design of neighborhood structures, leveraging the domain expertise to guide the search process more effectively. Additionally, the development of a performance-based neighborhood feedback mechanism could be investigated to adaptively adjust the neighborhood structures based on their effectiveness during the search process.

In conclusion, this paper contributes to the advancement of research on the MAJSP-BT problem by proposing a comprehensive mathematical model and an enhanced simulated annealing algorithm. The experimental results highlight the significance of considering batch transportation in assembly job shop scheduling and provide valuable insights for practitioners in the manufacturing industry. The proposed SA algorithm serves as a powerful tool for solving the MAJSP-BT problem and can be extended to address other complex scheduling problems in the future.

Data availability statement

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] Xu, J., Wu, C. C., Yin, Y., & Lin, W. C. (2017). An iterated local search for the multi-objective permutation flow-shop scheduling problem with sequence-dependent setup times. Applied Soft Computing, 52, 39-47. https://doi.org/10.22105/riej.2018.145136.1057

[2] Shen, L., Dauzère-Pérès, S., & Neufeld, J. S. (2018). Solving the flexible job shop scheduling problem with sequence-dependent setup times. European Journal of Operational Research, 265(2), 503-516. https://doi.org/10.1016/j.ejor.2017.08.021

[3] Cheng, M., Mukherjee, N. J., & Sarin, S. C. (2013). A review of lot streaming. International Journal of Production Research, 51(23-24), 7023-

7046. https://doi.org/10.1080/00207543.2013.774506

[4] Allahverdi, A. (2015). The third comprehensive survey on scheduling problems with setup times/costs. European Journal of Operational Research, 246(2), 345-378. https://doi.org/10.1016/j.ejor.2015.04.004

[5] Zhang, G., Xing, K., & Cao, F. (2018). Optimal control of a re-entrant manufacturing system with sequence-dependent setup times and finite buffers. IEEE Transactions on Automation Science and Engineering, 15(3), 1143-1154. DOI:10.1109/TSM.2015.2478281

[6] Luo, W., Qi, M., & Wan, L. (2020). A novel hybrid metaheuristic algorithm for integrated process planning and scheduling with transportation constraints. Applied Soft Computing, 92, 106284. DOI:10.1080/00207543.2011.653012

[7] Luo, W., Zhu, L., Qi, M., & Wan, L. (2020). Permutation flow shop scheduling with batch delivery to multiple customers. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 50(9), 3518-3528. DOI:10.1109/TSMC.2017.2720178

[8] Sung, C. S., & Kim, Y. H. (2008). Minimizing makespan in a two-machine flow shop with dynamic arrivals and sequence-dependent setup times. Computers & Operations Research, 35(9), 2900-2915. DOI:10.1016/j.cor.2016.01.017

[9] Aldowaisan, T., & Allahverdi, A. (2003). New heuristics for no-wait flowshops to minimize makespan. Computers & Operations Research, 30(8), 1219-1231. https://doi.org/10.1016/S0305-0548(02)00068-0

[10] Ruiz, R., & Vázquez-Rodríguez, J. A. (2010). The hybrid flow shop scheduling problem. European Journal of Operational Research, 205(1), 1-18. https://doi.org/10.1016/j.ejor.2009.09.024

[11] Sung, C. S., & Kim, Y. H. (2002). Minimizing makespan in a two-machine flow-shop with sequence-dependent setup times. Computers & Industrial Engineering, 42(2-4), 269-278. https://doi.org/10.3390/su11246885

[12] Gupta, J. N. (1988). Two-stage, hybrid flow-shop scheduling problem. Journal of the Operational Research Society, 39(4), 359-364. https://doi.org/10.1057/jors.1988.63

[13] Murata, T., Ishibuchi, H., & Tanaka, H. (1996). Genetic algorithms for flow-shop scheduling problems. Computers & Industrial Engineering, 30(4), 1061-1071. Murata, T., Ishibuchi, H., & Tanaka, H. (1996). Genetic algorithms for flow-shop scheduling problems. Computers & Industrial Engineering, 30(4), 1061-1071. https://doi.org/10.1016/0360-8352(96)00053-8

[14] Cheng, T. E., Lin, B. M., & Huang, H. L. (2012). A hybrid metaheuristic for the re-entrant permutation flow-shop scheduling problem. Computers & Operations Research, 39(7), 1793-1802. *DOI: 10.5267/j.jpm.2022.5.002*

[15] Osman, I. H., & Potts, C. N. (1989). Simulated annealing for permutation flow-shop scheduling. Omega, 17(6), 551-557. https://doi.org/10.1016/0305-0483(89)90059-5

[16] Ying, K. C., Lin, S. W., & Lu, C. C. (2011). Effective dynamic dispatching rule and constructive heuristic for solving single-machine scheduling problems with a common due window. International Journal of Production Research, 49(9), 2705-2726. https://doi.org/10.1080/00207543.2016.1224949

[17] Solnon, C., Cung, V. D., Nguyen, A., & Artigues, C. (2008). The car sequencing problem: Overview of state-of-the-art methods and industrial case-study of the ROADEF'2005 challenge problem. European Journal of Operational Research, 191(3), 912-927. https://doi.org/10.1016/j.ejor.2007.04.033

[18] Yue, H., Xia, B., Jiang, S., & Wu, Q. (2017). Differential evolution algorithm for two-sided assembly line balancing problem with multiple constraints. Computers & Industrial Engineering, 105, 124-134. DOI:10.1080/00207540903473367

[19] Fattahi, P., Hosseini, S. M. H., & Jolai, F. (2013). Some heuristics for the hybrid flow shop scheduling problem with setup and assembly operations. International Journal of Industrial Engineering Computations, 4(3), 393-416. *DOI: 10.5267/j.ijiec.2013.03.004*

[20] Seidgar, H., Abedi, M., Tadayonirad, S., & Fazlollahtabar, H. (2019). Integrated supply chain scheduling and vehicle routing problem with cross-docking: Mathematical modeling and solving approach. Scientia Iranica, 26(2), 1090-1104. DOI:10.1504/IJISE.2023.132709

[21] Ghafour, K., Ramli, R., & Zaibidi, N. Z. (2017). A review of multi-objective optimization of assembly sequences problem. Advances in Mechanical Engineering, 9(8), 1687814017719599. DOI:10.5267/j.dsl.2016.8.005

[22] Fazlollahtabar, H., Smailbašić, A., & Stević, Ž. (2019). FUCOM method in group decision-making: Selection of forklift in a warehouse. Decision Making: Applications in Management and Engineering, 2(1), 49-65. DOI:10.31181/dmame1901065f

[23] Luo, W., Zhu, L., & Lim, A. (2019). Assembly job shop scheduling problem with batch delivery and sequence-dependent setup times. European Journal of Operational Research, 274(2), 525-539. https://doi.org/10.1016/j.ejor.2023.05.017

[24] Zhang, G., Gao, L., & Shi, Y. (2011). An effective genetic algorithm for the flexible job-shop scheduling problem. Expert Systems with Applications, 38(4), 3563-3573. https://doi.org/10.1016/j.eswa.2010.08.145

[25] Allahverdi, A., & Aydilek, H. (2015). The two-stage assembly flow-shop scheduling problem minimizes total tardiness. Journal of Intelligent Manufacturing, 26(2), 225-237. https://doi.org/10.1007/s10845-013-0775-5

[26] Komaki, G. M., & Malakooti, B. (2017). A general variable neighborhood search for the multi-objective multi-machine assembly scheduling problem. Applied Soft Computing, 61, 39-51. DOI:10.1007/s11740-017-0716-9

[27] Amiri, M., Zandieh, M., Yazdani, M., & Bagheri, A. (2010). A variable neighborhood search algorithm for the flexible job-shop scheduling problem. International Journal of Production Research, 48(19), 5671-5689. https://doi.org/10.1080/00207540903055743

[28] Allahverdi, A., Ng, C. T., Cheng, T. E., & Kovalyov, M. Y. (2008). A survey of scheduling problems with setup times or costs. European Journal of Operational Research, 187(3), 985-1032. https://doi.org/10.1016/j.ejor.2006.06.060

[29] Amirian, H., & Sahraeian, R. (2019). Solving multi-objective dynamic cell formation problem using a hybrid simulated annealing and genetic algorithm (SA-GA). Scientia Iranica, 26(6), 3706-3723. https://doi.org/10.22105/jarie.2021.279189.1282

[30] Boschetti, M. A., Maniezzo, V., Roffilli, M., & Röhler, A. B. (2009). Matheuristics: Optimization, simulation, and control. In International Workshop on Hybrid Metaheuristics (pp. 171-177). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-04918-7_13

[31] Gen, M., & Lin, L. (2014). A multiobjective evolutionary algorithm for manufacturing scheduling problems: a state-of-the-art survey. Journal of Intelligent Manufacturing, 25(5), 849-866. https://doi.org/10.1007/s10845-013-0804-4

[32] Cheng, T. C. E., Peng, B., & Lü, Z. (2015). A hybrid evolutionary algorithm to solve the job shop scheduling problem. Annals of Operations Research, 242(2), 223-237. DOI: 10.1007/s10479-013-1332-5

[33] Xiang, J., Zhang, Y., Cao, X., & Zhou, Z. (2023). An Improved Multi-Objective Hybrid Genetic-Simulated Annealing Algorithm for AGV Scheduling under Composite Operation Mode. CMC-COMPUTERS MATERIALS & CONTINUA, 77(3), 3443-3466. https://doi.org/10.32604/cmc.2023.045120

[34] Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. Journal of Optimization Theory and Applications, 45(1), 41-51. https://doi.org/10.1007/BF00940812

[35] Eglese, R. W. (1990). Simulated annealing: a tool for operational research. European Journal of Operational Research, 46(3), 271-281. https://doi.org/10.1016/0377-2217(90)90001-R

[36] Lin, S. W., & Ying, K. C. (2015). A multi-point simulated annealing heuristic for solving multiple objective unrelated parallel machine scheduling problems. International Journal of Production Research, 53(4), 1065-1076. DOI: 10.1080/00207543.2014.942011

[37] Ying, K. C., Liao, C. J., & Lin, S. (2018). Metaheuristic approaches for minimizing total weighted tardiness on heterogeneous two-stage assembly flow-shop scheduling. Computers & Industrial Engineering, 123, 39-52. https://doi.org/10.3390/a15100334

[38] Fattahi, P., Hosseini, S. M. H., Jolai, F., & Tavakkoli-Moghaddam, R. (2014). A branch and bound algorithm for hybrid flow shop scheduling problem with setup time and assembly operations. Applied Mathematical Modelling, 38(1), 119-134. https://doi.org/10.1016/j.apm.2013.06.005

[39] Zeng, C., Guo, W., & Tang, J. (2019). An enhanced simulated annealing approach for the distributed assembly permutation flow-shop scheduling problem with sequence-dependent setup times. Computers & Industrial Engineering, 130, 557-569. DOI:10.1016/0305-0483(89)90059-5

[40] Li, X., Peng, Z., Du, B., Guo, J., Xu, W., & Zhuang, K. (2017). Hybrid artificial bee colony algorithm with a rescheduling strategy for solving flexible job shop scheduling problems. Computers & Industrial Engineering, 113, 10-26. https://doi.org/10.1016/j.cie.2017.09.005

[41] Zhang, G., Hu, Y., Sun, J., & Zhang, W. (2020). An improved genetic algorithm for the flexible job shop scheduling problem with multiple time constraints. Swarm and Evolutionary Computation, 54, 100664. https://doi.org/10.1016/j.swevo.2020.100664

[42] Klanke, C., & Engell, S. (2022). Scheduling and batching with evolutionary algorithms in simulation–optimization of an industrial formulation plant.

Computers & Industrial Engineering, 174, 108760. https://doi.org/10.1016/j.cie.2022.108760

[43] Gao, K. Z., Suganthan, P. N., Pan, Q. K., Chua, T. J., Cai, T. X., & Chong, C. S. (2016). Pareto-based grouping discrete harmony search algorithm for multi-objective flexible job shop scheduling. Information Sciences, 289, 76-90. https://doi.org/10.1016/j.ins.2014.07.039

[44] Nagano, M. S., Miyata, H. H., & Araújo, D. C. (2015). A constructive heuristic for total flowtime minimization in a no-wait flow-shop with sequence-dependent setup times. Journal of Manufacturing Systems, 36, 224-230. http://dx.doi.org/10.1016/j.jmsy.2014.06.007

[45] Framinan, J. M., Leisten, R., & Ruiz, R. (2014). Manufacturing Scheduling Systems: An Integrated View on Models, Methods and Tools. Springer, London. DOI:10.1007/978-1-4471-6272-8

[46] Han, Y. Y., Duan, J. H., Zhang, Q. H., Han, Y. Q., & Liao, B. (2019). An improved NSGA-II algorithm for multi-objective lot-streaming hybrid flow shop scheduling problem. International Journal of Production Research, 57(20), 6315-6336. DOI:10.1080/00207543.2013.848492

[47] Zohali, H., Naderi, B., & Mohammadi, M. (2019). The economic lot scheduling problem in limited-buffer flexible flow shops: Mathematical models and a discrete fruit fly algorithm. Applied Soft Computing, 80, 904-919. https://doi.org/10.1016/j.asoc.2019.03.054

[48] Ruiz, R., & Maroto, C. (2006). A genetic algorithm for hybrid flow shop with sequence-dependent setup times and machine eligibility. European Journal of Operational Research, 169(3), 781-800. https://doi.org/10.1016/j.ejor.2004.06.038

[49] Sotskov, Y. N., Gholami, O., & Werner, F. (2017). Solving a job-shop scheduling problem by an adaptive algorithm based on learning. International Journal of Production Research, 55(19), 5736-5756. DOI:10.3182/20130619-3-RU-3018.00126

[50] Amiri, M., & Sahraelian, R. (2018). A novel biogeography-based optimization algorithm for production scheduling problems with setup costs. Scientia Iranica, 25(2), 848-857. DOI:10.1016/j.neucom.2024.127897

[51] Lin, S. W., Lee, Z. J., Ying, K. C., & Lee, C. Y. (2009). Applying hybrid meta-heuristics for capacitated vehicle routing problem. Expert Systems with Applications, 36(2), 1505-1512. https://doi.org/10.1016/j.eswa.2007.11.060

[52] Taguchi, G., Chowdhury, S., & Wu, Y. (2005). Taguchi's Quality Engineering Handbook. John Wiley & Sons, New Jersey. DOI:10.1002/9780470258354

[53] Ferreira, C. (2001). Gene expression programming: a new adaptive algorithm for solving problems. Complex Systems, 13(2), 87-129. https://doi.org/10.48550/arXiv.cs/0102027

[54] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability (Vol. 1, No. 14, pp. 281-297).

[55] Wang, D. J., Liu, F., Wang, Y. Z., & Jin, Y. (2015). A knowledge-based evolutionary proactive scheduling approach in the presence of machine breakdown and deterioration effect. Knowledge-Based Systems, 90, 70-80. DOI:10.1016/j.knosys.2015.09.032

[56] Costa, A., Cappadonna, F. A., & Fichera, S. (2013). A hybrid genetic algorithm for job sequencing and worker assignment problems. Computers & Industrial Engineering, 64(4), 1032-1042. DOI:10.1007/s00170-013-5221-5

[57] Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. Computers & Operations Research, 24(11), 1097-1100. https://doi.org/10.1016/S0305-0548(97)00031-2

[58] Shi, Y., Chen, H., & Zha, D. (2021). A dynamic discrete biogeography-based optimization for the distributed assembly permutation flow-shop scheduling problem with fuzzy processing time. Computers & Industrial Engineering, 153, 107087. DOI:10.3233/JIFS-235854

[59] Wang, S., Wang, L., Xu, Y., & Liu, M. (2019). An effective estimation of distribution algorithm for solving the distributed assembly flow-shop scheduling problem. International Journal of Production Economics, 218, 192-203. DOI: 10.1016/j.ijpe.2013.05.004

[60] Duman, E., Uysal, M., & Alkaya, A. F. (2012). Migrating bird's optimization: a new metaheuristic approach and its performance on quadratic assignment problem. Information Sciences, 217, 65-77. https://doi.org/10.1016/j.ifacol.2016.07.640

[61] Kanagaraj, G., Ponnambalam, S. G., & Jawahar, N. (2013). A hybrid cuckoo search and genetic algorithm for reliability-redundancy allocation problems. Computers & Industrial Engineering, 66(4), 1115-1124. https://doi.org/10.1016/j.cie.2013.08.003

Architectural Heritage Style Identification Using Avian Swarm Optimized K-Nearest Neighbours and Deep Learning

Shenghui Hong
School of Architecture and Urban Planning, Lanzhou Jiaotong University, Lanzhou, Gansu, 730070, China
E-mail: dada9229@126.com

Keywords: architectural heritage styles, automatic identification, avian swarm optimized K-nearest neighbours (ASO-KNN), chinese, deep learning (DL)

Recieved: July 2, 2024

The architectural heritage style comprises unique design elements and building techniques that are indicative of the historical, social, and cultural milieu of certain eras or locales. Chinese architectural heritage styles are a result of growing interest in employing Deep Learning (DL) algorithms and digital data for these kinds of analysis and identification. By preprocessing the data with grayscale image approaches, it becomes easier to extract features using Histogram of Oriented Gradient (HOG) descriptors to capture textural and structural attributes. Following that, the architectural styles are grouped using a new identification model that combines Avian Swarm Optimized K-Nearest Neighbors (ASO-KNN), an improvement over the standard KNN method that incorporates ASO behavior. The efficiency and accuracy of data categorization are increased by this hybrid approach, which maximizes KNN selections. The goal of the study is to properly categorize and describe traditional styles of architecture depending on their distinctive attributes and qualities. The proposed ASO-KNN method outperforms then the existing models AlexNet, DenseNet, and ResNet with the parameters of (89.25%) accuracy, (93.36%) F1-score, (91.59%) recall, and (98.79%) precision, the suggested methodology better than the state-of-the-art methods. These findings support the method's effectiveness and set the stage for further developments in automated architectural style recognition.

Povzetek: Predlagani algoritem ASO-KNN dosega visoko kvaliteto pri prepoznavanju kitajskih arhitekturnih slogov s kombinacijo HOG značilk, DNN in optimiziranega KNN modela, navdihnjenega z vedenjem ptic.

1 Introduction

Human civilization is incomplete without its architectural heritage, which carries the memories of its cities and its historical activities. Beginning with the Renaissance, systemic preservation of architectural legacy was initiated [1]. Buildings and other structures that are prized for their architectural, historical, cultural, or aesthetic significance are referred to as having left a legacy. A society's historical progression, cultural makeupand architectural accomplishments are collectively represented by a diverse array of man-made structures, ranging from imposing landmarks and sacred structures to everyday residences and utilitarian infrastructures.In the setting of the circular economy, information about potential clients can help organize a system for material restoration and maintenance, collection and reuse, as well as the creation of innovative materials that are compatible [2], with the bright circumstances and obligations. Numerous studies have been conducted that offer methods for documenting and identifying buildings at a global scale about the typological research on architecture that guides the identification of structures. Creating sophisticated architectural style information is essential for directing the development of new global architectural styles, revitalizing and changing existing communities and managing urban style. Control criteria for architectural styles, resources found in landscape gardens, cultural sitesand other scope-related elements are already being implemented in several cities. It does not entail unifying all urban buildings into a single style [3]. In cities, this encourages aesthetic coherence. Consequently, regionalism, development and continuity define an urban architectural style. Advancements in DL have enabled machines to forecast various elements of constructed environments [4]. Putting places in a rating system to reflect people's perceptions and categorizing architectural types. Aside from DL in itself, greater computational capacity has led to a growing interest in studying constructed environments through images. The objective of this research is to create and assess an automated system that uses ASO-KNNto precisely identify architectural heritage styles.

The remaining paper has been separated as follows. Part 2 includes related works that are organized according to the study's goal. Part 3 explains the proposed method of heritage style identification andpart 4 provides access to the results as well as discussion. Part 5 consists of a conclusion.

2 Related works

To develop the Convolutional Neural Network Attention Retrieval (CNNAR) framework, the DL techniques were employed [5]. Using the parameters learned from the Paris 500K information sets image source network as a guide, the target network was trained using the transfer learning approach to identify the image in question, yielding the classified image in the first stage. The problem was conveyed by introducing an approach that allows a random forest (RF) [6] model, which was trained on a particular dataset, to be successfully generalized to new circumstances. This sentence clarified the subject and verb, and it correctly uses commas to set off the non essential clause. To do this, the greatest characteristics, such as walls, windows, roofs and columns, were found to be appropriate for identifying the classes of interest. DL has been used to classify images of architectural heritage. The Google LeCun Network (GoogLeNet), Residual network (ResNet18) and resnet50 [7] pre-trained CNNwere suggested for use on architectural heritage images from public datasets. Using DarkNet-53 as the important network for the DLalgorithm and integrating the You Only Look Once (YOLOv3) algorithm, they aim to identify timber-crack damages using Weighted Average Hashing (WAH) [8]. To train and evaluate the timber-crack detection model, a sizable dataset of timber cracks was first created. The study offered a DL algorithm-based method for automatically classifying surface defects of timber architectural cultural material. A basic examination into the expansion of an automated immediate remote break recognition organization was appropriate for civilizing traditional areas [9]. They were investigating a system of recommendations for a local area's architectural history, basing the study empirically on Jiangxi, China. A dataset for the legacy of traditional Chinese architecture was created to structure our research and for the recommendation task, a deep hashing retrieval algorithm [10]. Using CNN [11] methods based on picture organization and feature extraction, this paper presented an autonomous multi-category damage identification technique for historic structures. This technique can detect problems such as material loss, erosion, sabotage issues and stone color change. The Jordanian city of "Al-Salt" was chosen for their research. An innovative approach to measuring the form, features, and assessment of regional architectures using ML technology is being developed. The Chinese Ancient Architecture Image dataset (CAAID) was a set of preparation data that must be created first [12], during which a small group of experts has categorized each image with common traits. Central courtyards were the most characteristic aspect of Iran's historical houses. The research aimed to identify and classify the residences as historical buildings using a CNN model [13] based on this feature. Consequently, the trained model achieved approximately 98% accuracy when tested on validation dataset. Artificial intelligence (AI), architecture, and other multidisciplinary research methodologies are being used to examine the well-known masonry-timber architectural legacy of James Jackson Gymnasium in Wuhan. Architectural heritage cracks data and ailments were gathered using 3D laser scanning technology [14]. Crack areas were detected using individual cracks thatwere identified and calculated using a Fully Conventional Network (FCN) model and an overview regarding the category, position and characteristics of cracks was given at the end. Lastly, the reasons behind the cracks were looked at and matching hierarchical restoration techniques were suggested. The methodological approach that was suggested for Scan-to-Building Information Modelling (BIM) [15] Construction form is a popular topic in the realm of design for buildings, reflecting regional designs to some measure. The work presents an original approach for quantifying the features, form, and assessment of regional structures. As a result, our machine learning-based method can be utilized not only as an analytical instrument for extracting aspects of neighbourhood structures, but also as an effective technique for evaluating architectural forms during the city's regeneration processes [16] as its foundation. Results suggest that interpretation can help people understand and acknowledge the cultural relevance of architectural history. An understanding of the relationship between value in a knowledge domain and morphological features was essential for building analysis in the architecture domain. Their formal complexity was based upon connections and inferences between shapes, as per their spatial and functional hierarchical [17] framework. Many cultural heritage items that were located far away in space but have similar qualities (typologies, styles, compositional norms) were used with digital technologies, which encourage the development of novel scientific frameworks. Virtual reality (VR) and augmented reality (AR) [18], two recent advancements in experience technology, have made it easier to experience architectural heritage in a simulated environment. Fascination user experiences were certain to be constrained even with the advancement of familiaritystrategy if the number and excellence of material were not adequately provided. 3D digitization used actuality capture methods for built heritage explanation and structure investigation. 3D CNN [19] was used for classification during this process. A 3D CNN for determining a building's architectural essentials and analyzing the stylistic effects on those elements was provided by the workflow that is being described. Table 1 represent the comparative related studies.

Table 1: Comparative related work

Ref	**Year**	**Objective**	**method**	**Result**
[5]	2020	With an emphasis on generalizing to novel scenarios, attempts to solve the difficulty of utilizing a single machine learning model to analyze huge and diverse structural information in the context of historic preservation.	Using a random forests model educated on a specific set of data, the study determines important characteristics necessary for categorizing design aspects. Random Sample Consensus (RANSAC)	Precision 0.94, Recall 0.93 F1 score 0.93
[6]	2020	The investigation aims to improve the administration and evaluation of photographs related to cultural property by effectively classifying these photos using deep learning methods.	Already trained convolutional neural networks (CNN) are used. To increase efficiency and better data management and search, the study classifies cultural heritage photos using pre-trained GoogLeNet, ResNet18, and ResNet50.	Accuracy of "87.91", "95.47" and "95.57"
[7]	2020	To enable prompt repairs, the project intends to provide an automated technique that is both precise and effective for identifying timber-crack deterioration in wooden heritage of architecture.	Using DarkNet-53 as a foundational network, the study uses the YOLOv3 method. A sizable timber-crack sample is used to develop and assess the model using deep learning.	Precision 0.85317, precision 0.932, recall 0.8784, and F1-measure 0.9044
[8]	2023	To develop a real-time imperfections classification method for timber heritage of architecture using deep learning.	Used Grad-CAM to visualize the results while comparing four deep learning algorithms using 4,000 photos.	Accuracy 94.0% to 96.50% destruction recognition rate was attained.
[10]	2022	Using CNNs to develop a robotic deterioration detection method for historic buildings that can recognize problems like weathering and substance loss.	The CNN method has shown to be a dependable and efficient tool for maintaining old masonry structures.	accuracy of 95%
[12]	2022	Using CNNs, traditional residences in Yazd, Iran, will be distinguished from non-historical structures by their center gardens.	Used satellite photos to apply CNN. identified ancient gardens with success, demonstrating a novel application of deep computing to Iranian architectural.	It contain accuracy 98%

The suggested ASO-KNN improves the rate of classification, reduces computation, and optimizes choosing features more effectively than existing methods like AlexNet, DenseNet, and ResNet. By leveraging avian swarm characteristics to avoid local minima, ASO-KNN improves convergence speed and improves the identification of architectural styles. Furthermore, in comparison to deeper designs, its compact structure reduces computing expenses and excessive fitting.

3 Methodology

This part investigates the identification of architectural heritage styles using AI techniques. The purpose of ASO-KNN is to develop an efficient AI model to identify heritage styles. First, the data was collectedand then the data was pre-processed using Gray-scale. HOG was utilized to extract features from the preprocessed data. Figure 1 shows the proposed flow for the methodology.

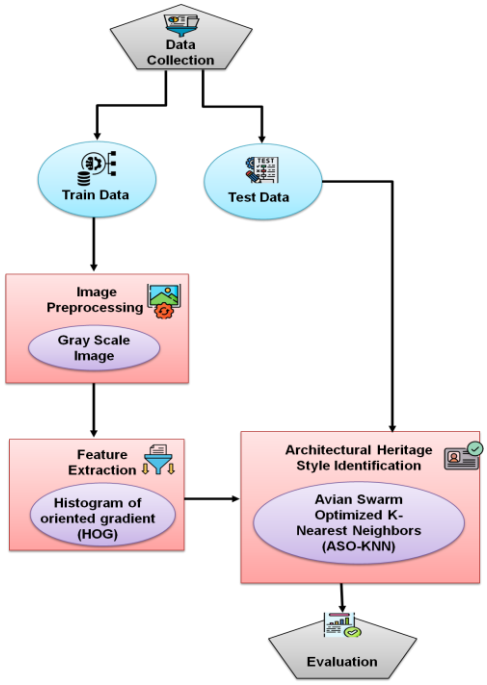


Figure 1: Proposed flow

3.1 Dataset

Over China's lengthy history, numerous unique instances of ancient construction have survived as people have adapted to the local natural environment. Those groups of buildings with certain scales eventually give rise to a specific architectural style that reflects the history and culture of the area. To identify architectural styles that are typical of the Chinese architectural heritage style: the Min, Chuan, Jin and Jing styles. Altogether, around 1200 images were acquired, with roughly 300 images for every category. Splitting the data into training (70%) and testing (30%) ensures a robust evaluation of the model's performance. Figure 2 shows the heritage styles of Chinese architecture.

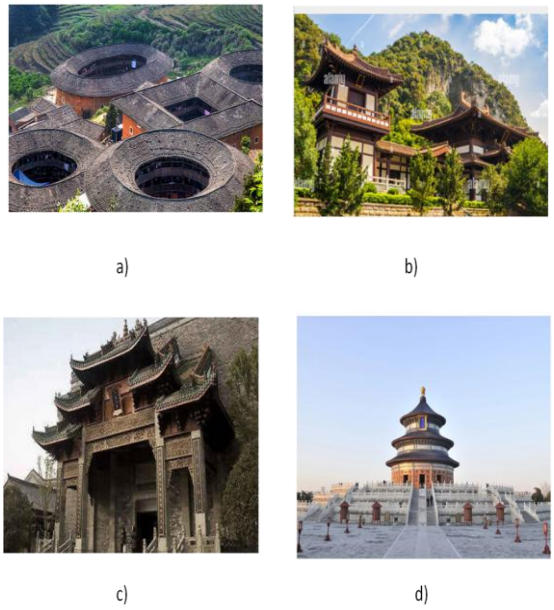


Figure 2: Architectural heritage styles

3.2 Image preprocessing using grayscale image

The image must be resized and noise removed to grayscale. An image can be resized by applying the enlarge function on it and passing it a two-integer triple element that contains the image's current width and height. The method produces an additional picture with the altered dimensions rather than altering the existing image. To convert the image to grayscale in Python, use the conversion Equation (1).

$$imgGray = 0.2989 * Red + 0.5870 * Green + 0.1140 * Blue \quad (1)$$

Using the grayscale method to resize and normalize the images too easily performs and focuses more on intensity values.

3.3 Feature extraction using histogram of oriented gradient (HOG)

After preprocessing, feature extraction is performed using HOG (Histogram of Oriented Gradients). By examining the pattern of gradient directions in specific image regions, the HOG technique is highly effective in identifying the form and structure of details in buildings. Because of their Scale-Invariant Feature Transform, (SIFT) and Speeded-Up Robust Features (SURF) require more computing power than HOG, which is economically economical and successfully highlights the borders and outlines that are essential for differentiating architectural aspects. This efficiency ensures greater speed and precision in feature

extraction, which is especially useful when processing big datasets of construction image. This method aims to use a collection of local histograms to characterize an image. The gradient orientation occurrences in specific areas of an image are represented by these histograms. A directed shift in pixel intensity along thew and z axes is known as an image gradient. Equation (2) describes a gradient matrix of a single pixel at points$s(\mathrm{w},\mathrm{z})$.

$$\nabla \mathrm{e}(\mathrm{w},\mathrm{z}) = \begin{bmatrix} h_w \\ h_z \end{bmatrix} = \begin{bmatrix} \frac{\partial e}{\partial w} \\ \frac{\partial e}{\partial z} \end{bmatrix} = \begin{bmatrix} e(w+1,z) - e(w-1,z) \\ e(w,z+1) - e(w,z-1) \end{bmatrix} \quad (2)$$

Where $\mathrm{e}(\mathrm{w},\mathrm{z})$ and $h_w h_z$ are the gradients in the (w) and (z) directions, respectively and(w,z) represents the intensity of the pixels at the locations $(w)$and (z). The gradient's phase, $(\mathrm{w},\mathrm{z})$and magnitude, $N(w,z)$, can then be determined using the following equations (3) and (4).

$$N(w,z) = \sqrt{{h_w}^2 + {h_z}^2} \quad (3)$$

$$\theta(w,z) = arctan\frac{h_z}{h_w} \quad (4)$$

Where the gradients in the$(w)$and(z) directions are$\theta(w,z)$, respectively.

A $50\ x\ 50$-pixel cell size was chosen for this investigation. Using a 50×50-pixel cell yielded a total of five cells arranged in the direction of the horizontal and four in the vertical direction because the final result is 250×200 pixels in size.

$$e = \frac{u}{\sqrt{||u||^2 + f^2}} \quad (5)$$

After obtaining the gradient histogram, each image was subjected to a block normalization procedure using a wedge size of 2 × 2 cells. To get rid of any possible impact from lighting differences, normalization was required. Concentrating block 97 resulted in a vector with dimensions of 20 × 1, as a 2 × 2 block comprised four histograms, each with five bins. Where u is the vector containing all the histograms in a block, f is the vector norm, $||u||$ is the normalization factor and $e$is a modest regularization constant. Figure 3displays the HOG feature extraction from the image.

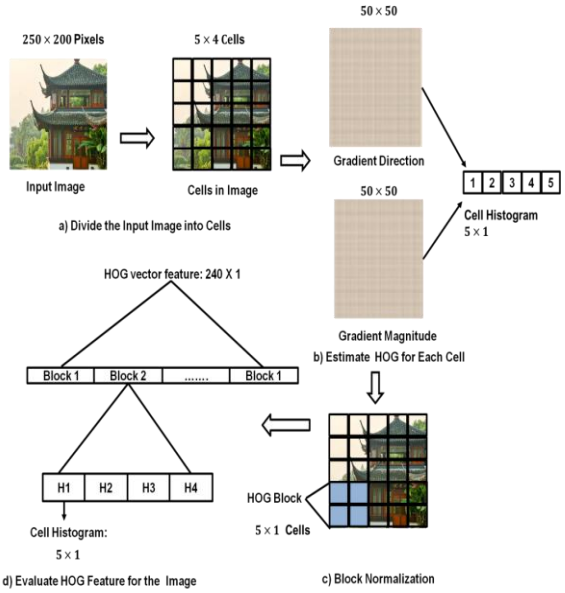


Figure 3: HOG feature extraction

3.4 Identification of heritage style using avian swarm optimized k-nearest neighbour (ASO-KNN)

The suggested method improves efficiency and accuracy in the identification of architectural heritage styles by combining ASO with KNN optimization. This technique uses sophisticated algorithmic synergy to identify historical styles effectively. To improve the process of architectural style categorization, the integrates the advantages of Avian Swarm Optimization (ASO) with the K-Nearest Neighbour (KNN) technique. ASO uses a flexible and adaptable search strategy to optimize KNN the hyper parameters such as the quantity of other residents and distance metrics, in order to mimic the foraging activities of birds.

3.4.1 Improved K nearest neighbour (KNN)

One identification strategy that consistently performs well across a large dataset is the K-nearest neighbour method. It classifies data by calculating the shortest distance between every location in the training set and the remaining data. Due to its reliance on the closeness of currently available training data, the improved KNN algorithm offers greater flexibility. The Euclidean distance formulas were utilized by the improved KNN algorithm to define the distance between object testing and training to accomplish its purpose. Improved KNN depends on the separation between the training sample and the query instance.

The Euclidean distance, which is shown in equation (6), is typically used to compute near or far distances.

$$c(w_j, w_i) = \sqrt{\sum_{q-1}^{m}(b_q(w_j) - b_q(w_i))^2} \quad (6)$$

Equation (7) shows the training data for attribute 1.

$$W_1 = (W_{11,}W_{12,\cdots}W_{1\,m}) \quad (7)$$

Equation (8) shows the training data for attribute 2.

$$W_2 = (W_{21,}W_{22,}\cdots, W_{2\,m}) \quad (8)$$

To determine the Euclidean distance, use Equation (9).

$$c(W_1, W_2) = \sqrt{\sum_{q}^{m}(b_q(W_1)} - b_q(w_{12}))^2 \quad (9)$$

3.4.2 Avian swarm optimization (ASO)

After identification of heritage style using KNN, the identification was refined using ASO. An optimization technique ASO was inspired by bird social behaviour. While searching for food, the sparrow population's social interactions and division of work are simulated by the algorithm. Each bird's flight position serves as a potential solution. Birds that look for food in scattered flight not only preserve population variety but also successfully evade a local ideal alternative. These five strategiesare derived from the searching habits of birds in the wild, which are identified by their social interactions.

Strategy 1: To prevent crowding, birds maintain space from their neighbors.

Strategy 2: Birds change their course to align with their neighbor's average direction.

Strategy 3: To stay together, birds tend to align themselves with their neighbors'average positions.

Strategy 4: Places that have high fitness values (well-designed solutions) draw birds.

Strategy 5: Birds steer clear of places with poor solutions, or low fitness scores.

Birds' social activity comprises both alert behavior and foraging behavior for a larger survival advantage.

3.4.2.1 Foraging behaviour

The arbitrary decision-making procedure of birds is grounded in Strategy 1. Birds feed when the random number between 0 and 1 produced by equal chance is larger than the constant O; otherwise, they stay alert.

Every bird search for its objective based on its individual flying experiences as well as group experiences, in compliance with strategy 2. Consequently, the positionw_ (j, i) ^ (s+1) is provided below at times + 1 are given in Equation (10).

$$w_{j,i}^{s+1} = w_{j,i}^{s} + D * rand(0,1) * (o_{ji} - w_{j,i}^{s}) + T * rand(0,1) * (h_i - w_{j,i}^{s}) \quad (10)$$

Every bird search for its objective based on its individual flying experiences as well as group experiences, in compliance with strategy 2. Consequently, the positionw_ (j, i) ^ (s+1) is provided below at times + 1 are given in Equation (10).

Birds begin to forage if the chance number (0,1) is smaller than a constant ϵ (0,1); otherwise, they stay vigilant.

3.4.2.2 Alert behaviour

The alert behaviour is further explained below Equations (11, 12 and 13) in compliance with strategy 3. Let$k$ be a positive integer ranging from 1 toM,, where $k \neq I$. The variables $a_1$and $a_2$belong to the interval $[0, 2]$. $oFit_j$ Represents the optimal fitness value of the i^{th} bird, while $sumFit_j$denotes the sum of the optimal fitness values of the entire colony. To prevent $sumFit_j$from being zero, a small constant is introduced. Additionally, $mean_j$is the average fitness value of the i^{th}bird.

$$w_{j,i}^{s=1} = w_{j,i}^{s} + B_1 * rand(0,1) * (mean_i - w_{j,i}^{s}) + B_2 * rand(-1,1) * (o_{l,i} - w_{j,i}^{s}) \quad (11)$$

$$B_1 = b_1 * \exp\left(-\frac{oFit_j}{sumFit_{j+F}}M\right), \quad (12)$$

$$B_2 = b_2 * \exp\left(\frac{oFit_j - oFit_l}{|oFit_j - oFit_l| + \varepsilon} * \frac{M * oFit_l}{sumFit_j + \varepsilon}\right) \quad (13)$$

3.4.2.3 Flight behaviour

Accordingly, per strategy 4, birds migrate to different places regularly to escape being chased or in search of food. Migration as a process is configured accordingly. They will eat once more when they get to their new location.

Flying habit is explained in Equation (14) and (15).

$$w_{j,i}^{s+1} = w_{j,i}^{s} + randm(0,1) * w_{j,i}^{s} \quad (14)$$

$$w_{j,i}^{s+1} = w_{j,i}^{s} + rand(0,1) * (w_{l,i}^{s} - w_{j,i}^{s}) * EK \quad (15)$$

Where; $(\epsilon\ [0, 2])$ represents the time intervals separating birds flying to a location to search for food and $(0, 1)$ is a typical Gaussian random number. The proposed Avian Swarm Optimized K-Nearest Neighbour (ASO-KNN) method was used to identify the architectural heritage style such as a) Min, b) Chuan, c) Jin, d) Jing.

4 Result

On a Windows 11 OS, weused the Python programming language to implement suggested techniques. This study suggests an autonomous identification of architectural heritage style based on ASO-KNN. Implementation of an architectural heritage style identification exercise using a collection of images that includes several types of historical buildings.

4.1 Confusion matrix

It is a grid that explains the differences between anticipated and actual class labels to display the proposed ASO-KNN model performance. It has measures that help assess model accuracy and spot misidentifications. Figure 4 shows a confusion matrix for identifying architectural heritage styles with a total of 860 instances. Use the same four styles: Min, Chuan, Jin and Jing. The confusion matrix illustrates a situation in which the suggested model ASO-KNN performs at identifying Chuan compared to the Min architecture.

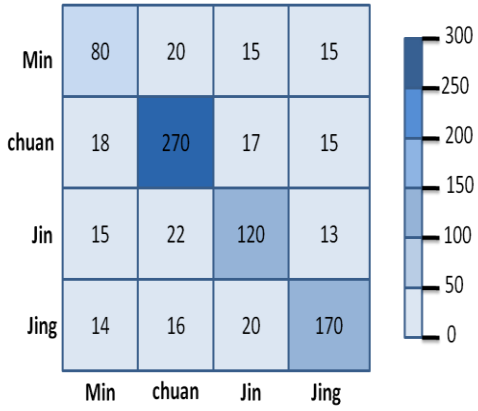


Figure 4: Confusion matrix

4.2 Comparative analysis

Comparative analysis based on metrics such as F1 score, recall, accuracy and precision can be utilized to compare existing methods to identify architectural heritage styles. Table 2 depicts performance analysis of the proposed ASO-KNN method and existing methods such as the Alex Krizhevsky Network (AlexNet) [20], Densely Connected Convolutional Network (DenseNet) [20] and Residual Network (ResNet) [20].

Table 2: Outcomes of proposed and existing method

Methods	**Precision (%)**	**Recall (%)**	**Accuracy (%)**	**F1 score (%)**
Alexnet [20]	77.21	57.53	57.53	63.56
Densenet [20]	97.08	82.92	82.92	88.56
Resnet [20]	94.80	73.02	73.02	81.41
ASO-KNN [Proposed]	**98.79**	**91.59**	**89.25**	**93.36**

Accuracy: The proportion of accurate forecasts among all of a model's forecasts is known as accuracy. Precise placement and measurement are crucial for the successful identification of architectural heritage style. The results of accuracy for the suggested and current approaches are shown in Figure 5. Compared to the AlexNet (57.53%), DenseNet (82.92%) and ResNet (73.02%) approaches, the recommended (ASO-KNN) method yielded a superior accuracy of 89.25%.

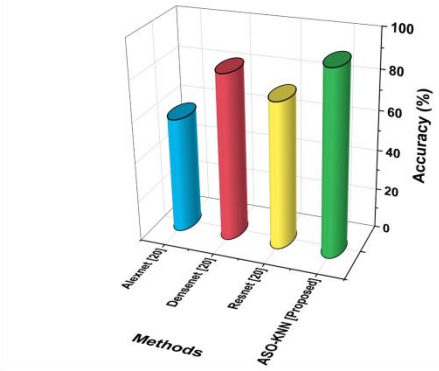


Figure 5: Outcomes of Accuracy

Precision: Precision can be defined as the proportion of accurately anticipated instances that were positive to all cases that the algorithm projected as positive. Figure 6 shows the comparative outcomes of precision. The suggestedASO-KNN approach produced a greater98.79% precision value whereas the AlexNet acquired 77.21%, the DenseNet obtained 97.08% and the ResNet displayed 94.80%.

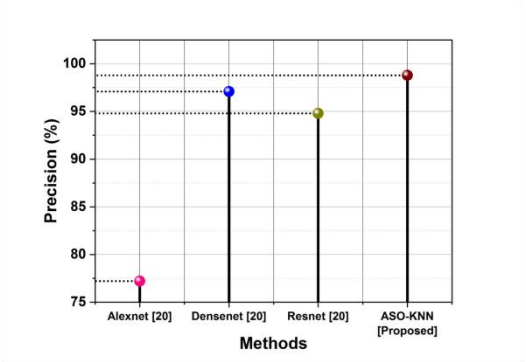


Figure 6: Outcomes of precision

F-1 Score: The F1 value is a metric that is commonly used in tasks involving binary identification to assess test performance. It is calculated using the harmonic means of recall and precision. Figure 7 shows the outcomes of precision values of proposed and existing methods. In contrast to other current approaches like AlexNet (63.56%), DenseNet (88.56%) and ResNet (81.41%), the suggested (ASO-KNN) approach produced better outcomes with 93.36% of F1-score.

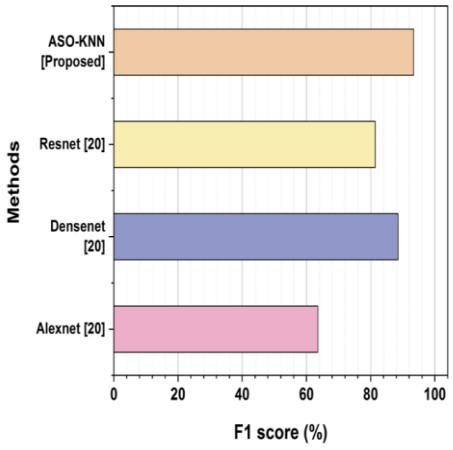


Figure 7: Evaluation of the F1 score

Recall: In the context of identifying architectural heritage style, recall refers to a detection system's capacity to accurately identify relevant examples of heritage style from among all of the actual instances that are present. Figure 8 shows the recall of heritage style identification. Compared to the AlexNet (57.53%), DenseNet (82.92%) and ResNet (73.02%) approaches, the recommended (ASO-KNN) method yielded anadvancedrecall of 91.59%.

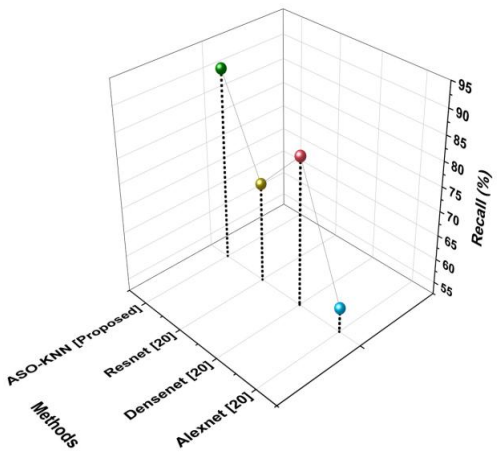


Figure 8: Comparative Outcomes of Recall

This hybrid strategy improves KNN's ability to identifying tiny distinctions between architectural styles by utilizing ASO's capacity to search the parameter space effectively. Compared to more conventional methods like weighted KNN or K-means clustering, this unique approach offers more accuracy and flexibility in heritage style detection by combining the sophisticated optimization of ASO with the basic categorization of KNN. ASO enhances KNN by optimizing the distance calculations, leading to better accuracy in identifying heritage styles. Cross-validation helps ensure the model performs well on different data subsets, reducing the risk of overfitting and making the results more reliable.

5 Discussion

The Avian Swarm Optimized K-Nearest Neighbors (ASO-KNN) model has various advantages for identifying architectural heritage styles, including adaptation to smaller datasets and efficiency in feature selection using avian swarm-inspired optimization approaches. This produces a model that is fewer computationally expensive and more comprehensible, making it ideal for scenarios with little information or capabilities. Conversely, cutting-edge models such as AlexNet, DenseNet, and ResNet, while significant, have significant limitations. Despite its revolutionary status, AlexNet's somewhat shallow architecture makes it less effective at managing complicated data. DenseNet, which addresses the issue of disappearing gradients with dense interactions, can be physically costly and susceptible to overlapping in extremely deep the internet. ResNet, have high computing requirements and difficulty, necessitating considerable time and money for development and inference. Thus, while ASO-KNN is a more environmentally friendly and uncomplicated strategy, sophisticated algorithms may have greater expenses and complications. Potential enhancements to the robustness of historical architectural style verification could involve integrating ASO-KNN with sophisticated neural networks for increased precision and effectiveness, using domain-specific extraction of features and communicate learning to adjust to different styles, and applying ensemble approaches to improve simulation dependability and speculation.

6 Conclusion

The study shows that cutting-edge Deep Learning (DL) algorithms and digital data may be used to efficiently assess and identify the distinctive architectural elements and construction methods characteristic of Chinese architectural heritage styles. The work improves the identification of architectural types by preprocessing data using grayscale picture techniques and capturing textural and structural aspects using Histogram of Oriented Gradient (HOG) descriptors. Accuracy and efficiency of data classification are greatly enhanced by the introduction of a hybrid model that leverages ASO behavior to combine Avian swarm-optimized K-Nearest Neighbors (ASO-

KNN) with regular KNN. The excellent performance measures achieved by the suggested methodology, which outperforms current state-of-the-art methods, include 89.25% accuracy, 93.36% F1-score, 91.59% recall, and 98.79% precision. These outcomes validate the method's efficacy and open new avenues for automated architectural style identification system development.

References

[1] X. Wang, J. Zhang, J. Cenci and V. Becue, 2021. "Spatial distribution characteristics and influencing factors of the world architectural heritage". *Heritage*, vol. 4, no. 4, pp. 2942-2959. https://doi.org/10.3390/heritage4040164

[2] P. González Martínez, 2022. "The preservation by relocation of Huizhou vernacular architecture: Shifting notions on the authenticity of rural heritage in China". *International Journal of Heritage Studies*, vol. 28, no. 2, pp. 200-215. https://doi.org/10.1080/13527258.2021.1955728

[3] H. Xu, H. Sun, L. Wang, X. Yu, and T. Li, 2023. "Urban Architectural Style Recognition and Dataset Construction Method under Deep Learning of Street View Images: A Case Study of Wuhan". *ISPRS International Journal of Geo-Information*, vol. 12, no. 7, p. 264. https://doi.org/10.3390/ijgi12070264

[4] L. Gao, Y. Wu, T. Yang, X. Zhang, Z. Zeng, C.K.D. Chan, and W. Chen, 2023. "Research on Image Classification and Retrieval Using Deep Learning with Attention Mechanism on Diaspora Chinese Architectural Heritage in Jiangmen, China". *Buildings*, vol. 13, no. 2, p. 275. https://doi.org/10.3390/buildings13020275

[5] E. Grilli and F. Remondino, 2020. "Machine learning generalization across different 3D architectural heritage". *ISPRS International Journal of Geo-Information*, vol. 9, no. 6, p. 379. https://doi.org/10.3390/ijgi9060379

[6] M.H. Abed, M. Al-Asfoor, and Z.M, Hussain, 2020. "Architectural heritage image classification using deep learning with CNN". http://ceur-ws.org/Vol-2602/

[7] Y. Liu, M. Hou, A. Li, Y. Dong, L. Xie, and Y. Ji, 2020. "Automatic detection of timber cracks in wooden architectural heritage using the Yolov3 algorithm". *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 43, pp. 1471-1476. https://doi.org/10.5194/isprs-archives-xliii-b2-2020-1471-2020

[8] J. Lee, and J.M. Yu, 2023. "Automatic Surface Damage Classification Developed Based on Deep Learning for Wooden Architectural Heritage". *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp.151-157. https://doi.org/10.5194/isprs-annals-x-m-1-2023-151-2023

[9] K. Ma, B. Wang, Y. Li, and J. Zhang, 2022. "Image retrieval for local architectural heritage recommendation based on deep hashing". *Buildings*, vol. 12, no. 6, p. 809. https://doi.org/10.3390/buildings12060809

[10] M. Samhouri, L. Al-Arabiat, and F. Al-Atrash, 2022. "Prediction and measurement of damage to architectural heritage facades using convolutional neural networks". *Neural Computing and Applications*, vol. 34, no. 20, pp. 18125-18141. https://doi.org/10.1007/s00521-022-07461-5

[11] H. Zou, J. Ge, R. Liu, and L. He, 2023. "Feature recognition of regional architecture forms based on machine learning: A case study of architecture heritage in Hubei Province, China". *Sustainability*, vol. 15, no. 4, p. 3504. https://doi.org/10.3390/su15043504

[12] [H. Yazdi, S. Sad Berenji, F. Ludwig, and S. Moazen, 2022. "Deep learning in historical architecture remote sensing: Automated historical courtyard house recognition in Yazd, Iran". *Heritage*, 5(4), pp.3066-3080. https://doi.org/10.3390/heritage5040159

[13] S. Luo, and H. Wang, 2024. "Digital Twin Research on Masonry–Timber Architectural Heritage Pathology Cracks Using 3D Laser Scanning and Deep Learning Model". *Buildings*, vol. 14, no. 4, p. 1129. https://doi.org/10.3390/buildings14041129

[14] V. Croce, G. Caroti, A. Piemonte, L. De Luca, and P. Véron, 2023. "H-BIM and artificial intelligence: classification of architectural heritage for semi-automatic scan-to-BIM reconstruction". *Sensors*, vol. 23, no. 5, p. 2497. https://doi.org/10.3390/s23052497

[15] M. Costa, and M.J. Carneiro, 2021. "The influence of interpretation on learning about architectural heritage and on the perception of cultural significance". *Journal of Tourism and Cultural Change*, vol. 19, no. 2, pp. 230-249. https://doi.org/10.1080/14766825.2020.1737705

[16] H. Zou, J. Ge, R. Liu, and L. He, 2023. "Feature recognition of regional architecture forms based on machine learning: A case study of architecture heritage in Hubei Province, China". *Sustainability*, vol. 15, no. 4, 3504. https://doi.org/10.3390/su15043504

[17] S.Y. Jang, and S.A. Kim, 2023. "Automatic extraction and linkage between textual and spatial data for architectural heritage". *ACM Journal on Computing and Cultural Heritage*, vol. 16, no. 3, pp. 1-19. https://doi.org/10.1145/3586158

[18] M. Deligiorgi, M.I. Maslioukova, M. Averkiou, A.C. andreou, P. Selvaraju, E. Kalogerakis, G. Patow, Y. Chrysanthou, and G. Artopoulos, 2021. "A 3D digitization workflow for architecture-specific annotation of built heritage". *Journal of Archaeological Science: Reports*, vol. 37, p. 102787.

https://doi.org/10.1016/j.jasrep.2020.102787

[19] M. Sun, F. Zhang, F. Duarte, and C. Ratti, 2022. "Understanding architecture age and style through deep learning". Cities, vol. 128, p. 103787. https://doi.org/10.1016/j.cities.2022.103787

[20] Q. Han, C. Yin, Y. Deng, and P., Liu, 2022. "Towards a classification of architectural styles of Chinese traditional settlements using deep learning: A dataset, a new framework and its interpretability". *Remote sensing*, vol. 14, no. 20, p. 5250. https://doi.org/10.3390/rs14205250

Improved VITS-Based Multilingual AI Speech Synthesis Model with Domain Adaptors and Acoustic Feature Optimization

Xing Yang
Communication University of Shanxi, Shanxi Film Academy, Jinzhong, 030619, China
E-mail: soundgoal0506@outlook.com

Keywords: speech synthesis technology, acoustic characteristics, domain adaptors, audio conversion, multilingual speech synthesis

Received: November 19, 2024

Speech synthesis technology plays an important role in global economic and cultural exchanges, and multilingual speech synthesis and output are still unable to meet the current development needs of the global market. The study proposes the use of acoustic feature conversion methods and steps for decoupling multilingual information, combined with modules of domain adaptors to improve end-to-end text to speech variational inference and adversarial learning models, to adapt to the application of multilingual speech synthesis. Through the evaluation of speech synthesis technology indicators, it was found that the average selection score of the model after removing the regularization term for similarity in different languages was 4.93. The synthesis model without domain adaptors significantly reduced the naturalness of speech synthesis by 0.8 compared to multilingual speech synthesis models, indicating that domain adaptors have a good effect on the naturalness of speech synthesis. In cross-lingual indicator analysis, the model proposed by the research achieved the highest naturalness result, with an average selection score of 4.26 and 3.96 for naturalness and similarity in transit English. In the intermediate day voice synthesis with a data volume of 200, the highest accuracy was 94.58%, which was 16.53% higher than traditional speech synthesis frameworks. Comparing the cross-lingual synthesis performance of the synthesis model, it was found that the model had an accuracy rate of 94.58% and a time of 3.12 seconds for the synthesis of Chinese to Japanese conversion with a data volume of 200. The above results demonstrate the feasibility and superiority of the multilingual speech synthesis model based on domain adaptors, which adds multilingual imagery to speech synthesis applications in the field of artificial intelligence and promotes the industrial development and intelligent services of speech synthesis technology.

Povzetek: Raziskava predstavi izboljšan govorni sintetiziator z modelom VITS in domenskimi adapterji, kar poveča naravnost, točnost in časovno učinkovitost sinteze govora v različnih jezikih.

1 Introduction

With the development of technology, artificial intelligence (AI) is constantly integrating into people's lives 0. Speech recognition and synthesis technology have promoted robots to “speak” and become important means of human-computer interaction. Speech recognition can convert speech into text, while synthesis technology can transform textual information into speech patterns, and the emotions, timbre, rhythm, and other aspects of synthesized speech are increasingly catering to human image and scene needs. In addition, the diversity of global language types has also led to a wider range of objects for speech synthesis, and the transformation of multilingual synthesized speeches can meet different language environments and multimedia platforms 0. In personalized development and social demand, speech synthesis and speech recognition technologies typically perceive language environment features to adapt to the user’s environment. In the oral application of patients with hearing impairment, autism, and language disorders, speech recognition technology can extract features from the original voice and enhance the test speech to improve the language ability of patients with disabilities [3]. Speech synthesis technology, based on speech recognition, adds text processing and audio conversion to language, enhancing the quality of speech synthesis through analysis of audio signals. In recent years, scholars at home and abroad have conducted a lot of research on speech recognition and synthesis technology. In speech recognition technology, Jiang et al. proposed to address the technical issues of text to speech synthesis by using semantic dependencies to extract speech information from the original text. They combined convolutional neural networks and self attention mechanisms to improve pronunciation accuracy, and separated tones and phonemes in model training to demonstrate the reliability of their model in speech synthesis [4]. Liu et al. proposed a multi-modal speech synthesis method in the field of dubbing, which utilizes multi-modal and multi-scale text to speech technology to provide style embedding for video input. Multi-scale styles are reused to convert the speech style of reference videos, thereby generating high-quality video aligned speech [5]. In the problem of distinguishing speech

emotions, Long et al. proposed a multi-distribution method based on Mel frequency cepstral graph and parameter transfer, and proved its effectiveness in classifying language emotions through model training [6].

In the subsequent field of speech synthesis technology, breakthroughs were made in speech classification and recognition applications, and the development of speech conversion and output was also expanded. Ghosh et al. proposed the use of visual transformers and attention mechanisms to capture the spectral and temporal features of speech in the audio-visual field, thereby improving the quality of speech synthesis [7]. Okamoto et al. proposed using a neural encoder and a multi-speaker corpus to resample acoustic features for speech rate conversion technology, demonstrating that their method has high-quality speech rate conversion [8]. Masood et al. proposed the use of generative adversarial networks and deep forgery generation tools to detect audio and video data in response to speech conversion and forgery issues on social media, demonstrating the rapid development of speech conversion and the application of synthesis technology [9]. In the application of language and culture, Zhang et al. proposed a cross-lingual pre-training method for adversarial training of text and image generation, thereby increasing the robustness of the model to the target language [10]. Kaur et al. proposed a method for text to speech synthesis based on deep learning to construct an acoustic model, which can improve the fidelity of synthesized speech and achieve high performance evaluation [11]. However, in terms of specific multi-lingual synthesis techniques and audio conversion, current research still lacks in-depth analysis, and the phonemes, pronunciations, and duration of language data have not been adjusted. Due to the significant differences between different languages, it is difficult to establish a cross-lingual sound learning model. At the same time, sound collection and datasets for multiple languages are scarce, which leads to model training only with a small number of language data. The generalization ability and speech output quality of the model cannot meet people's actual needs. Currently, cross-lingual speech synthesis cannot fully achieve the mutual conversion of multiple languages. Therefore, based on the variational inference with adversarial learning for end-to-end text to speech (VITS) model, this study innovatively uses neighborhood adaptors to extend multi-lingual language rules and improves the structure of the VITS model to construct a multi-lingual AI speech synthesis model based on domain adaptors. To meet the application needs of speech synthesis in various fields and globalization, research is being conducted on the mutual conversion and recognition of output sounds from multiple languages. At the same time, advanced technology is being used to establish an AI image with multi-lingual speech synthesis capabilities, aiming to improve the fluency and naturalness of speech synthesis in different languages, and provide personalized speech experience and technical reference for the field of AI.

This study is mainly divided into five sections. The first section elaborates on the existing research to introduce the content of this study. The second section is to analyze speech synthesis technology and acoustic feature conversion methods, and establish a VITS model, providing a foundation for the subsequent construction of synthetic models. The third section is to optimize and improve the VITS model using domain adaptors and multi-lingual information modules to establish a multi-lingual speech synthesis model. The fourth section is to combine the improved multi-lingual speech synthesis model with the AI field to form a multi-lingual AI speech synthesis model. The final section uses publicly available speech datasets to analyze the evaluation indicators of the proposed speech synthesis model, to demonstrate the reliability and superiority of the research method.

To demonstrate the novelty and innovation of the research design ed method, a summary of the above work has been made, as shown in Table 1.

Table 1: Summary of related works

Reference	Method	Result	Shortcomings
Jiang et al [4]	Using semantic dependencies to extract speech information, convolutional neural networks and self attention mechanisms provide speech training.	Model training can separate tones and phonemes, improving the processing of pauses, stress, and intonation in speech.	The method of extracting speech information from text is prone to limitations and dependencies on contextual information.
Liu et al [5]	The multi-modal and multi-scale text to speech technology provides style embedding and conversion for video input.	Transform speech styles in the field of dubbing and generate high-quality video aligned speech.	The speech extracted from the public grid corpus may have limitations in a single language, making it impossible to achieve dubbing videos between different languages.
Long et al [6]	Multi-distribution method based on Mel frequency cepstral graph and parameter transfer.	The model training proves its effectiveness in classifying language emotions.	The collection environment and equipment for voice data are not universal, and the network's processing of voice is relatively

			complex.
Ghosh et al [7]	In the audio-visual field, visual transformers and attention mechanisms are used to capture the spectral and temporal features of speech.	Improved the quality of speech synthesis.	Only comparing the clarity of speech in the speaking environment lacks quality detection of the synthesized input speech.
Okamoto et al [8]	Use neural encoder and multi-speaker corpus to resample acoustic features.	Its method has high-quality speech rate conversion.	The experimental results of the assistive robot cannot provide reference for the application of real-world scenarios or human voice input.
Masood et al [9]	Generate adversarial networks and deep forgery generation tools to detect audio and video data.	Improved the rapid development of speech conversion and the application of synthesis technology.	There is a lack of more advanced technology in the process and detection of deepfake videos and speech.
Zhang et al [10]	Use cross-language pre training methods for adversarial training of text and image generation.	Increased the robustness of the model to the target language.	Lack of low resource language training for parallel texts in practice can limit the generation of some cross linguistic text to image results.
Kaur et al [11]	Build acoustic models based on deep learning.	Improving the fidelity of synthesized sound, the performance of text to speech synthesis is highly evaluated.	Speech processing heavily relies on the use of synthesizer acoustic models, and the processing of acoustic models is still relatively complex.
Li et al [18]	Use error correction code to improve the Bidirectional Encoder Representation from Transformers model and handle contextual information.	The ability to extract contextual information features and natural language classification has been improved.	Natural language processing is still constrained by the error correcting codes of the model, resulting in a lower ability to recognize its structure.
This study	A domain adaptive multilingual AI sound synthesis model.	In multi-lingual cross-speech synthesis technology, the training efficiency and speech quality of the synthesis model have been improved.	The language input lacks intervention for issues such as accent, and the text model needs to be refined.

2 Methods and materials

Through the development of speech synthesis technology and acoustic feature conversion, the VITS model and the addition of neighborhood adaptors are studied for module improvement, and a series of loss functions are designed to ensure the quality of speech synthesis.

2.1 Speech synthesis technology and acoustic feature conversion methods

In the development process of computers and acoustics, the basic content of speech synthesis technology includes audio processing technology and front-end processing methods for speech synthesis [12-13]. Among them, audio pre-emphasis processing is a common signal processing technique, which enhances the energy of high-frequency signals by adding an emphasis filter to the audio signal, thereby improving the frequency response of the audio signal, as shown in equation (1).

$$J(s) = d(s) - \lambda \times d(s-1) \tag{1}$$

In equation (1), $J(s)$ represents the audio signal after emphasis. $d(s)$ is the initial, unprocessed audio signal. Among them, s is the audio signal, λ is the audio pre-emphasis coefficient, and $[0.9,1.0]$. Fourier transform and wavelet transform are common frequency domain analysis methods in signal processing technology, mainly converting time-domain signals into frequency-domain signals and analyzing the effectiveness of digital signal processing.

In addition, the front-end processing technology of speech synthesis, as an important part of speech synthesis methods, focuses on the processing and conversion of text information, converting human text into appropriate rhythms and vocalizations, thereby providing signals and parameters for the establishment of acoustic models [14]. In the field of speech synthesis, it is necessary to convert

the processed acoustic features into playable speech waveforms, as shown in Figure 1.

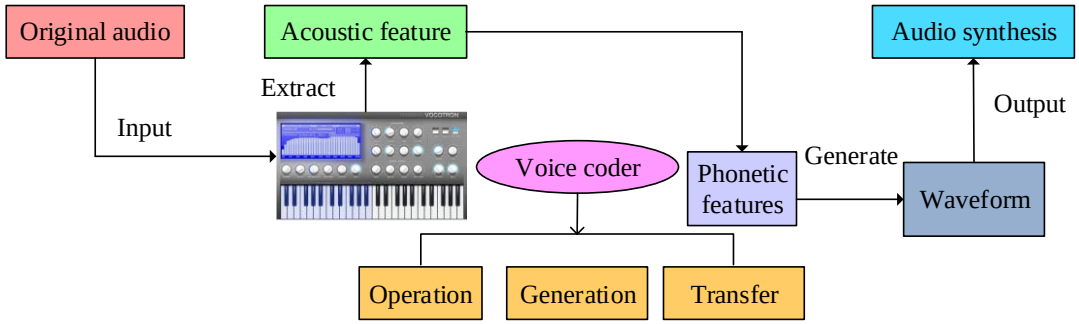


Figure 1: Schematic diagram of the operation of the voice encoder

From Figure 1, the encoder extracts acoustic features from the original audio, and the speech encoder generates audio waveforms through feature operations, generation, and transmission, ultimately achieving the playback of synthesized speech. As the digitalization level improves, codecs have better processing technology in terms of speech quality and compression efficiency. A speech synthesis model based on deep learning technology is developed for end-to-end speech synthesis, which includes modules of encoder and decoder to generate speech waveforms in text sequence input and acoustic feature output. The VITS model is an advanced speech synthesis model in recent years, which shows good performance in generating speech quality and fidelity, as shown in Figure 2.

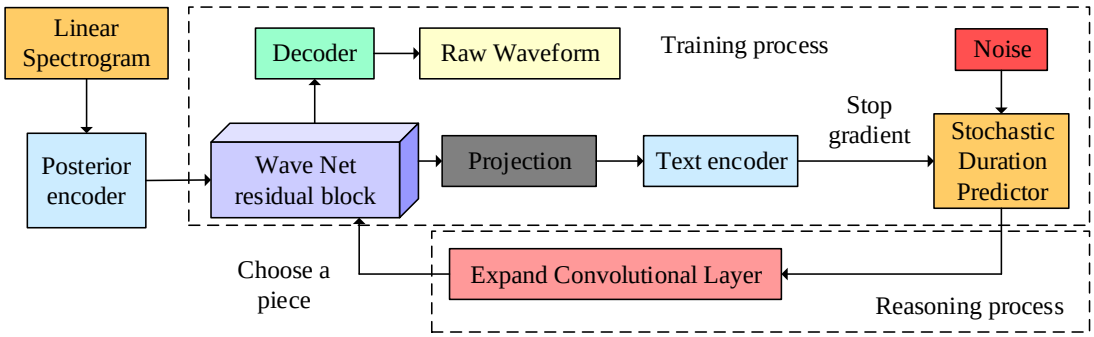


Figure 2: Schematic diagram of the training and inference process of the ITS model

From Figure 2, the structure of the VITS model mainly includes modules such as text encoder, decoder, posterior encoder, and random duration predictor. The original waveform can be converted into text information through the processing of the encoder and residual block. In the inference process, the text encoder is a converter encoder, and the input phonemes are passed through the linear projection layer on top of the text encoder to obtain corresponding representations from the input phonemes. At the same time, the position of this layer is used to construct the mean and variance of the prior distribution, and then the normalization process of monotonic alignment search can be completed. Among them, post encoding selects a residual block from the residual blocks to train the convolutional layer, and embeds the voice of the sound input in the residual block, which can help improve the performance of subsequent speech conversion. In addition, speech noise can be parameterized in the speech layer through random duration predictors and discriminators to improve the performance and quality of the conversion.

2.2 Improvement of domain adaptors and multi-lingual prediction methods

Due to the fact that multilingual speech synthesis technology and its model training have become a research hotspot in cross-lingual communication, a cross-lingual speech synthesis model is established by integrating multi-lingual information and grammar rules into the VITS model. When incorporating multi-lingual information, the VITS model needs to replace the text encoder in the model structure, and add language embeddings and time predictors in the standardized residual block to comprehensively consider the duration differences of language factors. In the VITS model, adversarial learning is used to optimize the variational auto encoder (VAE) to improve the quality of synthesized speech and real-world context [15]. The loss function of VAE is shown in equation (2).

$$L_{VAE} = (\mu, \lambda; d) = L_{\mathrm{Re}}(\mu, \lambda; d) + L_{En}(\mu, \psi; d) \quad (2)$$

In equation (2), Ψ represents the optimization parameter for gradient reversal. To maximize the input data, the principle of VAE is studied and explained. Assuming the input data d and its hidden representation z, the relationship between the two is first calculated using conditional probability distribution $y(d|z)$ and prior probability distribution $y(d)$, which also represents the prior distribution of unknown data in the hidden space. Edge likelihood is performed on the posterior distribution of hidden variables, as shown in equation (3).

$$y(d)=\int y(d|z)\times y(z)\times z' \tag{3}$$

In equation (3), $y(d)$ represents the distribution of real data, z is the dimension of the hidden space, and z' is the partial derivative of the dimension. Due to the uncertainty of dimensions, the prior distribution is not a single result, so an approximation of the posterior distribution is introduced to solve the integral. Under variational theory, the prior distribution also needs to be close to its approximate value, as shown in equation (4).

$$y_{\mu}(d)=\int y_{\lambda}(z|d)\times\frac{y_{\mu}(d|z)y(z)}{y_{\lambda}(z|d)}\times z' \tag{4}$$

In equation (4), $y_{\mu}(d|z)$ represents an approximate posterior distribution, $y_{\lambda}(z|d)$ is an approximate prior distribution, and $y(z)$ is a prior distribution. After incorporating multilingual information, the study added a classifier for voice input in the synthesis model, namely the domain adaptors, as shown in Figure 3.

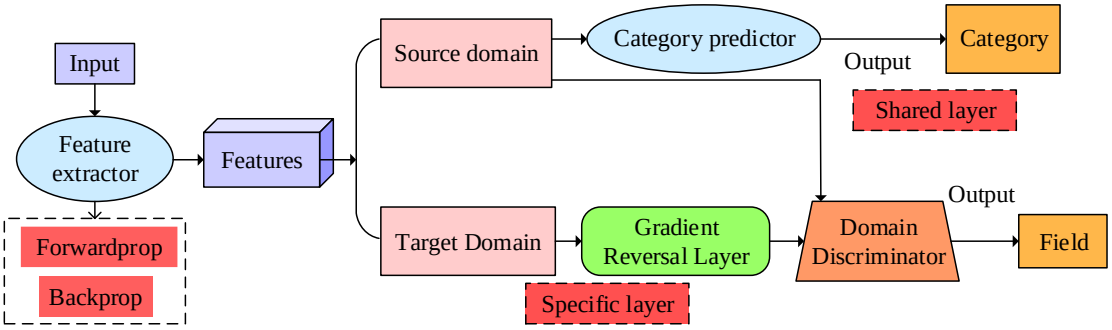


Figure 3: Structure and gradient reverse layer of domain adversarial neural networks

From Figure 3, the domain adversarial neural network will use a feature extractor to separate the features into the source domain and the target in two layers when inputting content. These two layers are the shared layer and the specific layer, with the source domain and target domain as the two assumed datasets in the model, and trained separately at each layer to update weights. In addition, each dataset has a specific classifier and domain classifier used to determine whether the input samples come from the source domain or the target domain. The source domain is located in the category predictor to classify the domain, and the weights are updated through backpropagation. The target domain uses a gradient reverse layer (GRL) in a specific layer to reverse the sign, and then passes it to the shared layer, so that the model ignores the differences between domains [16-17]. In the process of network training, the weights of the shared layer are updated through the source domain classifier and its loss. In the GRL, the classifier's loss is added to the shared layer's loss, and then gradient backpropagation is used to update the weights. Due to GRL being a layer in the network structure, it evolves into negative values through gradient reversal in reverse broadcasting, transferring data from one domain to another, thereby prompting the network model to ignore differences and preserve common data features. In addition, in a specific layer, the target domain needs to be combined with a neighborhood discriminator to determine the type of input domain. At the same time, the introduction of domain classifiers not only achieves classification of each sample, but also helps GRL train its discriminator in backpropagation, making the network model robust to input samples from different datasets, as shown in equation (5).

$$g_{\gamma}(a)=a \quad \frac{dg_{\gamma}}{da}=-\gamma I \tag{5}$$

In equation (5), g represents the gradient backpropagation period, a is the input content in the network structure, and γ is the parameter value of dynamic transformation. dg_{γ} represents the process of reversing the gradient direction, while I represents the identity transformation in the forward propagation process. Therefore, GRL can help train domain classifiers in the backpropagation process, allowing the model to ignore differences between domains. At the same time, gradient inversion to negative values can not only achieve domain adaptation, but also preserve the common features of the data during transmission. Due to the fact that the learning rate parameter also changes with the iteration process in network architecture, to ensure the form of identity transformation, the dynamic expression of the parameter is shown in equation (6).

$$\gamma_L = 2 \times \frac{1}{1 + \exp\left(-\gamma \times L\right)} - 1 \tag{6}$$

In equation (6), L is the linear variation during the learning process of the network model, and $\exp$ is an exponential function with the natural constant e as the base. This can force domain classifiers to ignore irrelevant information while forcing feature extraction, thereby increasing the model's generalization ability and improving data processing capabilities in other domains. However, to distinguish the accent and text information of the voice input, a regularization loss term is added in the decoupling process between the input and speech information, as shown in equation (7).

$$R = \left\| E_{x \in X} \left[conv\left(S_x\right) \right] \right\|_2 \tag{7}$$

In equation (7), R is the regularization loss term, E represents the embedding of the speech input, $conv$ is the convolutional layer, and the kernel is. S_x is the embedding vector of the speech input in data point x. Specifically, the representation of the sound input is extracted, the average value of the embedding vector of the sound input is solved, and then the hidden average value is pushed to the zero vector through a convolutional layer, thereby helping the synthesis model to separate the sound input from its content when generating audio, and the generated sound is not affected by language or accent.

2.3 Construction of multilingual AI speech synthesis model

In the development of the AI era, the field of speech synthesis is closely linked to the processing of AI technology, while speech synthesis technology has been widely applied in the AI field. Therefore, the improved multi-lingual speech synthesis model will be combined with the AI field to form a multilingual AI speech synthesis model, as shown in Figure 4.

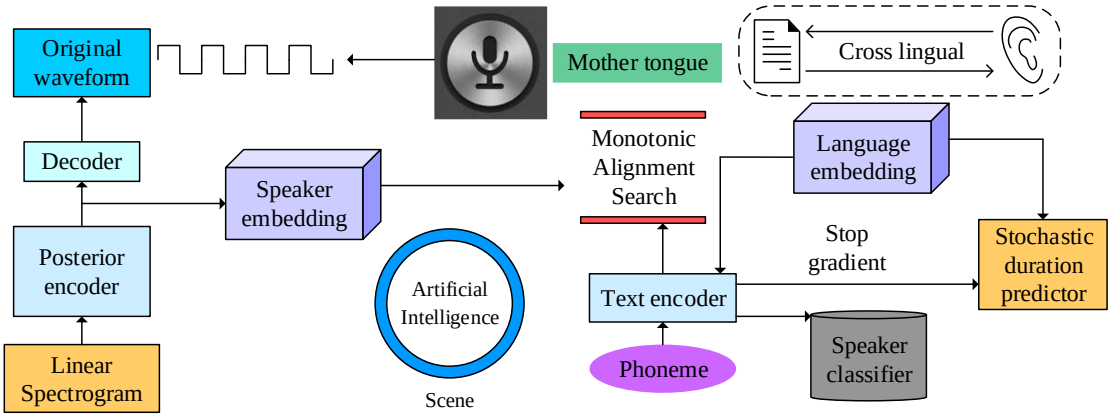


Figure 4: Schematic diagram of the process of multi-lingual AI speech synthesis model

From Figure 4, through the propagation description of text and sound, multi-lingual information input can be achieved, and the task of text editor can be realized under language embedding. The classifier that reuses sound input performs audio processing on the input sound information. The original waveform generated by the input speech is processed by the decoder's processor, which can improve the efficiency of sound generation and the naturalness of synthesis. When processing different audio text data, the synthesis model uses a random duration predictor and classifier for the overall embedding of input text or speech to avoid a large difference in audio length between converted data. This can reduce the instability of the sound synthesis model and solve the task of decoupling phonemes and information of the sound input in the model, achieving multilingual sound synthesis. In addition, in AI scenario applications, the multi-lingual speech synthesis model is integrated with the speaker embedding module to facilitate the implementation of multi-lingual AI speech synthesis models for other languages through sound input and encoder processing.

The improved VITS model utilizes a joint loss function to improve the conditional encoder during training, including recombination loss function, relative entropy loss function, and adversarial function. The first two are used to optimize the conditional VAE, while adversarial loss is used to optimize the discriminator network. The recombination loss function can balance the difference between the recombined speech and the initial speech in the speech synthesis model, as shown in equation (8).

$$L_{\text{Re}} = \left\| d_{MEL} - \hat{d}_{MEL} \right\|_1 \tag{8}$$

In equation (8), L_{Re} represents the difference calculation of the recombination loss function on the audio. MEL represents Mel spectrum. d_{MEL} and $\hat{d}_{MEL}$ are the Mel spectra of the initial audio and generated audio, respectively. The relative entropy loss function is shown in equation (9).

$$L_{En} = \log y_{\lambda}\left(H \middle| d_{lin}\right) - \log y_{\mu}\left(t_C, A_C\right) \quad (9)$$

In equation (9), L_{En} represents the relative entropy loss function, d is the sample of the input model, that is, the speech or text description. H is the output latent variable, and $y_{\lambda}\left(H \middle| d_{lin}\right)$ and $y_{\mu}\left(t_C, A_C\right)$ are the posterior and prior distributions, respectively. Among them, λ and μ are the parameters of the encoder and decoder, respectively. At this point, the use of encoders brings stronger nonlinear capabilities to the model, reducing reliance on coupling layers and streamlining model size. In addition, $\left(t_C, A_C\right)$ is the encoder output result of upsampling in the model structure, which specifically includes the phoneme t_C extracted from the text and its permutation combination A_C with variables. The latent variable is shown in equation (10).

$$H \sim y_{\lambda}\left(H \middle| d_{lin}\right) = N\left(H; M_{\lambda}\left(d_{lin}\right), \sigma_{\lambda}\left(d_{lin}\right)\right) \quad (10)$$

In equation (10), N is the maximum number of samples for the VAE model. N is the mathematical expectation of a normal distribution, and σ is the standard deviation. To provide higher resolution information to the posterior encoder y_{λ}, the above equation uses Mel as the input content. At the same time, to improve the expressive power of prior distributions, a standardized flow is introduced in the text encoder to perform reversible transformations between simple distributions and complex distributions corresponding to latent variables, thereby generating samples that approach reality. However, the reversible transformation of the encoder output is shown in equation (11).

$$y_{\mu}\left(H \middle| t_C\right) = N\left(f_{\mu}\left(H\right); M_{\mu}\left(t_C\right), \sigma_{\mu}\left(t_C\right)\right) \times \left| \det \frac{\partial f_{\mu}\left(H\right)}{\partial_H} \right| \quad (11)$$

In equation (11), f is an invertible function, f_{μ} is the mathematical concept of normalized flow, where $\det$ represents the determinant and is used to calculate the linear transformation properties of a square matrix, and ∂ is the partial derivative of the latent variable H. Due to the misalignment of the input and extraction of the prior encoder with the true labels, the model estimates the alignment information during training iterations to provide more high-resolution information for subsequent encoders. Therefore, the study adopts a monotonic alignment search normalization process, which is also a search alignment method that maximizes the possibility of parameterized data in the normalized flow, thereby achieving calibration of output text and target sound, as shown in equation (12).

$$A_C = Arg\max \mathrm{Logp}\left(d \middle| t_C, \hat{A}_C\right) \quad (12)$$

In equation (12), $Arg\max$ represents the function expression for finding parameters, and p is the prior distribution of the standardized flow. The random duration predictor in the synthetic model can achieve search calibration using VAE, thereby maximizing the possibility of normalized parameterized data, as shown in equation (13).

$$Arg\max_{\hat{A}_C} \log p_{\mu}\left(d_{MEL} \middle| H\right) - \log \frac{y_{\mu}\left(H \middle| d_{MEL}\right)}{y_{\mu}\left(H \middle| t_C, \hat{A}_C\right)} = \log N\left(f_{\mu}\left(H\right); M_{\mu}\left(t_C, \hat{A}_C\right), \sigma_{\mu}\left(t_C, \hat{A}_C\right)\right) \quad (13)$$

In equation (13), $\hat{A}_C$ represents the maximum degree of sorting combination of hidden variable logarithms, which simplifies the sorting combination while ensuring its rationality and readability. Due to the fact that the duration of input phonemes is extracted from random noise samples by a random duration predictor, its lower limit of variation is shown in equation (14).

$$\log_{y_{\mu}}\left(D \middle| t_C\right) \geq E_{y_{\lambda}\left(u,v \middle| D, t_C\right)}\left[\log \frac{y_{\mu}\left(D-u, v \middle| t_C\right)}{y_{\lambda}\left(u, v \middle| D, t_C\right)}\right] \quad (14)$$

In equation (14), u and v are added random variables to balance the time length. D is the spatial channel, which is the discriminator added in the VITS model to distinguish between the decoder output content and the real waveform. Through variable sampling, high-dimensional transformed integers can be obtained. Therefore, the joint loss function of the multilingual AI speech synthesis model is shown in equation (15).

$$L = L_{\mathrm{Re}} + L_{En} + L_{dur} + L_{Adv} + L_{Fm} \quad (15)$$

In equation (15), L_{dur} is the duration function, L_{Adv} is the adversarial loss function, and L_{Fm} is the linear regression function. Through the conditions of a multilingual AI speech synthesis model and a joint loss function, efficient decoupling of speech input and language information can be achieved in AI applications, thereby synthesizing multilingual speech outputs.

3 Results

Based on the improvement method of multi-lingual AI speech synthesis model, this study used evaluation indicators of speech synthesis technology to train the synthesis model, and tested and analyzed single-language

synthesis and cross-lingual synthesis separately to verify the feasibility and superiority of the model.

3.1 Analysis of evaluation indicators for speech synthesis models in a single language

Due to the diversity of languages, multi-lingual speech synthesis data plays an important role in testing the training of synthesis models. Therefore, the research chose speech datasets with multiple scenarios, environments, and continuous updates, and these datasets have broad openness and professional labeling, suitable for analysis in specific fields. Specifically, it includes the Chinese synthesized dataset AIshell-2, while the English synthesized dataset includes the centre for speech technology voice cloning toolkit (CSTR VCTK) dataset, text to speech (LJSpeech) dataset, and library transcriptional speech (libriSpeech) dataset. The Japanese synthesis dataset is a text transcription and reading audio (JSUT) dataset. To ensure the training of the multilingual synthesis model, the selected audio sampling rate was 22.05KHz and the sampling depth was 16 bits.

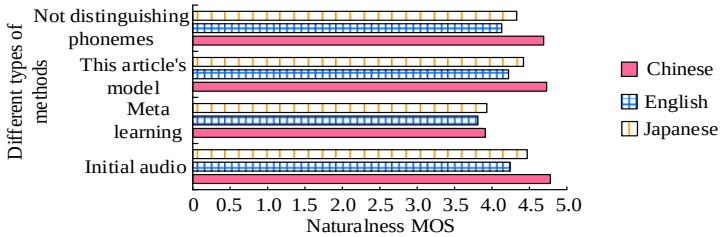
According to the evaluation metrics of speech synthesis technology and speech synthesis models, the natural effects of speech synthesis are usually evaluated using the mean option score (MOS) and similarity methods. The dataset and parameters involved in synthesizing the model are shown in Table 2.

Table 2: Results of dataset selection and model parameter setting

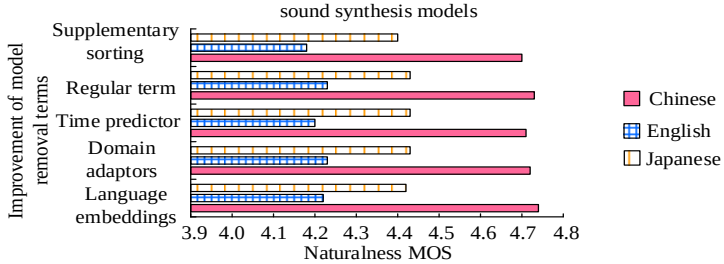
Data sets	The number of speech input users (unit)	Data duration (h)
AIshell-2 Data set	250	400h
CSTR VCTK Data set	500	250h
JSUT Data set	130	40h
Total	880	690h
Model parameters and training configuration	Input classifier	$\gamma = 10$
	Video Memory	Tesla V100 2G
	Batchsize	64
	Training frequency	800 epochs
	Training duration	110h

From Table 2, the multi-lingual speech synthesis dataset was selected, resulting in a total of 880 input data with a duration of 690h. Among them, 250 input data were selected for the Chinese speech synthesis data, and a 300ms silence segment was added to the beginning and end of each language audio data to ensure the unified output of the model during training. Considering the diversity of speech datasets and the processing of synthesis techniques, the model parameters set the weight coefficients of the classifier to 10, and the training times and duration to 800 and 110 hours, respectively.

The MOS evaluation index consisted of a 1–5-point scoring table and an intelligent platform. The study selected a meta learning synthesis model and sequentially removed modules from the proposed model to verify its synthesis effect. This type of model experiment that removes a single component from the model is called ablation experiment, which tests the changes in the initial audio and indistinguishable phonemes to demonstrate the evaluation of language MOS by different components in the model, and also proves the optimization level of cross language synthesis model. The naturalness MOS results are shown in Figure 5.



(a) Multi language naturalness evaluation results of different sound synthesis models



(b) The multilingual evaluation results of different terms in the sound synthesis model presented in this article

Figure 5: Evaluation results of different models on the naturalness MOS of multiple languages

From Figure 5 (a), the initial audio and different speech synthesis models had a naturalness MOS of around 4.4 in multiple languages. The model with the lowest evaluation result was the meta learning speech synthesis model, with naturalness MOS of 3.91, 3.81, and 3.93 for Chinese, English, and Japanese, respectively. The speech synthesis model proposed in the study rated 4.73, 4.22, and 4.42 for the three languages, with the overall evaluation performance being the best. In Figure 5 (b), after removing the domain adaptors, the results of the multilingual naturalness MOS were 4.72, 4.23, and 4.43, respectively. The results after replacing the time predictor were 4.71, 4.20, and 4.43, respectively. The similarity of multiple languages was evaluated and analyzed, as shown in Figure 6.

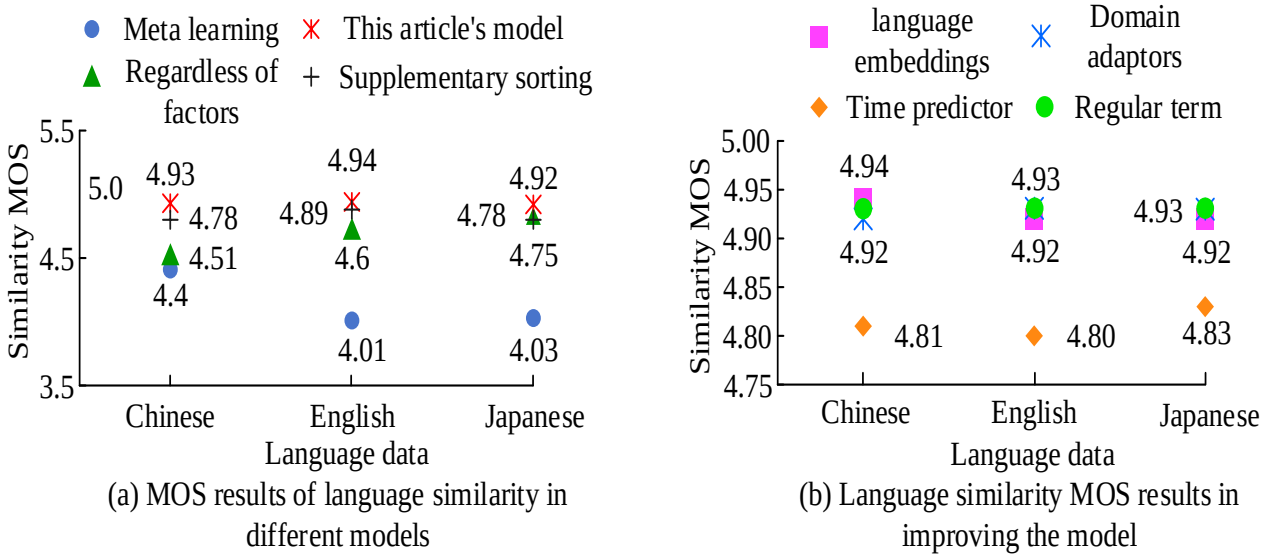
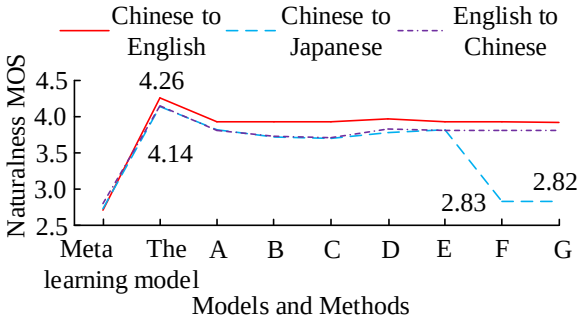


Figure 6: Comparison of MOS similarity results for multiple languages under different models and states

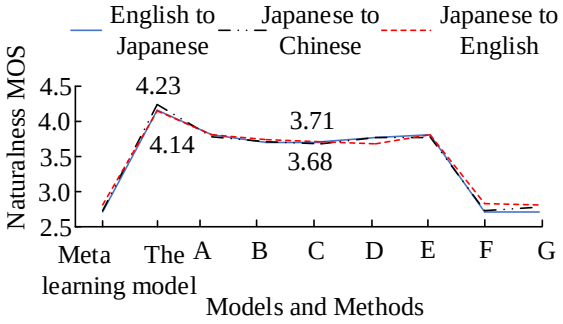
From Figure 6 (a), the similarity results of the three languages in the meta learning synthesis model were relatively low, with values of 4.4, 4.01, and 4.03, respectively. The speech synthesis model without the supplementary sorting component had similar similarity values in the three languages, with values of 4.78, 4.89, and 4.78, respectively. After removing the module that does not distinguish phonemes, the improved synthesis model had the lowest similarity of 4.51 in the Chinese language, while the proposed speech synthesis model had the highest similarity MOS values in the three languages, specifically 4.93, 4.94, and 4.92. In Figure 6 (b), the similarity MOS values of the research model significantly decreased after removing the time predictor, with values of 4.81, 4.80, and 4.83, respectively. After removing the regularization term, the language similarity MOS was 4.93 for all three languages. After removing the domain adaptors, the improved synthesis model had similar similarity scores for the three languages, which were 4.92, 4.93, and 4.93, respectively. Compared with other methods, the improved model with domain adaptors removed had better similarity MOS effect, which proves the reliability of domain adaptors in the model. Taking into account the MOS values of all individual languages, it can be concluded that the proposed speech synthesis model performed well in synthesizing individual languages.

3.2 Analysis of evaluation indicators for cross-lingual speech synthesis models

The multi-lingual AI speech synthesis model proposed by the research, which involves domain adaptors, random duration predictors, and regularization terms in the improvement method, cannot reflect the conversion between language audio and text in the synthesis effect of a single language. Therefore, in the same comparative model and method, the naturalness and similarity of the cross-lingual synthesis effects of Chinese, English, and Japanese were analyzed. The proposed model was translated into A~G to simplify the results by removing language embeddings, domain adaptors, replacement time predictors, regularization terms, supplementary sorting, zero vectors, and uniform phonemes. In the analysis of the naturalness MOS index, the results are shown in Figure 7.



(a) The naturalness results of Chinese English conversion using different model methods

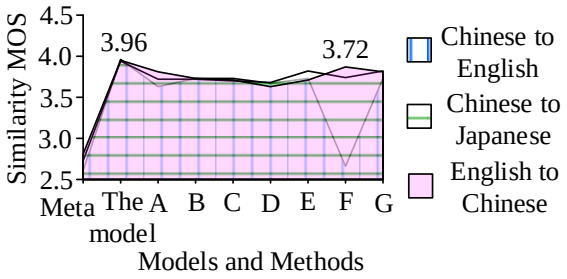


(b) The naturalness results of English Japanese conversion using different model methods

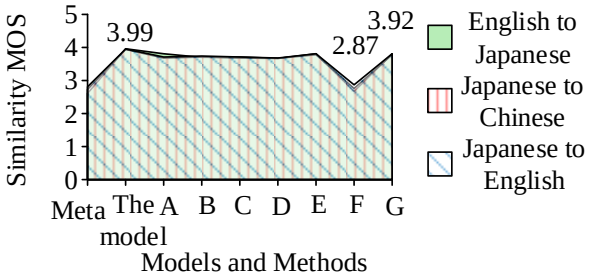
Figure 7: The naturalness MOS results of cross-lingual synthesized speeches using different models

From Figure 7 (a), the meta learning synthesized speech model had the lowest naturalness in Chinese English, Chinese Japanese, and English Japanese conversion, all below 3.0. The naturalness results of the model proposed by the research reached the highest level, with a naturalness MOS value of 4.26 for transit English and 4.14 for speech synthesis naturalness MOS on transit day. When zero vectors and unified phonemes were removed, the naturalness results of the model for Chinese Japanese conversion were the lowest, at 2.83 and 2.82, respectively. In Figure 7 (b), all models had consistent changes in the naturalness of speech synthesis during the Chienese to Japanese and English to Japanese conversions. When replacing the time predictor values, the result values were 3.71, 3.68, and 3.71, indicating that the multi-lingual AI speech synthesis model had the best naturalness effect in language conversion.

Further analysis was conducted on the cross-lingual speech synthesis model based on similarity, and the results are shown in Figure 8.



(a) The similarity results of Chinese English conversion using different model methods

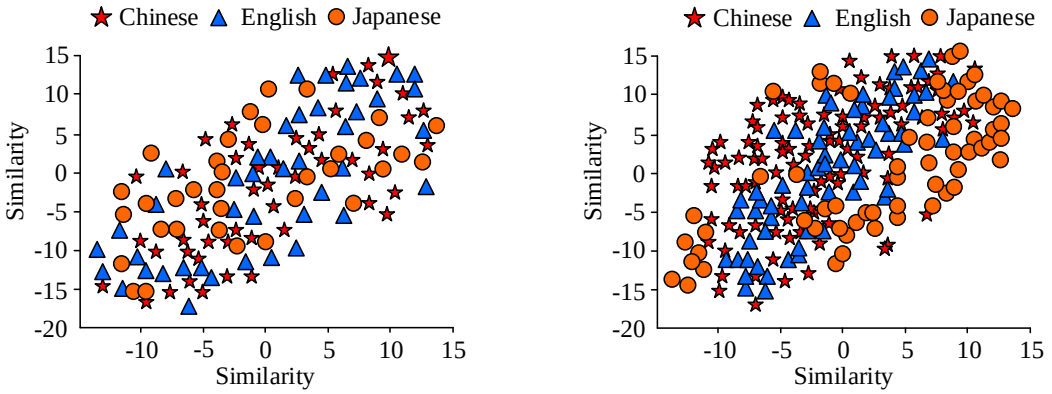


(b) The similarity results of English Japanese conversion using different model methods

Figure 8: Similarity MOS results of cross-lingual speech synthesis model

From Figure 8 (a), the similarity of all models for Chinese English conversion was below 4.0. Among them, the similarity MOS values of our model for Chinese English conversion, Chinese Japanese conversion, and English Chinese conversion were 3.96, 3.94, and 3.95, respectively, which were significantly higher than other improved models. In Figure 8 (b), among all the improved synthesis models for cross-lingual conversion, the one with the lowest similarity MOS was the one with zero vector removed. Its MOS values for English Japanese conversion, Japanese Chinese conversion, and Japanese English conversion were 2.67, 2.76, and 2.87, respectively. Therefore, it indicated that the removal of the zero-vector module had a significant impact on the cross-lingual sound synthesis model in cross-lingual training, making the synthesis model less fluent in expressing the rhyme of language input, and the overall naturalness and similarity were lower than those of single language synthesis. This also proved that the proposed model had good fluency and authenticity in cross-lingual sound synthesis.

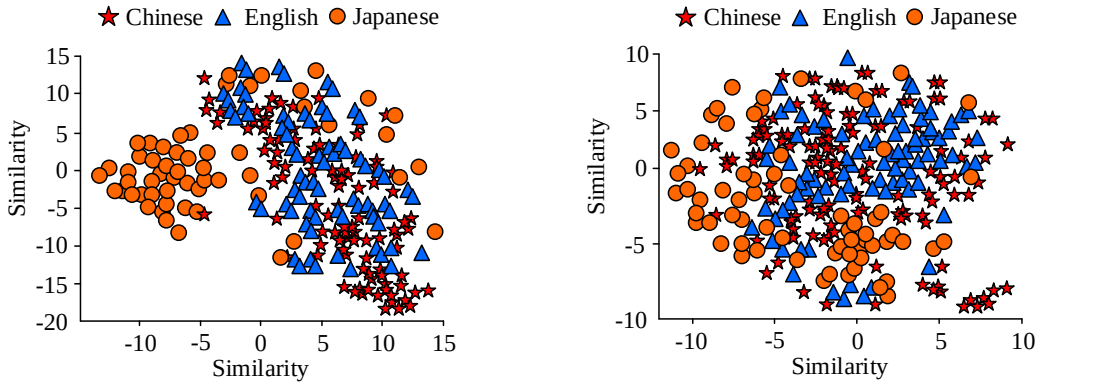
Afterwards, to minimize the coupling between the speech input and language information, an unsupervised nonlinear technical analysis was conducted on the proposed model and its improvement scheme, using t-distributed stochastic neighbor embedding (t-SNE), as shown in Figure 9.



(a) The t-SNE diagram of the model in this article (b) Remove t-SNE graph of domain adaptors

Figure 9: t-SNE diagram of the model in this article and the method of removing domain adaptors

From Figure 9 (a), the decoupling degree of the proposed model for speech embedding information in different languages was relatively uniform. The t-SNE graph of the three languages showed a good proportion of similarity distribution, which was basically within the range of -15 to 15. In Figure 9 (b), after removing the domain adaptors, the degree of information decoupling in Chinese language was mainly concentrated between -20 and 15, and the distribution in the left domain was relatively dense. However, the decoupling of language information in Japanese was not thorough enough, and the similarity distribution was quite extreme, mainly between 10 and 15. The degree of information decoupling in English language was relatively uniform compared to other languages, with a similarity between -10 and 10. However, the t-SNE graph obtained by removing regularization terms and language embeddings from the research model is shown in Figure 10.



(a) T-SNE graph with regularization term removed (b) Remove t-SNE graph with language embedding

Figure 10: T-SNE graph with regularization term and language embedding removed

From Figure 10 (a), the model without regularization terms had poor decoupling of language information. Among them, the similarity information of Japanese was more concentrated, which was different from other languages, with a value range of -10 to 5, and the horizontal distribution range was relatively dense, with a value range of -10 to -5. The similarity between Chinese and English languages mainly ranged from -5 to 15, and Chinese language was more densely distributed than English language. In Figure 10 (b), the model without language embeddings had a relatively uniform distribution of t-SNE for languages, and the overall distribution of the three languages was irregular with a wide range. The decoupling degree of Japanese language was mainly on the left and bottom, with values ranging from -10 to 10 and horizontally distributed between -10 and 5. The distribution of English languages was relatively concentrated, ranging from -5 to 5, while the distribution of Chinese languages was relatively even and had a wide range, ranging from -10 to 10. The visualization results of t-SNE above demonstrate that the proposed multilingual AI speech synthesis model has the best decoupling degree for language information.

Finally, different models were used to analyze the accuracy, real-time performance, and scalability of three mcross-lingual synthesized speeches. The Tacotron 2 speech synthesis technology of the spectrogram prediction network and the bidirectional encoder representation from transformers (BERT) technology of the transformer were selected. Tacotron 2 speech synthesis technology is an end-to-end speech synthesis framework that introduces attention mechanisms to address the limitations of traditional models in processing long sequences, improving the quality and naturalness of speech synthesis. In addition, its framework can directly convert text into speech during

the training process. However, BERT technology is a pre trained language model based on a bidirectional encoder representation architecture. In the technology of text to speech conversion, it can process text to adapt to the scene and user needs. In text to speech output, more natural and accurate speech can be generated. Compared with the model proposed in this article, the results are shown in Table 3.

Table 3: Performance comparison of different models for cross-lingual speech synthesis

Model	Cross-lingual speech synthesis	Data volume and performance results					
		50		100		200	
		Accuracy (%)	Real time(s)	Accuracy (%)	Real time (s)	Accuracy (%)	Real time (s)
Tacotron 2 speech synthesis framework	Chinese to English	89.03%	5.89s	86.94%	8.12s	79.38%	13.23s
	Chinese to Japanese	87.11%	5.48s	89.71%	8.07s	78.05%	13.49s
	English to Chinese	87.62%	4.19s	84.02%	9.54s	76.98%	15.84s
BERT speech synthesis	Chinese to English	88.67%	5.01s	87.49%	6.49s	83.54%	11.57s
	Chinese to Japanese	87.84%	5.04s	87.55%	5.13s	83.69%	11.62s
	English to Chinese	84.31%	5.62s	86.31%	5.98s	82.11%	10.48s
This article's model	Chinese to English	98.45%	0.68s	95.48%	1.15s	93.67%	3.05s
	Chinese to Japanese	98.36%	0.59s	96.16%	1.54s	94.58%	3.12s
	English to Chinese	97.26%	1.01s	94.59%	2.07s	93.24%	4.08s

From Table 3, as the amount of input speech data increased, the accuracy and real-time performance of the model gradually decreased, while scalability was also related to the amount of data and real-time conversion. Among them, the accuracy of Tacotron 2 speech synthesis technology was the lowest, and when the data volume was 50, the accuracy of Chinese to English conversion was the highest at 89.03%. The speech synthesis framework of BERT had the highest accuracy of 87.49% among 100 data sets. The model proposed by the research had an accuracy rate of up to 94.58% in 200 data sets, and the fastest real-time conversion time was 3.05 seconds, thus proving the superiority of the improved model in multilingual synthesis.

The signal-to-noise ratio (SNR) and Mel Cepstral Distortion (MCD) of 100 speech and audio data were further objectively evaluated to validate the results of MOS and the synthesis quality of the model. The calculation of SNR involved using analytical instruments to measure the power of audio signals and background noise. The higher the value, the stronger the relative strength of the audio signal to the background noise, resulting in more reliable and accurate detection results. MCD is the difference between the Mel frequency cepstral coefficients of synthesized speech and initial speech. After weighted averaging, the smaller the difference, the higher the similarity between synthesized speech and initial natural speech, and the better the speech quality. The results are shown in Table 4.

Table 4: Objective evaluation results of three models on 100 speech audio datasets

Model	Cross-lingual speech synthesis	SNR（dB）	MCD
Tacotron 2 speech synthesis framework	Chinese to English	4.9583 dB	14.56
	Chinese to Japanese	5.1389 dB	13.29
	English to Chinese	4.0534 dB	14.09
BERT text to speech	Chinese to English	4.1059 dB	12.38
	Chinese to Japanese	4.6615 dB	13.47
	English to Chinese	4.8915 dB	12.95
This article's model	Chinese to English	5.2649 dB	9.82
	Chinese to Japanese	5.4184 dB	8.96
	English to Chinese	5.1123 dB	9.51

According to Table 4, the average SNR of the Tacotron 2 speech synthesis framework for three language conversions was 4.7169, The average SNR of BERT technology was 4.5530. The model proposed in the study had a high SNR for speech and audio, which was 0.7122 higher than the average SNR of BERT technology, indicating that its model has accurate detection results for synthesized speech. In MCD calculation, the proposed model was much lower than other models, with MCD values of 9.82, 8.96, and 9.51 in Chinese English, Chinese Japanese, and English Japanese speech and audio, respectively. The above results demonstrate that the proposed sound synthesis model has a higher similarity to natural speech and better synthesis quality.

4 Discussion

Speech synthesis technology has become an important part of human-computer interaction in the field of AI. Research proposed a multi-lingual AI speech synthesis model based on domain adaptors, and module improvements were made to the VITS model to adapt to the technical operations of multilingual conversion. In the context of multi-lingual background and application requirements, domain adaptors were incorporated into the VITS model and other module structures were improved to solve the coupling relationship between speech input phonemes and language information, thereby improving the quality and output of multilingual speech synthesis. This is because domain adaptors use classifiers and predictors to weight speech datasets in the structure of adversarial neural networks, which can improve the model's generalization ability and add a regularization loss term to the subsequent decoupling process. The regularization loss term utilized the structure and convolutional layers of neural networks to separate the audio and content of the synthesized model, thereby achieving decoupling of language information in the audio input domain. After analyzing the evaluation indicators of speech synthesis technology, it was found that the improved speech synthesis model had the highest naturalness MOS of 4.73 and similarity of 4.93 in the Chinese language. However, in cross-lingual conversion, the naturalness and similarity of the synthesized model decreased due to significant differences in pronunciation methods such as factors, rhythm, and rhyme between different languages. At the same time, the speech conversion process also needed to consider the duration of input and output, which made the speech synthesis not smooth and realistic enough. The model proposed in the study had a synthetic similarity MOS of 3.96 in English, which was lower than the similarity of 4.93 and 4.94 in Chinese and English language synthesis. Compared with Li et al., the extraction of contextual information features and natural language classification problems were improved. This is because BERT's ability to recognize the structure of natural language processing was still limited by the model's error correction codes [18]. The multi-lingual speech synthesis model proposed in the study could achieve the task of text editor through language embedding when integrating AI scenes, processing input audio to improve the naturalness of generation. The random duration predictor could reduce the instability in the speech synthesis process and combine it with monotonic alignment search to achieve the alignment of the target sound. Therefore, in AI applications, cross-lingual speech synthesis models can not only separate the input voice and content while outputting, but also ensure the fluency and authenticity of AI synthesized sounds across multiple scenarios and languages.

5 Conclusion

Cross-lingual speech synthesis technology is currently the main research direction in the field of speech synthesis. The VITS model and its algorithm were studied, and module technology and domain adaptors were improved to construct a domain adaptive multilingual AI speech synthesis model. According to the analysis of the synthesis indicators of Chinese, English, and Japanese languages, the naturalness MOS of these three languages in the meta learning sound synthesis model were 3.91, 3.81, and 3.93, respectively. By analyzing the synthesis indicators of Chinese, English, and Japanese languages, a meta learning speech synthesis model was obtained, with a naturalness MOS of 3.91, 3.81, and 3.93 for Chinese, English, and Japanese, respectively. The evaluation of the proposed speech synthesis model for the three languages was 4.73, 4.22, and 4.42, respectively. In the cross-lingual speech synthesis experiment, the improved synthesis model that removes zero vectors and unified phonemes achieved naturalness MOS values of 2.83 and 2.82 for Chinese to Japanese conversion, respectively. The model proposed in this paper had the highest similarity MOS values for Chinese English, Chinese Japanese, and English Chinese, which were 3.96, 3.94, and 3.95, respectively. After removing the domain adaptors, the degree of information decoupling in Chinese language was mainly concentrated between 0 and 10, and the t-SNE graph distribution was uneven, indicating the feasibility of a multilingual AI speech synthesis model based on domain adaptors. In the cross-lingual conversion test, the accuracy of our model for language conversion was above 93%, and the conversion efficiency was the fastest. However, the BERT model was affected by a large amount of data resources during training and computation, which reduced its synthesis processing performance. Therefore, it proves the superiority of the research model. However, research on the quality of synthesized audio across languages still lacks specific factors to consider, and there is a lack of intervention in issues such as accents for language input types. There is also a lack of corresponding constraints for model training. Therefore, in future research, the text model in the early stage should be refined to train high-quality phoneme and audio data, and a reference module should be added to constrain the accent of the voice input to improve the quality of synthesized audio. In addition, in the parameter training process of multi-lingual cross-speech

synthesis technology, the idea of distillation model can be borrowed to effectively reduce the training workload of the synthesis model and ensure the feasibility and accuracy of the model.

Fundings

The research is supported by Shanxi Provincial Education Science "14th Five Year Plan" 2023 Education Digitalization and Teaching Reform Research Special Project, Research on the Teaching Path of "Dual-Qualification Teacher Classroom" Assisted by Virtual Digital Human, Project Number: SZH-230013.

References

[1] Sachdeva S, Ruan H, Hamarneh G, Behne D M, Jongman A, Sereno J A, Wang Y. Plain-to-clear speech video conversion for enhanced intelligibility. International journal of speech technology, 2023, 26(1): 163-184. https://doi.org/10.1007/s10772-023-10018-z

[2] Maruoka M, Tsujimura S, Asakura T. Effects of Artificial Synthetic Speech Control of SNR and Speech Rate on the Intelligibility of Train Station Announcements. Acoustics Australia, 2024, 52(1): 77-86. https://doi.org/10.1007/s40857-023-00306-8

[3] Revathi A, Sasikaladevi N, Arunprasanth D, Amirtharajan R. A Strategic Approach for Robust Dysarthric Speech Recognition. Wireless Personal Communications, 2024, 134(4):2315-2346. https://doi.org/10.1007/s11277-024-11029-y

[4] Jiang C, Gao Y, Ng W W Y, Zhou J, Zhong J, Zhen H, Hu X. Semantic dependency and local convolution for enhancing naturalness and tone in text-to-speech synthesis. Neurocomputing, 2024, 608(12): 1-11. https://doi.org/10.1016/j.neucom.2024.128430

[5] Liu Y, Wei L F, Qian X C. M3TTS: Multi-modal text-to-speech of multi-scale style control for dubbing. Pattern recognition letters, 2024, 179(3):158-164. https://doi.org/10.1016/j.patrec.2024.02.005

[6] Long L, Liang T. Multi-Distributed Speech Emotion Recognition Based on Mel Frequency Cepstogram and Parameter Transfer. Chinese Journal of Electronics, 2022, 31(1): 155-167.

[7] Ghosh S, Sarkar S, Ghosh S, Zalkow F, Jana N D. Audio-visual speech synthesis using vision transformer–enhanced autoencoders with ensemble of loss functions. Applied Intelligence, 2024, 54(6):4507-4524. https://doi.org/10.1007/s10489-024-05380-7

[8] Okamoto T, Matsubara K, Toda T, Shiga Y, Kawai H. Neural speech-rate conversion with multispeaker WaveNet vocoder. Speech Communication, 2022, 138(3): 1-12. https://doi.org/10.1016/j.specom.2022.01.003

[9] Masood M, Nawaz M, Malik K M, Javed A, Irtaza A, Malik H. Deepfakes generation and detection: state-of-the-art, open challenges, countermeasures, and way forward. Applied Intelligence, 2022, 53(4): 3974-4026. https://doi.org/10.1007/s10489-022-03766-z

[10] Zhang H, Yang S, Zhu H. CJE-TIG: Zero-shot cross-lingual text-to-image generation by Corpora-based Joint Encoding. Knowledge-based systems, 2022, 239(3): 108006-108017. https://doi.org/10.1016/j.knosys.2021.108006

[11] Kaur N, Singh P. Conventional and contemporary approaches used in text to speech synthesis: a review. Artificial Intelligence Review, 2022, 13(11): 1-44. https://doi.org/10.1007/s10462-022-10315-0

[12] Bahraini T, Sadigh A N. Proposing a robust RLS based subband adaptive filtering for audio noise cancellation. Applied acoustics, 2024, 216(1):109755-109765. https://doi.org/10.1016/j.apacoust.2023.109755

[13] Wang J, Yue K, Duan L. Models and techniques for domain relation extraction: a survey. Journal of Data Science and Intelligent Systems, 2023, 1(2): 65-82. https://doi.org/10.47852/bonviewjdsis3202973

[14] Higuchi Y, Moritz N, Roux J L, Hori T. Momentum Pseudo-Labeling: Semi-Supervised ASR With Continuously Improving Pseudo-Labels. IEEE Journal of Selected Topics in Signal Processing, 2022, 16(6): 1424-1438. https://doi.org/10.1109/jstsp.2022.3195367

[15] Zhou Y, Ding Z, Liu X, Shen C, Tong L, Guan X. Infer-AVAE: An attribute inference model based on adversarial variational autoencoder. Neurocomputing, 2022, 483(4): 105-115. https://doi.org/10.1016/j.neucom.2022.02.006

[16] Meng J, Zhu F. Seek for commonalities: Shared features extraction for multi-task reinforcement learning via adversarial training. Expert Syst. Appl. 2023, 224(8): 119975-119986. https://doi.org/10.1016/j.eswa.2023.119975

[17] Haifeng Z, Fengqian Z, Shengtian S, Li Y, Li X, Hu K, Chen Y. An unsupervised intelligent fault diagnosis research for rotating machinery based on NND-SAM method. Measurement Science & Technology, 2023, 34(3): 35906-35922. https://doi.org/10.1088/1361-6501/aca98f

[18] Li S, Hu X, Huang Z Z J. ECC-BERT: Classification of error correcting codes using the improved bidirectional encoder representation from transformers. IET communications, 2022, 16(4): 359-368. https://doi.org/10.1049/cmu2.12357

Multi-Objective Optimization for Human Resource Allocation Using Reinforcement Learning and Enhanced Cuckoo Search Algorithm

Liuyang Qiao, Le Li*, Shanshan Yu
School of Guangdong Country Garden Polytechnic, Qingyuan 511500, China
E-mail: Linyichuan0220@163.com
*Corresponding Author

Keywords: human resource allocation, optimization algorithm, rhododendron search, reinforcement learning, multi-objective optimization, imperialist competition

Received: December 5, 2024

In today's fiercely competitive business environment, enterprises are increasingly relying on efficient human resource allocation to improve operational efficiency and reduce operating costs. To better allocate human resources, this study proposes a multi-objective imperialist competition algorithm that integrates optimized Cuckoo Search algorithm and reinforcement learning, and creates a new human resource allocation optimization model. The new model can effectively explore solution space and adapt to talent allocation under different conditions by simulating the parasitic behavior of cuckoos and competition between empires. The results indicated that the new model performed the best when the population size was 50, the number of ruling countries was 40, the task exchange probability was 0.1, the resource replacement probability was 0.05, the colonial power coefficient was 0.2, and the number of colonies was 2. The average ideal distance of the mixed integer non-derivative optimization algorithm was 0.71, the diffusion of non-dominated solutions was 0.73, the momentum volume was 0.77, and the average response time of the solution was 2.43s. The indicators corresponding to the new model were 0.69, 0.76, 0.78, and 0.51s, respectively. Compared with the mixed integer non-derivative optimization algorithm, the new model reduced the average ideal distance by 0.02 and improved the diffusion of non-dominated solutions by 0.03. In addition, the momentum volume increased by 0.01, the average response time for solving was 0.51s, and the speed increased by 1.92s, all of which were better than the comparative algorithms. The average score of the new model after allocating human resources was above 9. The data showed that the new model had good convergence, fast solving speed, stable and high-quality results, and could effectively allocate talents. The research model has important practical significance for improving the efficiency of human resource scheduling and decision-making quality.

Povzetek: Raziskava predstavi optimizacijski model IDCS-OMOICA za razporejanje kadrov, ki s kombinacijo okrepljenega iskanja kukavičjih gnezd in učenja s krepitvijo izboljša učinkovitost, odzivnost in raznolikost rešitev.

1 Introduction

In the contemporary epoch of the knowledge economy, knowledge and information have emerged as the paramount factors of production. The core competitiveness of enterprises is increasingly contingent on the knowledge, skills, and innovation abilities of their employees [1]. With the continuous advancement of globalization and technological innovation, enterprises are facing more intense market competition and constantly changing business needs [2]. On the one hand, enterprises need to ensure sufficient talent support for key positions to maintain daily operations and achieve short-term goals [3]. On the other hand, companies also need to invest in the long-term development of their employees to build sustainable competitive advantages [4]. Traditional Human Resources (HR) management methods are no longer able to meet the needs of modern enterprises. Enterprises must find new ways to attract, retain, and develop talent while ensuring that the allocation of HRs maximizes work efficiency and employee satisfaction [5]. Based on this, how to effectively manage and configure HR to quickly adapt to market changes and meet corporate strategic goals has become a common focus of attention for enterprise managers and scholars [6]. Seifi H et al. proposed a Grey Wolf Optimization (GWO) algorithm based on Sugeno fuzzy inference model to optimize the allocation of HR in small and medium-sized enterprises. This method significantly improved model performance in larger B100 and B200 datasets, and could obtain the best allocation results and solutions [7]. Dabirian S et al. proposed a dynamic model to effectively allocate labor to solve the problem of HR allocation in construction projects. This model could accurately estimate labor demand and effectively allocate it to ensure timely supply and

distribution of project labor [8]. Rodgers W et al. constructed a throughput model framework to address ethical decision-making in HR management, describing the algorithm's individual decision-making process in the HR management environment. The new model demonstrated how perception, judgment, and information usage affect strategy selection, and identified how adopting certain ethical decision-making algorithm paths can support different strategies [9]. Waldkirch M et al. proposed a clustering-based incremental association rule mining algorithm to improve the efficiency of data mining and incorporated it into the HR management system of universities. The experimental results showed that the algorithm achieved visualization of HR information system data and could effectively mine large-scale databases [10]. Tarafdar et al. proposed a theoretical framework to explore the role interaction between humans and algorithms in algorithmic work, thereby addressing human-machine interaction in task allocation. Algorithms lacked transparency in task allocation, task tracking, and performance evaluation. They provide a new perspective for understanding algorithm management in HR allocation and emphasizing the need to focus on human experience and feedback when designing algorithms [11]. Park H et al. proposed a stakeholder centered solution to alleviate the tense relationship between artificial intelligence and stakeholders in HR management. This plan has successfully eased these tensions by focusing on stakeholders and promoting harmony among various stakeholders [12]. Nikzad et al. proposed a two-stage random mixed integer model that considers multiple factors simultaneously to better allocate HR in the healthcare system, considering the uncertainty of travel and service times. The test results have demonstrated the algorithm's ability to solve large instances [13].

Table 1: Summary and comparison table of various studies.

Study	Method	Advantages and Limitations
Seifi H et al. [7]	GWO based on Sugeno Fuzzy Inference Model	The model performance significantly improves in larger B100 and B200 datasets, achieving optimal allocation results and solutions.
Dabirian S et al. [8]	Dynamic Model	Accurately estimates labor demand and effectively allocates resources to ensure timely labor supply and distribution in construction projects.
Rodgers W et al. [9]	Throughput Model Framework	Demonstrates how perception, judgment, and information use influence strategy selection and identifies how certain ethical decision-making algorithm paths support different strategies.
Waldkirch et al. [10]	Clustering-based Incremental Association Rule Mining Algorithm	Achieves visualization of human resource information system data and effectively mines large-scale databases.
Tarafdar M et al. [11]	Theoretical Framework	Explores the role interaction between humans and algorithms in algorithmic work, emphasizing the need to consider human experience and feedback in algorithm design.
Park H et al. [12]	Stakeholder-Centric Solution	Promotes harmony among various stakeholders, alleviating tensions surrounding algorithmic evaluation in human resource management.
Nikzad E et al. [13]	Two-Stage Stochastic Mixed-Integer Model	Considers uncertainties in travel and service times, with experimental results proving the algorithm's capability to solve large-scale instances.
Proposed in This Study	IDCS-OMOICA	Exhibits good convergence, fast solving speed, stable and high-quality results, effectively allocating talent. The average ideal distance, diffusion of non dominated solutions, momentum volume, and average response time for solving are 0.69, 0.76, 0.78, and 0.51 seconds, respectively

Compared with other results, the proposed method provides more indicators to quantify the performance of the algorithm, making it easier to evaluate and compare (Table 1). The solving speed is 0.51s, suitable for fast response scenarios such as real-time HR allocation. The findings are consistent and of superior quality, demonstrating the capacity to produce varied and high-quality solution sets. This property is of particular significance for multi-objective optimization problems, as it overcomes the challenge of unstable results that may be present in alternative methods. This method not only performs well in theory but also has important practical application significance, which can effectively improve the efficiency of HR scheduling and decision quality. Although those proposed algorithms have achieved certain results in their respective applications, there are still some shortcomings. For example, the GWO algorithm may face slow convergence speed and local optima when dealing with complex multi-objective problems, and its prediction accuracy may be affected by data quality. Although dynamic models can effectively allocate labor, they may not be flexible enough to respond to rapidly changing demands. The Cuckoo

Search (CS) algorithm has strong global search capability. To address these issues and allocate HR efficiently and quickly, this study combines the Improved Discrete Cuckoo Search (IDCS) algorithm and the Optimized Multi-Objective Imperialist Competition Algorithm (OMOICA) based on Reinforcement Learning (RL) optimization to form a new HR configuration optimization model (IDCS-OMOICA). The innovation lies in adjusting search strategies in real-time through RL mechanisms to respond to changes in project requirements and HR conditions. The search parameters are automatically adjusted through Q-Learning to reduce human intervention and improve the robustness and adaptability of the model. Efficient global search is achieved through Lévy flight and random walk mechanisms. The local optimization is carried out using Q-Learning and Empire Competition mechanisms to improve computational speed. Through the above improvement methods, the GWO algorithm has been studied to solve the problems of slow convergence speed, local optima, and prediction accuracy affected by data quality when dealing with complex multi-objective problems. At the same time, it can also adapt to the flexibility of rapidly changing demands. The corresponding HR Scheduling Management System (HRSMS) has been developed, and the algorithm results have been transformed into practical and operable tools, providing technical support for enterprises. Its six-layer architecture design ensures the efficiency, security, and scalability of the system.

2 Method

2.1 Fusion of IDCS Algorithm and OMOICA

In today's rapidly developing industrial and social environment, HR Configuration Optimization (HRCO) has become a key factor in improving organizational efficiency and reducing costs [14]. Especially in multi-project environments, how to effectively schedule limited HR to meet the specific needs of each project has become a complex and challenging problem. Some algorithms, such as genetic algorithms, have complex operations such as encoding, crossover, and mutation, with multiple parameter settings, requiring a large amount of computing resources and time. The problem of HR allocation has a lot of data and many constraints, making it difficult for these algorithms to process efficiently. Traditional algorithms such as gradient descent rely on gradient information and are prone to getting stuck in local optima. HR allocation problems have multi-modal characteristics, making it difficult for these algorithms to globally search for optimal solutions. The CS algorithm has strong global search capabilities and can explore the vast solution space of HR configuration. RL algorithms can refine and optimize strategies through trial-and-error learning in this solution space to adapt to changes in project requirements and uncertainties in the work environment. For example, a software company is facing an urgent project with a sudden increase in demand, while some employees are absent from training. RL considers this change as a new state and dynamically adjusts configuration strategies based on historical data and real-time feedback, such as project progress, employee skills, and availability. RL interacts with the environment (project and HRs system), with the agent (scheduling system) selecting actions (assigning tasks), and the environment providing rewards (project completion on time, cost control). Intelligent agents learn optimal strategies, adjust configurations in real-time, ensure smooth project progress, and efficiently allocate HRs. However, traditional CS algorithms may encounter slow convergence speed and be prone to getting stuck in local optima when dealing with complex HR configuration problems. Based on this, to effectively allocate HRs in complex situations and meet the needs of various departments, this study optimizes the CS algorithm. The structure of the IDCS algorithm is shown in Figure 1.

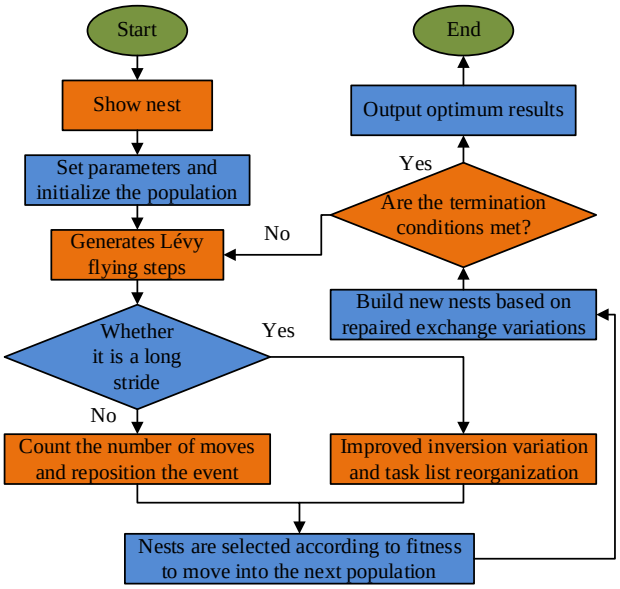


Figure 1: IDCS algorithm structure diagram.

In Figure 1, the IDCS algorithm is mainly used to solve complex optimization problems, such as resource-constrained multi-project scheduling problems. IDCS explores the solution space and finds the optimal solution through Lévy flight and random walk mechanism. The Lévy Flight Step Size (LFSS) has a heavy tail characteristic, which means there is a high probability of large step jumps occurring during the search process. This approach enables the algorithm to rapidly explore a more extensive search space in a short period, thereby avoiding the potential stagnation in local optima. Additionally, it facilitates a more comprehensive consideration of the allocation of HRs from a macro perspective, ultimately leading to the identification of a more optimal overall configuration. Random walk is a mathematical statistical model of irregular motion, where neighboring points are randomly selected for comparison each time. It enables the algorithm to perform detailed local searches in relatively concentrated areas, fine-tune and optimize existing HR allocation schemes, and improve the accuracy and adaptability of the schemes. The structure of IDCS mainly includes steps such as initializing the population, evaluating fitness, Lévy flight updating the nest, random walk generating new nests, and selecting the next generation population. In the initialization phase, the algorithm generates an initial population, with each individual representing a solution. In the fitness evaluation stage, the algorithm calculates the fitness value of each individual, which is their quality as a solution. Lévy flight is a key component of IDCS. IDCS updates the solution by simulating the behavior of cuckoos searching for new nests [15]. In this process, the algorithm uses an Lévy distribution to generate step sizes, allowing the algorithm to conduct detailed searches in local areas as well as wide area explorations on a global scale. The LFSS is calculated using the Mantegna algorithm, and the calculation formula is shown in equation (1) [16].

$$L = \frac{|x_i|}{\mu}\left(\frac{\Gamma(1+\lambda)\sin(\pi\lambda/2)}{\Gamma((1+\lambda)/2)}\lambda^{-\lambda}\right)^{1/\lambda} \quad (1)$$

In equation (1), L represents LFSS. Γ is the gamma function. λ is a parameter of the step size distribution. x_i and μ are variables in the algorithm. When LFSS is between [0,2], small step length movements are performed, and the relationship between the number of movements and the step length is shown in equation (2) [17].

$$\begin{cases} M_s = \left\lceil \frac{L \square D}{3} \right\rceil, L \in [0,1] \\ M_s = \left\lceil \frac{(L+1) \square D}{3} \right\rceil, L \in (1,2] \end{cases} \quad (2)$$

In equation (2), M_s is the number of small step movements. D is the dimension of the solution space. $\lceil\ \rceil$ is rounded up. When LFSS is greater than 2, large step jumps are executed. The number of jumps is determined by the step size, and the calculation method is shown in equation (3).

$$M_l = \left\lfloor \frac{L}{2} \right\rfloor \quad (3)$$

In equation (3), M_l is the number of jumps, and $\lfloor\ \rfloor$ is rounded down. In the random walk stage, the algorithm eliminates solutions with poor fitness with a certain probability and generates new solutions to replace them. Finally, the algorithm selects the best individual based on fitness to enter the next generation, iterating continuously until the termination condition is met, such as reaching a predetermined number of iterations or the quality of the solution reaching a certain threshold. The formula for adaptively adjusting the elimination probability based on individual fitness values is shown in equation (4) [18].

$$p_a = 1 - e^{-(f(x_i)/\bar{f})^{-1.1}} \quad (4)$$

In equation (4), p_a is the elimination probability. $f(x_i)$ is an individual's fitness value. $\bar{f}$ is the average fitness value in the population. The IDCS algorithm can effectively balance global search capability and local search accuracy through iterative search, thereby improving the probability of finding the global optimal solution. In multi-objective problems, it is necessary to adopt an algorithm with multi-objective processing capability and integrate it with IDCS [19]. HR allocation needs to take into account multiple objectives such as cost, efficiency, and employee satisfaction. OMOICA, through strategies such as Pareto dominance, can simultaneously optimize multiple conflicting objectives, find an equilibrium solution set, and meet comprehensive optimization needs. However, other algorithms such as IDCS are relatively weak in multi-objective coordination. The problem of HR allocation is complex and ever-changing, with flexible OMOICA parameters that can adapt to different scales and constraints, and quickly respond to changes in enterprise needs. Other multi-objective algorithms, such as multi-objective genetic algorithms, have complex parameter settings, relatively poor adaptability, and are difficult to maintain stable performance in changing scenarios. Therefore, this study uses the OMOICA to search for targets with multiple constraints to meet the needs of HRCO, and the solving method is Pareto. Figure 2 shows the Pareto solution process.

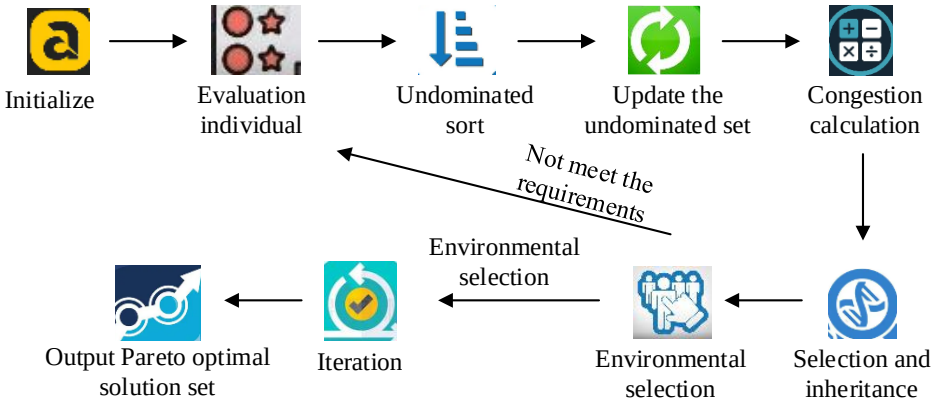


Figure 2: Pareto solution process diagram.

In Figure 2, the process first initializes the solution set, evaluates the objective function values of the solutions, and performs non-dominated sorting to determine the Pareto dominance relationship of the solutions. After updating the non dominated set, the crowding degree is calculated to maintain the diversity of solutions, followed by genetic operations and environmental selection to determine the next generation solution set. These steps are iterated until the Pareto optimal solution set is found. The definition of Pareto solution is shown in equation (5) [20].

$$\begin{cases} f_i(x_1) \le f_i(x_2) \\ f_j(x_1) < f_j(x_2) \end{cases} \quad (5)$$

In equation (5), x_1 and x_2 represent the solution values. If for all objective functions f_i, equation $f_i(x_1) \le f_i(x_2)$ is satisfied, and there is at least one objective function f_j that satisfies $f_j(x_1) < f_j(x_2)$, then it is called x_1 dominating x_2. The multi-objective optimization problem can be expressed as minimizing the objective function vector, as shown in equation (6).

$$F(x) = [f_1(x), f_2(x), ..., f_m(x)] \quad (6)$$

In equation (6), $F(x)$ is a set of vectors, f_m is a different objective function, and x is the solution value. The definition of feasible region constraints for multi-objective optimization problems is shown in equation (7).

$$D' = \{x \mid g_i'(x) \ge 0, i \in [1, M'], h_j'(x) = 0, j \in [1, L']\} \quad (7)$$

In equation (7), D' is the feasible region. $g_i'(x)$ is the i-th inequality constraint function. M' is the total number of inequality constraints. $h_j'(x)$ is the j-th equality constraint function. L' is the total number of equality constraints. i and j are indices that respectively traverse all inequalities and equality constraints. The crowding distance is used to measure the distribution density of solutions on the Pareto front, as shown in equation (8) [21].

$$d(x^*) = \sum_{k=1}^{m} \frac{f_k(x^*) - f_k^{\min}}{f_k^{\max} - f_k^{\min}} \quad (8)$$

In equation (8), $d(x^*)$ is the crowding distance. $f_k^{\max}$ and $f_k^{\min}$ are the maximum and minimum values of the k-th objective function. x^* is the distribution term. Figure 3 shows the structure of the OMOICA.

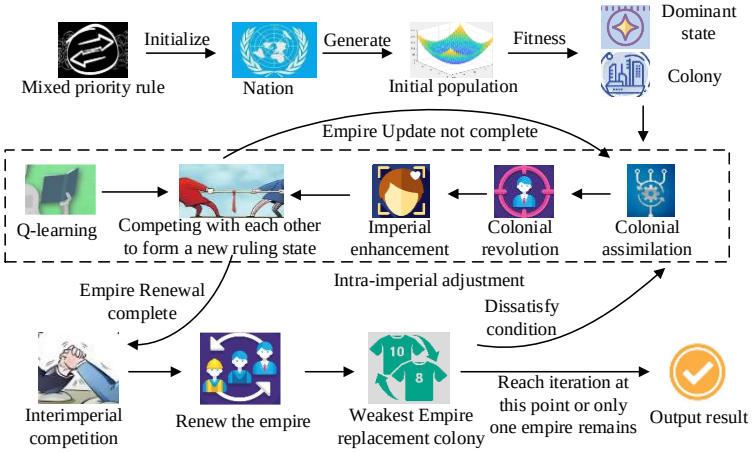


Figure 3: OMOICA structure diagram

In Figure 3, OMOICA is optimized using Q-Learning method (QRL). OMOICA first generates an initial population through mixed priority rules, dividing "countries" into "ruling countries" and "colonies", and enhancing the diversity and quality of solutions through assimilation and competition mechanisms. This process introduces a mechanism of learning from non-dominated solutions within external archives to enhance the diversity of the population. Next, the algorithm introduces QRL to optimize the search strategy. When the preset number of iterations is reached or all countries are unified into one empire, the algorithm terminates and outputs the result. The process of integrating IDCS and OMOICA mainly combines the efficient search capability of IDCS with the multi-objective optimization and RL capability of OMOICA. IDCS explores the solution space through Lévy flight and random walk, while OMOICA utilizes QRL and IDCS optimization strategies [22]. After the fusion of the two, the IDCS-OMOICA algorithm can find the optimal HR configuration solution that meets the requirements in a dynamic environment.

2.2 Construction of HR configuration optimization model

After successfully integrating IDCS and OMOICA, this study will construct an efficient IDCS-OMOICA model. This model will integrate the advantages of the aforementioned hybrid algorithms to adapt to the ever-changing industrial environment and achieve optimal resource allocation. The optimization problem of the IDCS-OMOICA model is to maximize organizational efficiency and employee satisfaction while meeting project requirements, employee skill matching, time constraints, cost-effectiveness, and other constraints. The purpose is to optimize the overall performance of the organization, fully realize the individual value of employees, and minimize the company's costs (including training time costs and employee turnover costs). The construction of IDCS-OMOICA model is a complex process that requires consideration of multiple factors, including project requirements, employee skills, time constraints, and cost-effectiveness. The structure of the IDCS-OMOICA model is shown in Figure 4.

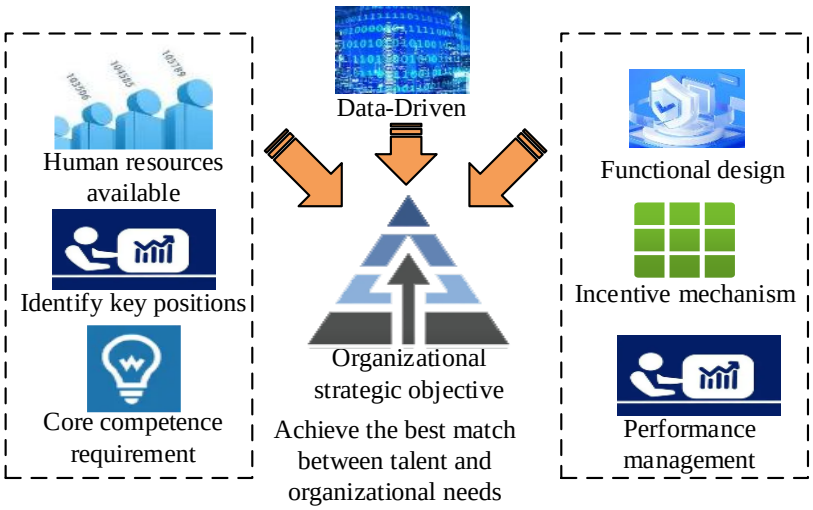


Figure 4: IDCS-OMOICA model structure diagram.

In Figure 4, the IDCS-OMOICA model is a systematic framework aimed at improving organizational effectiveness and employee satisfaction. This model focuses on organizational strategic goals and identifies key positions and core competency requirements by analyzing and evaluating the existing HR situation. It optimizes talent recruitment, training, promotion, and retention strategies through data-driven methods that combine market trends, employee performance and potential assessments, as well as organizational culture and values. The data-driven part is mainly achieved by integrating IDCS and OMOICA algorithms. The data used in "data-driven" include internal data such as employee performance, skill evaluation, project progress, as well as external data such as market trends, industry standards, etc. The data sources are extensive, covering enterprise information systems, market research, public datasets, etc. When enterprises compete for HR allocation, standardization cost is commonly used as one of the allocation indicators. Departments with low standardization costs have high resource utilization efficiency and require fewer personnel. Departments with high costs face significant cost pressures, so priority should be given to support allocation to ensure smooth project progress. The calculation of standardization cost is shown in equation (9).

$$C_n = c_n - \max_i \{c_i\} \quad (9)$$

In equation (9), C_n is the standardization cost of the n-th department or project group. c_n is the cost of key positions in the n-th core employee. $\max_i \{c_i\}$ is the maximum cost of key positions among all core employees. The expression for the relative power of core employees is shown in equation (10).

$$p_n = \frac{C_n}{\sum_{i=1}^{N_{\text{imp}}} C_i} \quad (10)$$

In equation (10), p_n is the relative power of the n-th core employee. N_{imp} is the total number of key positions. The equation for assigning new employees is shown in equation (11).

$$N_{C_n} = \text{round}\{p_n \times N_{\text{col}}\} \quad (11)$$

In equation (11), NC_n is the number of employees assigned to the n-th team or department that can be managed and trained, which is determined based on the relative strength p_n of the department or project group. N_{col} is the total number of available new employees. round is a rounding function used to calculate the number of new employees. $p_n \times N_{col}$ is converted to the nearest integer. The total cost calculation of the department or project team is shown in equation (12).

$$T_{C \cdot n} = f(imp_n) + \xi \times \frac{\sum_{i=1}^{N_{C_n}} f(col_i)}{N_{C_{in}}} \quad (12)$$

In equation (12), $T_{C \cdot n}$ is the total cost of the department or project team. $f(imp_n)$ is a core employee. $f(col_i)$ is a new employee. ξ is the weight coefficient. The search process for finding the optimal allocation plan is optimized using the RL method. The status represents the current allocation of HRs, including employee skills, project requirements, department status, etc. Action is a resource allocation decision based on the current state, such as assigning new employees, adjusting employee positions, etc. Rewards reflect the effectiveness of allocation strategies and are related to indicators such as project completion, employee satisfaction, cost-effectiveness, etc. The state value function is shown in equation (13).

$$V_\pi(s) = E_\pi[G_t \mid S_t = s] \quad (13)$$

In equation (13), $V_\pi(s)$ is the expected return following strategy π in state s. E_π is the expected value of a performance metric when following a specific strategy π. G_t is the total return starting from time t. S_t is the state of time. The action value function is shown in equation (14).

$$Q_\pi(s,a) = E_\pi[G_t \mid S_t = s, A_t = a] \quad (14)$$

In equation (14), $Q_\pi(s,a)$ is the expected reward for executing action a in state s and following strategy π. A_t is the action of time t. The Bellman Expectation Equation (BEE) can transform dynamic optimization problems into simple sub-problems, describing the relationship between the current state value and the next state value, and is a core concept in RL. BEE can provide a mathematical foundation for solving optimal strategies, helping to find strategies that maximize cumulative rewards by evaluating state value and action value. The state value function is shown in equation (15).

$$\begin{cases} V_\pi'(s) = \sum_{a \in A} \pi(a \mid s) Q_\pi(s,a) \\ Q_\pi'(s,a) = r(s,a) + \gamma \sum_{s' \in S} P(s' \mid s,a) V_\pi'(s') \end{cases} \quad (15)$$

In equation (15), s' and $Q_\pi'(s,a)$ are the state value function and action value function of the BEE, and $r(s,a)$ is the immediate reward for performing action a in state s. γ is the discount factor. $P(s' \mid s,a)$ is the transition probability from executing action a in state s to state s'. When a plan loses all new employees, the plan is abandoned, and ultimately, there is only one optimal HR allocation plan left, at which point the algorithm terminates. The IDCS-OMOICA model also involves redesigning workflows and responsibilities to ensure that employees can unleash their maximum potential. In addition, the model also includes continuous improvement of incentive mechanisms and performance management systems to achieve the best match between talent and organizational needs. Optimizing incentive mechanisms includes linking performance results with salary, promotion, and training to meet employees' material and spiritual needs. For example, companies can establish honors such as Outstanding Employee Awards and Team Awards to stimulate employees' sense of honor and belonging. Establishing an open and inclusive corporate culture can increase employee engagement and satisfaction. The optimization of performance management system includes clarifying goals, setting indicators, adopting diversified assessment methods, strengthening communication and feedback, continuous improvement, etc., to improve the operational efficiency and core competitiveness of enterprises. To facilitate HR scheduling management, this study establishes an HRSMS based on the IDCS-OMOICA model, and its system architecture is shown in Figure 5.

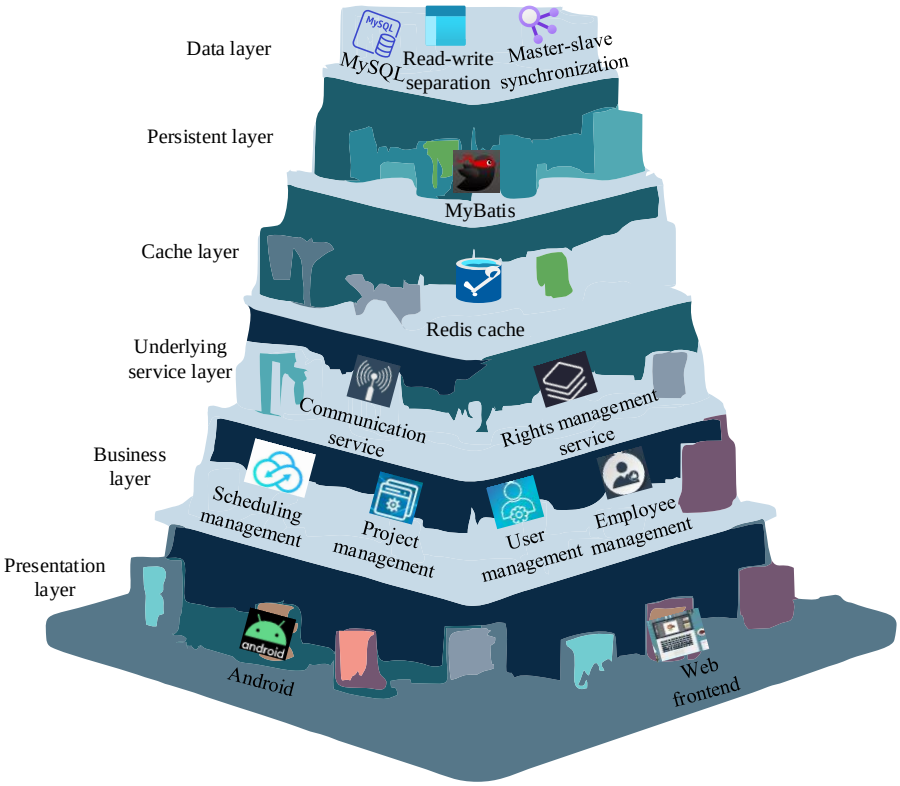


Figure 5: HR scheduling management system architecture diagram.

In Figure 5, HRSMS is built on top of a six-layer architecture, with the top layer being the user interface for web and mobile devices, used to display information and receive user operations. The business logic layer includes key functions such as user management, employee information, project data, and scheduling instructions. The scheduling management module in the business logic layer includes the IDCS-OMOICA model, which is the most core technical part of the entire system. The IDCS-OMOICA model analyzes HR data and allocates personnel based on user needs to derive HR allocation plans. The basic service layer provides permission verification and message services to ensure data security and timely communication. The caching layer utilizes Redis technology to optimize data access speed. The persistence layer works together with MySQL database through MyBatis to manage data storage and retrieval. The bottom data layer is mainly responsible for maintaining and processing data. The entire architecture aims to ensure the efficiency, security, and scalability of the system. The entire function of HRSMS is shown in Figure 6.

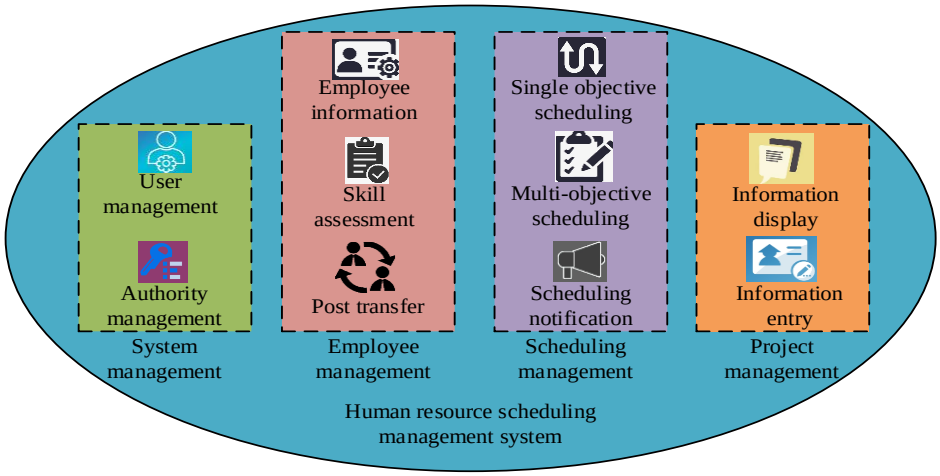


Figure 6: Function structure diagram of HRSMS.

The structure of Figure 6 highlights the main responsibilities and interactions of each module. The system management module is responsible for configuring users and permissions to ensure data security

and operational compliance. The employee management module provides employee information and skill assessment to assist in job adjustments. The project management module is responsible for displaying and entering project data, allowing users to have a clear understanding of tasks and resource requirements. The core scheduling management module integrates multiple optimization algorithms to solve complex HR scheduling problems, and displays results through intuitive tables and Gantt charts, combined with message push functions to ensure real-time updates of scheduling information. These four modules together form a complete functional system aimed at improving HR scheduling efficiency and decision quality.

3 Result

3.1 Analysis of optimal parameter settings for IDCS-OMOICA

The experimental results obtained by IDCS-OMOICA are closely related to the configuration of algorithm parameters. This study sets the Population Size (PS) for algorithm parameter configuration at five levels: 50, 100, 150, 200, and 250. There are five options for the Number Of Ruling States (NORS): 5, 10, 20, 30, and 40. The Task Switching Probability (TSP) has five gradients of 0.1, 0.2, 0.4, 0.5, 0.6, and 0.7. The Resource Replacement Probability (RRP) is divided into five levels: 0.05, 0.25, 0.45, 0.55, and 0.65. The colonial power coefficient ξ has five scales of 0.1, 0.2, 0.3, 0.4, 0.5, and 0.7. The number of colonies N_{col} ranges from 1 to 5 in five stages. For the convenience of chart drawing, the above parameters are uniformly numbered from 1 to 5. The algorithm learning rate is 0.1 and the discount factor is 0.5. These parameters together form the basis for algorithm optimization. These parameter ranges are determined based on practical experience and can cover common application scenarios and problem scales. Choosing these parameter ranges can adapt to problems of different complexities. For example, larger population sizes and more ruling countries are suitable for complex problems, while smaller values are suitable for simpler problems. Conducting experiments within these parameter ranges can quickly find the optimal or approximate optimal solution while avoiding excessive computational costs. PS represents the number of individuals in the initial population of the algorithm. Each individual represents a possible solution. NORS represents the number of excellent solutions in the Imperial Competition algorithm. TSP represents the probability of an algorithm switching from one task to another during execution. RRP represents the probability of replacing solutions with poor fitness during the random walk stage of the algorithm. In the course of experimental analysis of HR scheduling problems, the objective is to reduce environmental interference, ensure the accuracy and reliability of results, and maximize the algorithm's optimal performance. To this end, the experimental platform parameters are set as listed in Table 2.

Table 2: Parameters of the experimental platform.

Operating system	Ubuntu Server 20.04 LTS 64-bit
RAM	16GB
CPU	Quad-core AMD EPYC 7401P 2.0GHz
Database	MySQL-8.0.34
Cache	Memcached-1.6.9
Deploying container engines	Kubernetes-1.20.0
Background development framework	Quarkus-1.12.0
Front-end development framework	React-17.0.2
Css preprocessor	Sass-1.32.8
Database management framework	MyBatis
Message push middleware	RabbitMQ-3.9.10 (Web) Firebase Cloud Messaging (Android)

The performance evaluation criteria for algorithms are Mean Ideal Distance (MID), Spread of Non-dominated Solutions (SNS), and Hyper Volume (HV). MID is a measure of the gap between HR allocation plans and ideal states. A low MID value indicates that the plan is close to ideal, resource allocation is reasonable, and employee and organizational goals are highly matched. SNS is a diversity indicator for evaluating HR allocation plans. A high SNS value means that the solution is evenly distributed and can cover multiple demand scenarios, improving the organization's ability to respond to changes. HV reflects the overall quality and scope of HR allocation plans. A high HV value indicates that the solution set covers a wide range

and can optimize multiple objectives simultaneously, such as cost and efficiency. These performance indicators have been normalized. Figure 7 shows the performance indicators of the IDCS-OMOICA under different parameters.

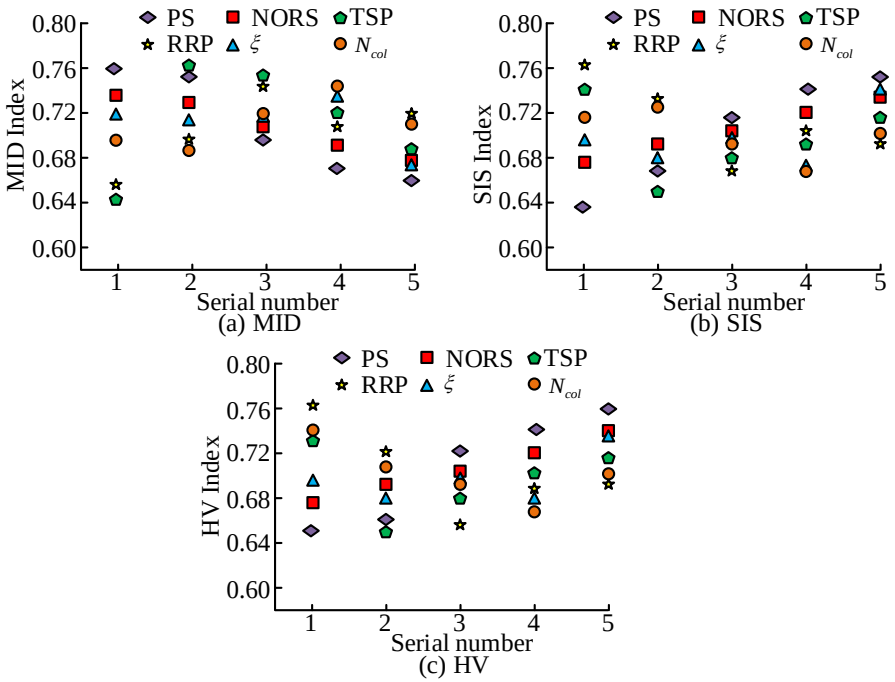


Figure 7: Test results of MID, SIS, and HV Performance indicators of IDCS-OMOICA under sequence number: 1-5.

In Figure 7 (a), taking the minimum MID value as the optimal criterion, when the PS of IDCS-OMOICA is 50, NORS is 40, TSP is 0.1, RRP is 0.05, ξ is 0.2, and N_{col} is 2, the MID values are 0.65, 0.66, 0.64, 0.66, 0.67, and 0.68. Low MID values help improve employee satisfaction and organizational performance, reducing resource waste. In Figure 7 (b), the algorithm parameters at the maximum SIS value are PS=50, NORS=40, TSP=0.1, RRP=0.05, ξ =0.2, and N_{col} =2. High SIS values can enhance organizational flexibility and innovation capabilities, adapting to changing environments. In Figure 7 (c), the algorithm parameters are also consistent with the maximum SIS value at the maximum HV value. High HV values can enhance an organization's comprehensive competitiveness under multiple objectives and achieve balanced development. Therefore, PS=50, NORS=40, TSP=0.1, RRP=0.05, ξ =0.2, and N_{col} =2 are the optimal parameter values for the IDCS-OMOICA. Under this parameter condition, the MID, SNS, and HV performance indicators of IDCS-OMOICA during training and testing are shown in Figure 8 after 400 iterations.

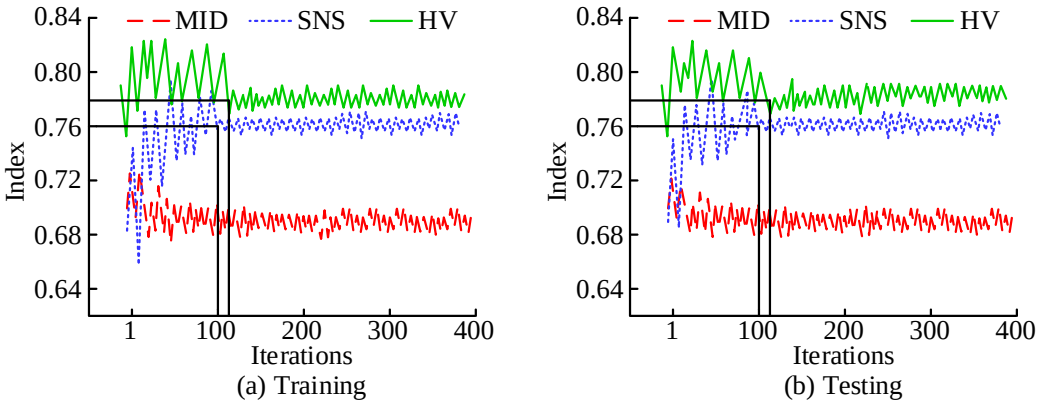


Figure 8: Comparison results of MID, SNS, and HV performance of IDCS-OMOICA during training and testing.

In Figure 8 (a), during the testing process, the MID index of IDCS-OMOICA tends to stabilize throughout the iteration, indicating that the overall solution of the algorithm is almost the same as the optimal solution. The

SNS and HV indicators tend to stabilize around 100 and 110 iterations, and eventually stabilize around 0.76 and 0.78. In Figure 8 (b), during practical application, IDCS-OMOICA exhibits a state that is basically consistent with the training process. The initial solutions to algorithms are usually randomly generated, and the quality of these solutions varies greatly, resulting in significant fluctuations in the objective function values during the initial iteration. MID stability is indicative of HR allocation that approaches optimal levels, with resources allocated in a suitable manner and employees aligned with organizational goals. This, in turn, fosters enhanced employee satisfaction and improved organizational performance. SNS is stable and high, which means that the configuration scheme is evenly distributed, covering multiple scenarios, enhancing organizational flexibility and innovation, and adapting to market changes. HV is stable and high, representing a wide range of configuration options that can consider multiple objectives such as cost and efficiency, and enhance the overall competitiveness of the organization for optimization. Therefore, the algorithm has good convergence effect, and the solution obtained under the set optimal algorithm parameters has high practicality, approaching the ideal state.

3.2 Comparative analysis of IDCS-OMOICA model performance

To further analyze the HR configuration performance of the IDCS-OMOICA model, this study compares the Constrained Evolutionary Algorithm with Dual Populations (CAEAD) and the Mixed-Integer Stochastic Optimization-Non-smooth Optimization by Mesh Adaptive Direct Search (MISO-NOMAD) algorithms. The CAEAD algorithm can effectively handle complex constraints in multi-objective optimization problems and is suitable for scenarios involving multiple constraints (such as cost, skill matching, project requirements, etc.) in HR allocation. The MISO-NOMAD algorithm is specifically designed to handle mixed integer optimization problems and is suitable for scenarios involving discrete decision variables (such as employee allocation) and continuous decision variables (such as cost and efficiency) in HR allocation. The comparison of MID, SNS, and HV indicators of each algorithm during the experimental process at iteration 400 is shown in Figure 9.

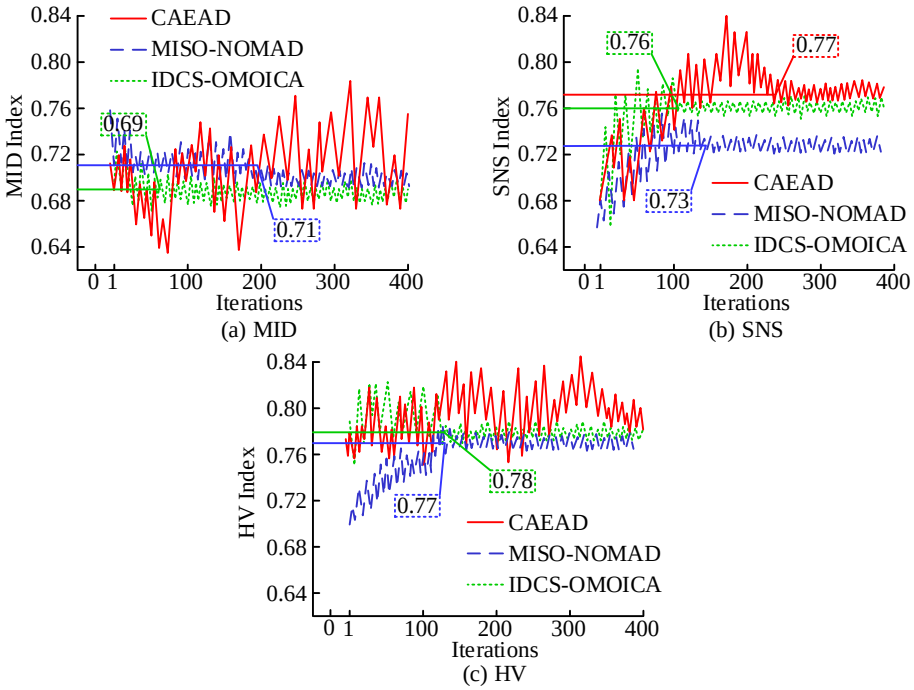


Figure 9: Test results of MID, SNS and HV indexes of each algorithm during 400 iterations.

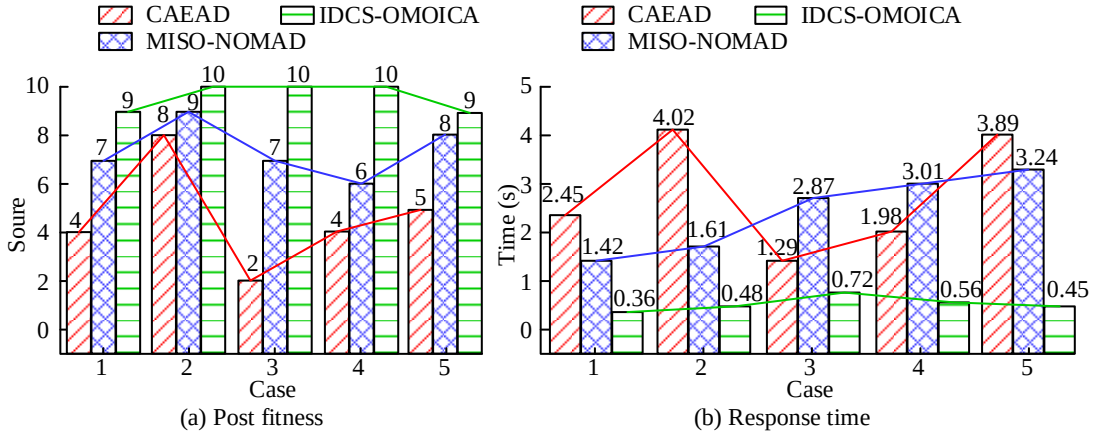
In Figure 9 (a), the MID value of CAEAD is relatively high, but it is unstable throughout the entire iteration process. The MID value of MISO-NOMAD stabilizes around 0.71 when the iteration reaches around 190. IDCS-OMOICA remains relatively stable throughout the entire iteration, with an average MID value of 0.69, showing the best performance. In Figure 9 (b), the SNS values of CAEAD, MISO-NOMAD, and IDCS-OMOICA remain stable around 0.77, 0.73, and 0.76 at iterations 230, 130, and 100. Although the SNS value of CAEAD algorithm is relatively high after stabilization, it is not much different from the IDCS-OMOICA model, and the convergence speed is too slow. In Figure 9 (c), the HV value of CAEAD fluctuates significantly during the iteration process. The HV values of MISO-NOMAD and IDCS-OMOICA remain stable around 0.77 and 0.78 at iterations of 130 and 110. Overall, influenced by the Lévy flight strategy, the

IDCS-OMOICA model can converge faster and reach a stable state among all algorithms, and all indicators are superior to the compared algorithms. Compared with the MISO-NOMAD algorithm, the MID value of IDCS-OMOICA decreases by 0.02, the SNS value increases by 0.03, and the HV value increases by 0.01. Table 3 shows the actual performance of each model in the optimization of HR allocation in five project teams of a certain company.

Table 3: The performance of each model in real cases.

Case	IDCS-OMOICA			MISO-NOMAD			CAEAD		
/	MID	SNS	HV	MID	SNS	HV	MID	SNS	HV
1	0.66	0.77	0.78	0.69	0.71	0.74	0.35	0.61	0.60
2	0.64	0.76	0.77	0.71	0.69	0.75	0.87	0.62	0.58
3	0.65	0.78	0.75	0.71	0.71	0.76	0.54	0.69	0.55
4	0.65	0.79	0.76	0.69	0.68	0.74	0.36	0.61	0.57
5	0.63	0.77	0.78	0.73	0.70	0.76	0.74	0.71	0.59

To conduct a statistical analysis of the differences in the data in Table 3 and calculate the *P*-value, the study uses the One-Way ANOVA method to compare whether there are significant differences in the means of three or more groups of data. The study uses the Python's scipy stats library to calculate the F-value and *P*-value of analysis of variance. The final results showed that the *P*-values of MID, SNS, and HV are 0.0013, 0.0456, and 0.0012, respectively, all less than 0.05, indicating significant differences in the mean values of the three models in various indicators. In Table 3, IDCS-OMOICA demonstrates good performance in all cases. Its MID value is generally better than the comparison algorithm, which means that the solution obtained by this model is closer to the ideal solution. A lower MID value represents that the model can achieve more ideal arrangements in HR allocation, ensuring that employees are closely aligned with organizational goals. In terms of SNS, IDCS-OMOICA also performs well in most cases, demonstrating the diversity of its solutions. A higher SNS value indicates that the model can adjust flexible solutions in dynamic environments to meet diverse scene requirements. In terms of HV indicators, the performance of the model is particularly outstanding in cases 1 and 5, indicating that its solution set occupies a large volume in multidimensional space, that is, the model has high precision and accuracy. A higher HV value highlights the efficiency of the model in balancing multiple objectives, such as cost and efficiency, thereby enhancing the overall competitiveness of the organization. Therefore, the IDCS-OMOICA model has demonstrated good optimization capabilities and universality in practical cases. Figure 10 shows the personnel fit and response time of HR allocation schemes obtained by various algorithms on these instances.



(a) Post fitness (b) Response time

Figure 10: Comparison results of fitness and algorithm response time of various algorithms for HR allocation in practical cases.

In Figure 10 (a), the adaptability is evaluated on a scale of 1-10, with higher scores indicating that employees who have undergone HRCO are more suitable for their respective positions. The source of fitness score is a comprehensive evaluation of employees' performance in their positions, including multiple dimensions such as skill matching, job performance, task completion quality, team collaboration ability, and the fit between personal career development and the position. The data are collected through questionnaire surveys, performance

evaluations, and feedback from colleagues and superiors. The data analysis models are used for quantitative scoring to obtain a fitness score ranging from 1 to 10. Influenced by the BEE, the IDCS-OMOICA model scores above 9 points on all instances, making it the highest scoring algorithm among all algorithms. In Figure 10 (b), the response time solved by CAEAD is very unstable. This is because CAAAD algorithm requires frequent adjustment of the solution structure when dealing with complex constraints in multi-objective optimization problems, which increases the computational complexity and uncertainty of response time. The average response time solved by the IDCS-OMOICA model is 0.51s, which is 1.92s faster than the MISO-NOMAD algorithm (2.43s). In summary, the IDCS-OMOICA model can achieve good results in optimizing HR configuration and help enterprises better apply talent. This model has a fast response time, can quickly obtain results, and has strong practicality. A short response time indicates the ability to quickly solve problems, reduce computational resource consumption, and improve overall efficiency. Faster response time can significantly improve user experience, reduce waiting time, and increase user satisfaction. High personnel adaptability indicates good solution performance. The experimental results indicate that the IDCS-OMOICA model can efficiently allocate HRs.

4 Discussion

In the era of knowledge economy, enterprises are facing fierce market competition and constantly changing business demands. Traditional HR management methods are no longer sufficient to meet the needs of modern enterprises. Enterprises need new methods for management to ensure that the allocation of HRs can maximize work efficiency and employee satisfaction. Based on this, this study proposes an IDCS-OMOICA model, aiming to effectively manage and allocate HRs, enabling them to quickly adapt to market changes and meet corporate strategic goals. The experimental results showed that the IDCS-OMOICA model outperformed the comparative CAEAD and MISO-NOMAD in performance indicators such as MID, SNS, and HV. Specifically, the average MID value of the IDCS-OMOICA model was 0.69, the SNS value stabilized around 0.76 at 100 iterations, and the HV value stabilized around 0.78 at 110 iterations. The stability and superiority of these indicators suggested that the IDCS-OMOICA model would rapidly converge and attain a stable state, producing diverse and high-quality solution sets. This was of particular importance for multi-objective optimization problems. Compared with the research of Seifi H et al. [7], the IDCS-OMOICA model not only performs well on large datasets but also has advantages in multi-objective optimization. The GWO algorithm based on Sugeno fuzzy inference model proposed by Seifi H et al. may face slow convergence speed and local optima when dealing with complex multi-objective problems. The IDCS-OMOICA model effectively avoids these problems through Lévy flight and random walk mechanisms, improving the algorithm's global search capability and resolution quality. As the size of the dataset increases, the model can adapt to more complex scenarios by adjusting parameters and optimizing algorithms, ensuring effective allocation of HRs in multi-project environments. Small and medium-sized enterprises often face challenges of limited resources and talent loss. Through efficient allocation of HRs, they can maximize production efficiency, reduce operating costs, and improve employee satisfaction, thereby enhancing their market competitiveness. In addition, technology companies and the healthcare industry can also utilize this model to optimize HR allocation in response to rapidly changing market demands and complex project management.

5 Conclusion

To solve the HRCO problem, this study proposed a fusion algorithm, IDCS-OMOICA, and established a corresponding model. The IDCS-OMOICA model effectively explored and adapted to dynamic environments by simulating the parasitic behavior of cuckoos and competition between empires. In the experiment, the SNS index of the IDCS-OMOICA model remained stable around 0.76 at iteration 100, and the HV index remained stable around 0.78 at iteration 110. This model outperformed the comparison algorithms in performance indicators such as MID, SNS, and HV. Compared with the MISO-NOMAD, its MID value decreased by 0.02, SNS value increased by 0.03, and HV value increased by 0.01. The average response time solved by this model was 0.51s, which was 1.92s faster than MISO-NOMAD. In summary, the IDCS-OMOICA model has good optimization ability and universality in practical applications. The research model has fast convergence and high solution quality. However, due to the involvement of RL and imperialist competition mechanisms, it may consume more computing resources during operation, especially when dealing with large-scale problems. This may lead to low efficiency in resource-constrained environments, affecting the practicality and scalability of the model. The current dataset storage solution may not be able to seamlessly expand storage capacity when facing continuous growth in data volume. In the future, further research will be conducted on how to optimize the algorithm structure and computation process while ensuring stable output results, reducing the consumption of computing resources, and improving the performance of the model on large-scale problems. Future research will adopt more efficient distributed storage technology to solve the problem of storage capacity expansion. More diversity protection mechanisms will also be introduced, such as dynamic adjustment of crowding distance, to ensure a more uniform distribution of solutions on the Pareto front. In

the face of HR allocation in hospitals, it is necessary to simultaneously optimize the allocation of medical staff, equipment, and beds to improve the efficiency and quality of medical services. For small and medium-sized enterprises, optimization of employee recruitment, training, and promotion strategies should be considered to improve employee satisfaction and retention rates.

References

[1] Suwarno S, Fitria F, Azhar R (2023). Optimizing budget allocation: a strategic framework for aligning human resource investments with financial objectives and business goals. Atestasi: Jurnal Ilmiah Akuntansi, 6(2): 835-855. https://doi.org/10.57178/atestasi.v6i2.880

[2] Pota M, Minutolo A, Damiano E, De Pietro G, Esposito M (2022). Betting on yourself: a decision model for Human Resource Allocation enriched with self-assessment of soft skills and preferences. IEEE Access, 10(1): 26859-26875. https://doi.org/10.1109/ACCESS.2022.3157640

[3] Zhao L, Wang Y (2024). Research on optimizing human resource expenditure in the allocation of materials in universities. Information, 15(9): 522-534. https://doi.org/10.3390/info15090522

[4] Wanjiku C, Anyieni B (2022). Effects of resource allocation on performance of Nyandarua County Assembly. International Journal of Public Administration and Management, 7(1): 45-58. https://doi.org/10.29322/ijsrp.12.03.2022.p12356

[5] Yinusa A, Faezipour M (2023). Optimizing healthcare delivery: a model for staffing, patient assignment, and resource allocation. Applied System Innovation, 6(5): 78-103. https://doi.org/10.3390/asi6050078

[6] Chowdhury S, Dey P, Joel-Edgar S, Bhattacharya S, Rodriguez-Espindola O, Abadie A, Truong L (2023). Unlocking the value of artificial intelligence in human resource management through AI capability framework. Human Resource Management Review, 33(1): 100899-100912. https://doi.org/10.1016/j.hrmr.2022.100899

[7] Seifi H, Shams N, Mohammad Cyrus K (2022). A novel self-regulating and intelligence meta-heuristic-fuzzy approach for integrated and optimal human resource allocation in normal and critical conditions. International Journal of Fuzzy Systems, 24(1): 121-134. https://doi.org/10.1007/s40815-021-01123-9

[8] Dabirian S, Abbaspour S, Khanzadi M, Ahmadi M (2022). Dynamic modelling of human resource allocation in construction projects. International Journal of Construction Management, 22(2): 182-191. https://doi.org/10.1080/15623599.2019.1616411

[9] Rodgers W, Murray J M, Stefanidis A, Degbey W Y, Tarba S Y (2023). An artificial intelligence algorithmic approach to ethical decision-making in human resource management processes. Human Resource Management Review, 33(1): https://doi.org/100925-100944. 10.1016/j.hrmr.2022.100925

[10] Waldkirch M, Bucher E, Schou P K, Grünwald E (2021). Controlled by the algorithm, coached by the crowd–how HRM activities take shape on digital work platforms in the gig economy. The International Journal of Human Resource Management, 32(12): 2643-2682. https://doi.org/10.1080/09585192.2021.1914129

[11] Tarafdar M, Page X, Marabelli M (2023). Algorithms as co-workers: Human algorithm role interactions in algorithmic work. Information Systems Journal, 33(2): 232-267. https://doi.org/10.1111/isj.12389

[12] Park H, Ahn D, Hosanagar K, Lee J (2022). Designing fair AI in human resource management: Understanding tensions surrounding algorithmic evaluation and envisioning stakeholder-centered solutions. Proceedings of the 2022 CHI Conference on Human Factors in computing Systems, 51(1): 1-22. https://doi.org/10.1145/3491102.3517672

[13] Nikzad E, Bashiri M, Abbasi B (2021). A matheuristic algorithm for stochastic home health care planning. European Journal of Operational Research, 288(3): 753-774. https://doi.org/10.1016/j.ejor.2020.06.040

[14] Bankins S, Formosa P, Griep Y, Richards D (2022). AI decision making with dignity? Contrasting workers' justice perceptions of human and AI decision making in a human resource management context. Information Systems Frontiers, 24(3): 857-875. https://doi.org/10.1007/s10796-021-10223-8

[15] Mohiz M J, Baloch N K, Hussain F, Saleem S, Zikria Y B, Yu H (2021). Application mapping using cuckoo search optimization with Lévy flight for NoC-based system. IEEE Access, 9(1): 141778-141789. https://doi.org/10.1109/ACCESS.2021.3120079

[16] Al-Abaji M A (2021). Cuckoo search algorithm: review and its application. Tikrit Journal of Pure Science, 26(2): 137-144. http://dx.doi.org/10.25130/tjps.26.2021.039

[17] Abed-alguni B H, Alawad N A, Barhoush M, Hammad R (2021). Exploratory cuckoo search for solving single-objective optimization problems. Soft Computing, 25(15): 10167-10180. https://doi.org/10.1007/s00500-021-05939-3

[18] Hasanvand M, Nooshyar M, Moharamkhani E, Selyari A (2023). Machine learning methodology for identifying vehicles using image processing. Artificial Intelligence and Applications, 1(3): 170-178. https://doi.org/10.47852/bonviewAIA3202833

[19] Puertas-Martín S, Redondo J L, Ferrández M R,

Pérez-Sánchez H, Ortigosa P M (2022). MultiPharm-DT: A multi-objective decision tool for ligand-based virtual screening problems. Informatica, 33(1): 55-80. https://doi.org/10.15388/21-infor469

[20] Lin X, Yang Z, Zhang X, Zhang Q (2022). Pareto set learning for expensive multi-objective optimization. Advances in Neural Information Processing Systems, 35(1): 19231-19247. https://proceedings.neurips.cc/paper_files/paper/2022/file/7a583691ccfcf8945ab714b677ccbf0b-Paper-Conference.pdf

[21] Li Q, Zeng X, Wei W (2023). Multi-objective particle swarm optimization algorithm using Cauchy mutation and improved crowding distance. International Journal of Intelligent Computing and Cybernetics, 16(2): 250-276. https://doi.org/10.1108/ijicc-04-2022-0118

[22] Dilekh T, Benharzallah S, Mokeddem A, Kerdoudi S (2024). Dynamic context-aware recommender system for home automation through synergistic unsupervised and supervised learning algorithms. Acta Informatica Pragensia, 13(1): 38-61. https://doi.org/10.18267/j.aip.228

Data Mining Approach for River Flood Hazard Time-Series: Using a Combination of Triple Exponential Smoothing and Neural Networks, in Demak

Dimara Kusuma Hakim[1*], Rahmat Gernowo[2], Anang Widhi Nirwansyah [3], Totok Haryanto[4]
[1]Department of Informatics, Universitas Muhammadiyah Purwokerto, 53182, Indonesia
[2]Department of Physic Faculty of Mathematics and Natural Sciences, Diponegoro University, Indonesia
[3]Department of Geography Education, Universitas Muhammadiyah Purwokerto, 53182, Indonesia
[4]Bachelor of Management Program, Faculty of Economics and Business, Universitas Muhammadiyah Purwokerto, 53182, Indonesia
E-mail: dimarakusumahakim@gmail.com
[*]Corresponding author

Keywords: triple exponential smoothing, neural network, time-series, flood forecast, flood hazard

Received: October 9, 2024

Conventional flood hazard maps depict a static perspective on flood risk and are helpful in flood forecasting. Then, time-series flood forecasting research was developed and widely conducted. Still, it has limitations because the univariate method used does not consider the influence of other variables, unlike regression. Also, the determination of flood hazard weights, which is usually conducted through empirical studies using AHP, has a good level of accuracy but has the drawback of expert subjectivity.
A new approach is proposed for flood hazard forecasting, combining the Triple ES method with NN and determining weights in a data-driven manner using NN. A combination of time series and regression methods (significantly non-linear), namely Triple ES with NN, results in good accuracy with an MSE value of 3.03, MAE 1.20, RMSE 1.74, R^2 0.54, and a MAPE of 32.93%.
The evaluation results for flood hazard weight determination with an MSE of 0.0111 and a MAPE of 7.81% show promising results, and the weights can be used in Hazard Flood GIS. Visualization in the form of a GIS Hazard Map can be done after all related raster data have been combined. The computational outcomes, particularly the MSE and MAPE values, demonstrate the effectiveness of the proposed approach, providing a clear understanding of the model's performance.

Povzetek: Z razvojem modela Triple ES in nevronskih mrež učinkovito napoveduje poplavno nevarnost z vizualizacijo v obliki GIS.

1 Introduction

There is no doubt that river flooding is one of the most destructive natural disasters [1], [2], [3], as it encompasses impacts that include social, economic, and environmental dimensions. The development of flood control efforts and disaster reduction has successfully reduced some casualties due to flooding; however, this remains a global concern [4]. The flood in Demak that occurred some time ago has had a very significant impact on the local community. The flood in Demak that occurred some time ago has significantly impacted the local community. The flood caused by the breach of six levees on the Wulan River and heavy rainfall has resulted in 89 villages in 11 sub-districts submerged with water levels ranging from 30 to 80 cm, affecting approximately 93,149 [5]. This incident is not new for Demak, as it has frequently happened in previous years [6], [7], [8], [9], [10], [11].
The National Disaster Management Agency (BNPB) and the Meteorology, Climatology, and Geophysics Agency (BMKG) have implemented mitigation strategies to reduce the impact of flood disasters [12], [13], [14], and they also conduct flood risk projections based on weather data analysis and local geographical conditions. However, the predictions have a time limit for various reasons (weather and climate variability, environmental changes, and climate change). The BNPB and BMKG do not provide long-term flood predictions that extend beyond the current year [15], [16], [17], which presents research opportunities outside of these government agencies.
Conventional flood hazard maps present a static view of flood risks. Still, they are less effective in capturing the dynamic characteristics of the variables that influence these risks over time, which are essential [3], [18], [19], especially in terms of long-term forecasting. Previous research has been conducted to perform long-term forecasting using ARMA and ARIMA methods [20], [21], and their weaknesses due to not considering seasonality components [22], [23], [24] have been addressed with SARIMA and Triple Exponential Smoothing [25]. However, the univariate nature of these methods still does not resolve the issue of the influence of other variables, so a combination of techniques is needed to simultaneously address several issues: trends, seasonality, and the impact between variables.
In flood hazard mapping, an index must be multiplied by the variables influencing flooding. The determination of the flood hazard index, which has been carried out by BNPB in the form of the Indonesian Disaster Risk Index

(IRBI) using the Analytic Hierarchy Process (AHP) method, provides a guideline for the index used by decision-makers. However, it has several limitations [26], [27]: the necessity of having experts, difficulties in using comparative scales, potential low consistency, sensitivity to data changes, limitations in handling changes in priorities, and a limited number of variables. Therefore, research to create a separate index driven by data is an interesting endeavor to pursue [28].

This research aims to develop a flood hazard information system using a data mining approach with time series forecasting through a combination of the Triple Exponential Smoothing (Triple ES) method and Artificial Neural Networks (ANN). It is expected to address issues such as trends, seasonality, and the influence between variables simultaneously and determine the optimal weight of flood hazard variables using a data-driven approach with ANN. Optimization with gradient-based algorithms is used in every method to achieve the lowest possible error value.

2 Data

The data is in tabular, raster, and vector (shapefile) form. Tabular data contain relevant attributes or variables, such as rainfall, temperature, humidity, etc. Raster data (Figure 1) consists of a grid of pixels representing spatial information: land cover (COPERNICUS 100m Proba-V-C3 Global) & elevation (DEMNAS). The shapefiles used represent regional boundaries and river lines.

This study uses three data types for deeper analysis and understanding of the phenomenon studied: flood database (Table 1), meteorological data (Table 2), and raster data. Flood event information is collected, and we create a Flood database that contains information related to flood events and variables at the location (depth, runoff, river distance, land cover, wind speed, rainfall, etc.). This database looks at flood risk, finds places likely to flood, and plans how to protect those places. Including temperature, precipitation, humidity, surface pressure, and wind speed, meteorological data helps us understand how the atmosphere and weather trends affect flooding and rain. A thorough understanding of how meteorological conditions interact with flood episodes and how they affect ecosystems and populations is obtained by combining these two forms of data.

Elevation in raster form (satellite imagery) and river area (in shapefile/SHP format) of Demak Regency was obtained from the Geospatial Information Agency of Indonesia. River distance is obtained by performing Euclidean calculations for each pixel in the river area data and finding the nearest distance. The raster land cover data was obtained by accessing the API (Application Programming Interface) to the Copernicus Global Land Cover Layers (a product of Sentinel-2), which has a relatively high resolution (100 meters).

Runoff is obtained by converting land cover according to the Runoff Coefficient Table [29]. All this data is stored and converted into raster format and resampled to approximately 30 meters.

Then, the following data was obtained from ERA5-Land Hourly over 14 years (2011-2024) from the upstream area of the research location: precipitation, temperature, dew point, humidity, surface pressure, and wind speed. This data is stored as a database.

Additionally, data on flood events sourced from mass media and the National Disaster Management Agency is collected as material for determining the flood hazard weight. All data is processed to assess flood hazard weight and forecast time series.

Table 1: Flood event

id	date	location	coord	depth
1	2023-01-01	Betokan	-6.87, 110.63	55
2	2023-01-01	Bintoro	-6.89, 110.63	83
3	2023-01-01	Mangun jiwan	-6.88, 110.59	48
…	…	…	…	…

Table 2: Meteorological data

YEAR	MO	DY	T2M	T2MDEW	QV2M	PS	WS10M	PRECTOTCORR
2011	1	1	25.6	23.81	18.75	99.03	4.88	63.42
2011	1	2	25.2	23.09	17.92	99.14	4.4	37.1
2011	1	3	25.4	23.42	18.29	99.13	3.85	26.04
2011	1	4	25.3	23.4	18.27	99.08	4.73	11.12
2011	1	5	25.8	23.94	18.88	99.04	4.3	10.82
2011	1	6	25.9	23.82	18.73	99.13	3.42	3.62
2011	1	7	25.6	23.6	18.49	99.11	3.59	1.39
…	.	.	…	…	…	…	…	…

3 Method

3.1 Related works

Based on various previous studies on the topic of river floods (Fig 1), two things are often discussed: risk maps [30], [31] and prediction models [32], [33], [34]. This study attempts to combine the two domains with a data mining approach.

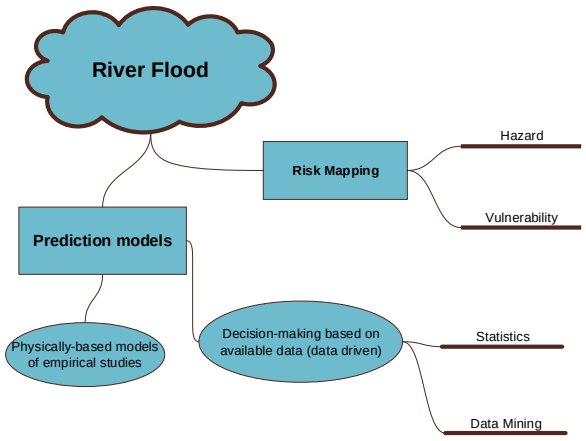


Figure 1: State of the art mind map

This summary table (Table 3) organize the findings from various previous studies and existing limitations.

Table 3: Related works

Method	Key findings	Limitations
Projected flood risk based on analysis of weather data and local geographical conditions [7], [8], [9]	Implementation of mitigation strategies to reduce the impact of flood disasters	Does not provide long-term flood predictions beyond the current year
ARMA & ARIMA [15], [16],	Long-term forecasting using ARMA and ARIMA methods	Doesn't consider seasonal components
SARIMA & Triple ES [17], [18], [19]	Overcoming the disadvantages of ARMA and ARIMA by considering seasonal components	Univariate properties do not resolve the influence of other variables
Analytic Hierarchy Process (AHP) [20], [21]	Determination of Indonesia's disaster risk index (IRBI)	Requires expertise, difficulty in using comparative scales, low potential for consistency, sensitivity to data changes, limitations in handling priority changes, and limited number of variables

This study develops a flood hazard information system using a data mining approach with time series forecasting through Triple Exponential Smoothing (Triple ES) and Artificial Neural Networks (ANN) methods. This approach is expected to overcome the problems of trends, seasonality, and influence between variables simultaneously and determine the optimal weight of flood hazard variables using a data-based approach with ANN. Each method is optimized with a gradient-based algorithm to achieve the lowest possible error values.

3.2 Determination of river flood hazard weight with ANN

Determining the weight of flood hazards, which is generally carried out using an empirical approach by using the Analytic Hierarchy Process method (AHP), is a very relevant research area in dealing with disaster threats by examining the relationship between hydrological variables in the past (such as rainfall, river flow, and topography), has provided insight into flood patterns [35], [36]. However, this has limitations because it heavily relies on the subjective assessments of experts; there is a significant possibility of personal bias that can influence the final results [37], [38]. Another data-driven approach is considered to be able to overcome this problem [39], [40]. Data-driven approaches, especially data mining with various methods, can utilize large amounts of data and identify complex patterns to determine the weight of flood hazards [41], [42]. In this task, the Artificial Neural Network (ANN) method is used, a computational model inspired by how the brain works [43]. Technically, it utilizes several processing units called neurons, processed through several hidden layers, requires an activation function to operate, and employs a specific optimization technique to achieve optimal predictions. Several steps were taken in this research to perform weight calculations using ANN.

3.2.1 Create a flood and precipitation database, download raster data

Historical data on flood events in the Demak region over the last 12 years has been collected, including the location and timing of the flood occurrences. Data on precipitation history was created for the upstream area (Grobogan Rain Station) over 14 years to complement the flood database and to understand the rainfall patterns that influence flood occurrences in the downstream area (Demak).

Raster data covering river elevation and area were downloaded from the Indonesian Geospatial Information Agency and land cover data from Sentinel (using the Copernicus API). The Surface Runoff band is created based on the land cover by converting the Runoff Coefficient Table [23]. In contrast, the river distance band is calculated using Euclidean distance (Eq 1) for each pixel relative to the nearest river area [44].

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

Flood data is updated based on raster: elevation, river distance, runoff, and land cover at the time and location of the flood event. The average value is calculated by reading the raster (GeoTIFF), then setting a radius of 500 meters from the flood location point, creating 50 random points, taking each value from those random points, and calculating the average value. The depth of flooding is one of the leading indicators in flood risk assessment.

3.2.2 Conducting modeling with ANN for calculating weights for each variable

Conducting modeling using Artificial Neural Network (ANN) to predict flood risk based on input variables (Fig 2) involves training the model, validation, and optimization to achieve accurate results. Calculating the weights for each variable involves the following steps: calculating the average weight, normalizing the weights, and determining the percentage weights. These weights indicate the extent of each variable's influence on flood risk and are used in creating the flood hazard index.

To perform the calculations, the flood event database was updated with average elevation, average river distance, average surface runoff, and average rainfall.

Then, outlier values in the data were removed using Z-score, and the data was divided into training and testing sets. Data is standardized using StandardScaler to ensure all features are at the same scale. The artificial neural network model was built using Sequential, with multiple layers of Dense and Dropout to prevent overfitting. The model is compiled using an ADAM optimizer with an adjusted learning rate and an MSE loss function. To improve the model's performance, EarlyStopping stops training early if there is no increase in value loss after a few epochs. The model is trained with training data and validated with a subset of training data. After training, the model is evaluated with test data to obtain optimal MSE values. The predictions are made on a test set, the prediction results are classified into the nearest bins, and finally, the MAPE is calculated to measure the accuracy of the predictions. The prediction results and actual values are used to generate a comparison graph.

Several layers are used to determine flood weights using ANN. First layer: 64 neurons with ReLU activation function. After this stage, 20% of the neurons will be dropped using the Dropout layer to prevent overfitting and improve the model's generalization. The second layer contains 32 neurons with a ReLU activation function. After this stage, 20% of the neurons will be dropped using the Dropout layer to prevent overfitting and improve the model's generalization—the third layer is one neuron with a linear activation function. The Adam optimizer (Adaptive Moment Estimation) is used with a learning rate of 0.001 because Adam combines the advantages of two optimizers, AdaGrad and RMSProp, to adjust the learning rate during training, so it requires less memory and is efficient. The MSE loss function was chosen because it is suitable for regression problems. The EarlyStopping callback automatically stops the training if the validation loss does not improve after five consecutive epochs and the best weight achieved during the training is selected. Model training using training data is carried out for a maximum of 100 epochs, with a batch size of 10. 20% of the training data will be used as validation data.

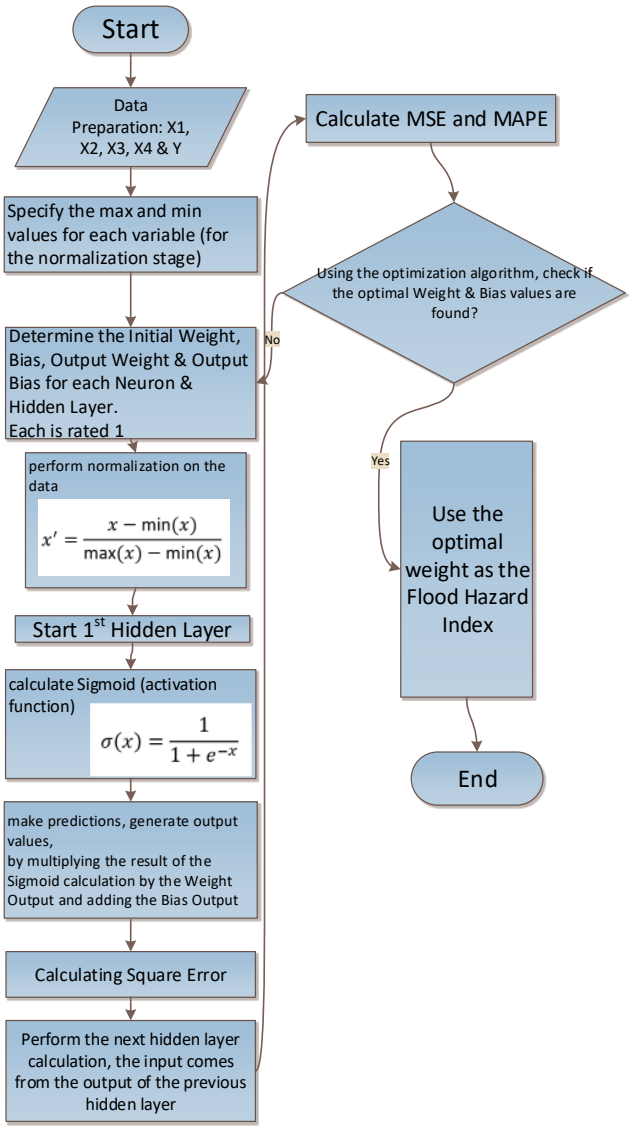


Figure 2: Determination of river flood hazard weight with ANN

3.3 Build a flood forecasting model with a Combination of Triple ES, ANN, and gradient-based optimization

The Triple ES method (Fig 3.a) is a technique for forecasting that considers level, trend, and season. This method is considered a univariate time series approach because it only predicts one variable. Meanwhile, the Neural Network method (Fig 3.b) predicts one dependent variable influenced by various independent variables but has no relationship with time.

The univariate time series approach, which has limitations in considering the influence between variables (because the forecast is only for one variable), has been improved with the multivariate approach (such as Vector Autoregression), which involves the influence between time series variables [45]. However, it still has limitations in handling situations where there are independent variables (X1, X2) that affect the dependent variable (Y) while also involving time [45].

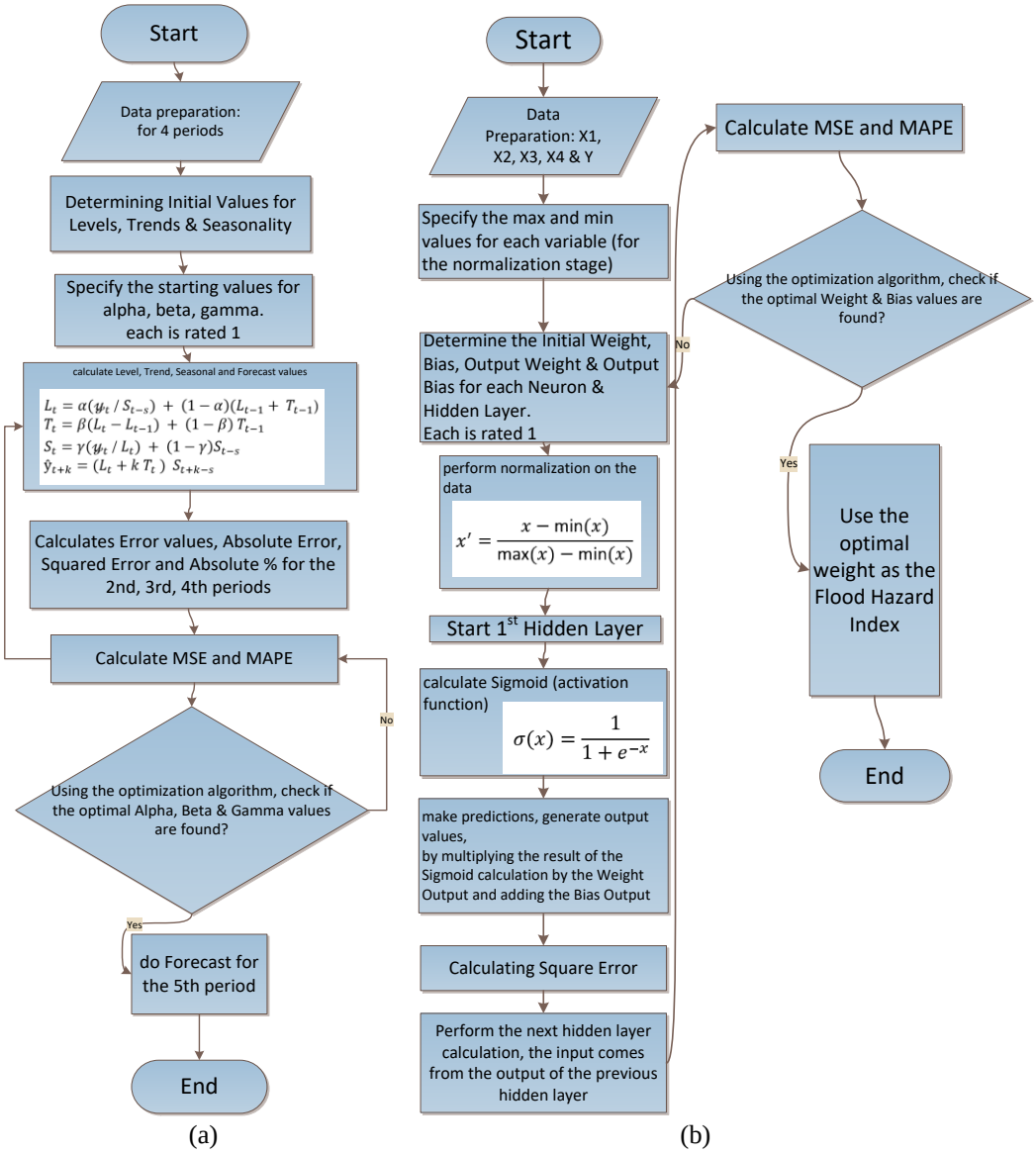


Figure 3: Basic methods: triple ES (a) and neural network (b)

The combination of the Triple ES method and ANN (Fig. 4) is carried out in this research to simultaneously address several issues: level, trend, seasonality, and the influence between variables. Forecasting each variable uses Triple ES first, and estimation is done using Neural Network.

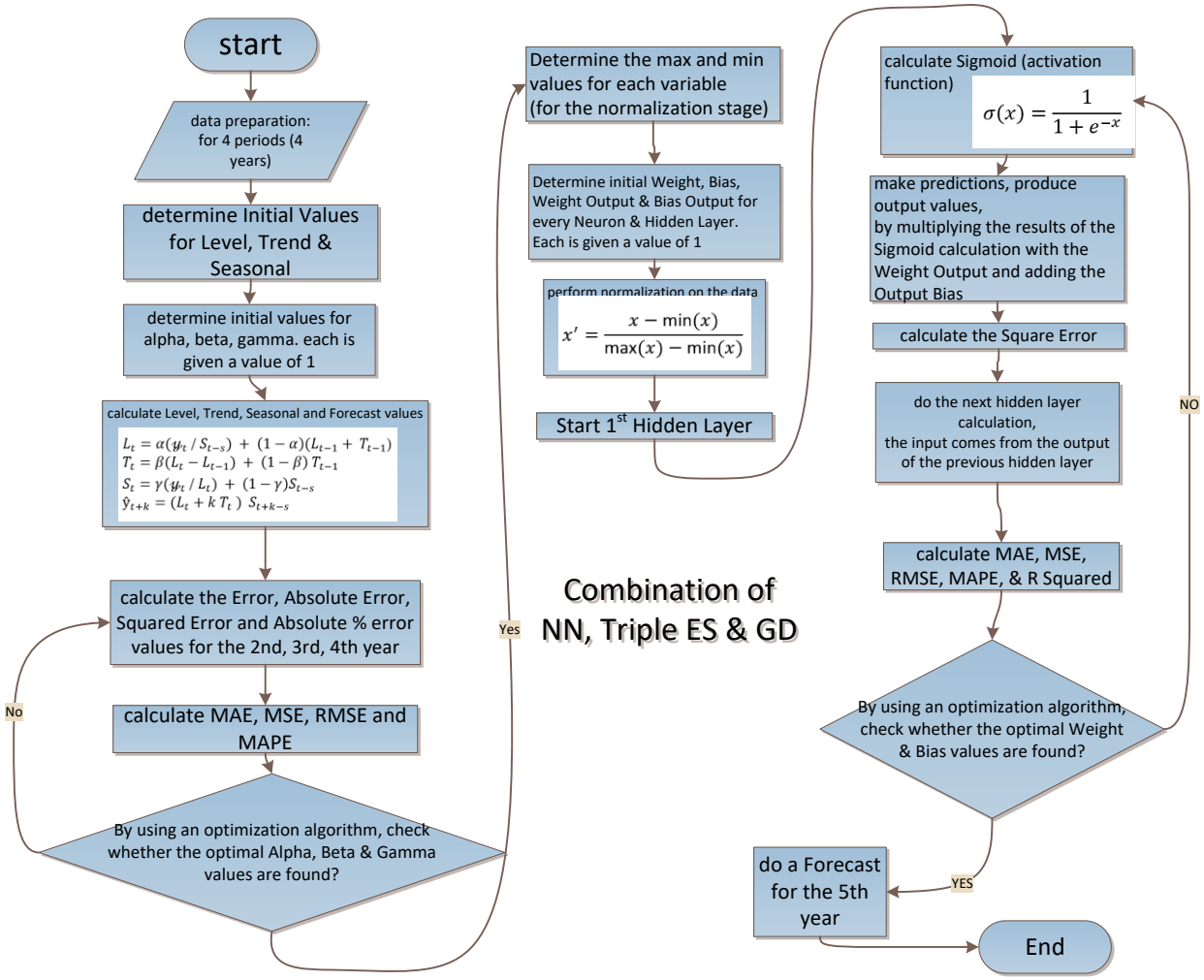


Figure 4: Forecast with hybrid approach

3.3.1 Preparation

Some data that affect rainfall are grouped by year and month: Temperature (C), Dewpoint (C), Humidity (g/kg), Surface Pressure (kPa), Wind Speed (m/s), and Total Precipitation (mm/day). The aggregation functions used in this data processing are max and sum. The sum function is used for total precipitation to obtain the total rainfall over a specific period, which is essential for predicting rainfall accumulation. Temperature, dew point, Humidity, surface pressure, and wind speed each use the max function, identifying extreme conditions that may affect rainfall. This aggregation function is beneficial in achieving accuracy in forecasting and prediction.

3.3.2 Prediction of each influential variable

Triple ES performs initial forecasting for each variable using gradient-based optimization: L-BFGS-B with loss: MSE. Then, evaluation is conducted by calculating MSE and MAPE for each variable and comparing the forecast results with the test data. At the end of this activity, model 1 for forecasting is saved.

Triple ES is a time series forecasting method that is an advanced form of exponential smoothing because it involves three parameters: α (alpha) as the level, β (beta) as the trend, and γ (gamma) as the seasonal component [46]. The steps taken in Triple ES are to calculate Level, Trend, and Seasonality and make forecasts (Eq 2-5). Where: L_t is the level at time (t), T_t is the trend at time (t), S_t is the seasonal component at time (t), Y_t is the actual value at time (t), (m) is the seasonal period, (k) is the forecast horizon.

$$L_t = \alpha \frac{Y_t}{S_{t-m}} + (1-\alpha)(L_{t-1} + T_{t-1}) \quad (2)$$

$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1} \quad (3)$$

$$S_t = \gamma \frac{Y_t}{L_t} + (1-\gamma)S_{t-m} \quad (4)$$

$$F_{t+k} = (L_t + kT_t)S_{t-m+k} \quad (5)$$

3.3.3 Creating and training a neural network model

$$z = \frac{x - \mu}{\sigma} \quad (6)$$

The standard scaler shifts the feature distribution so that it has a mean of zero and a standard deviation of one (Eq 6), ensuring that all features are on the same scale. Where: x

is the original value of the feature, μ is the mean of the feature, σ is the standard deviation of the feature, and z is the normalized value of the feature.

3.3.4 Conducting testing for combination method

It begins with forecasting the variables using Model 1 (forecasting) that has been created before, forecasting several months in the testing data as variable X. Then, it continues with the use of Model 2 (prediction) that has been developed to predict variable Y. The prediction results are then processed by rounding to the nearest class to round the predictions to the values closest to the score class. The prediction results are saved for evaluation. A join of actual data and predicted data was performed for evaluation, and then MSE, RMSE, R Squared and MAPE were calculated.Then, the optimal model produced is used to make forecasts for a period of 20 years (2025-2045).

3.4 Visualization and calculation of river flood hazards based on raster

Visualizing river flood hazards using rasters requires integrating geospatial data and raster processing (especially starting with DEM). Raster processing allows for data analysis in a grid format, which is very useful for illustrating variations in topography and land use that can influence water flow.

Several steps are taken to produce a Time Series Hazard Map for a particular month and year. This process begins with collecting relevant geospatial data, such as elevation maps, rainfall data, and land use information (to determine runoff values). Once the data is collected, the next step is preprocessing, which includes cleaning and normalizing the data to ensure consistency and accuracy. Next, a hydrological analysis is carried out to determine areas at high risk of flooding by considering various variables.

First, the prepared raster data will be analyzed using a hydrological model to simulate water flow and potential inundation. This model will produce a flood risk map showing the most vulnerable areas. Then, the analysis results will be visualized as a thematic map that is easy to understand.

3.4.1 DEM (digital elevation model)

To be able to retrieve DEM Raster data, DEMNAS Data Download is carried out from the Geospatial Information Agency (BIG) via tanahair.indonesia.go.id. Then, Raster Data Processing (Merging & Clipping) is carried out. Merging is carried out because the raster covering the Demak area consists of several GeoTIFF files, which are combined into one raster file. The Clipping process is carried out according to the boundaries of the study area (Shapefile, which is also obtained from BIG). Coordinate Transformation (resolution 30 m for each pixel) and Coordinate System equalization (EPSG: 4326) are needed because data integration is carried out, namely the combination of data from various sources; in this study, the integration of elevation raster with raster: rainfall, runoff, and river distance will be carried out. Coordinate Transformation and Coordinate System equalization are carried out by changing all rasters into the same coordinate system so that a more comprehensive analysis can be carried out. Normalization and inverting are carried out at the analysis stage. Because the higher the location, the lower the risk of flooding, so when using the Min-Max Normalization method as a score, the invert is carried out (Fig 5.a).

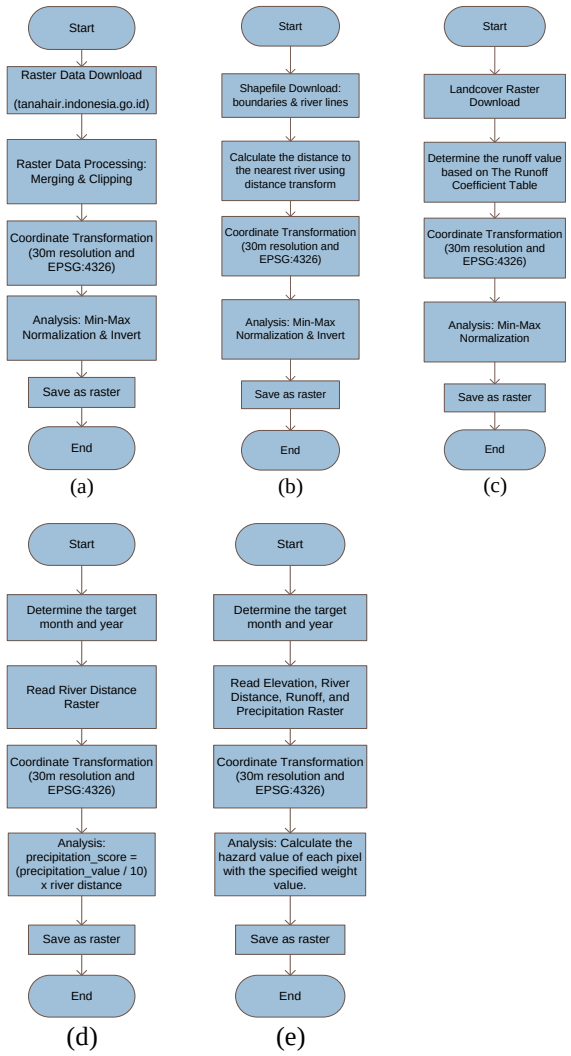


Figure 5: Raster Processing: a) DEM, b) River Distance, c) Runoff, d) Precipitation, e) Hazard

3.4.2 River distance raster creation

Two shapefiles are first downloaded from BIG (boundary and river line) to perform river distance raster processing (Fig 5.b), which is needed for the analysis. After that, the distance to the nearest river is calculated using the distance transformation method (euclidean distance) so that each pixel will contain the distance value to the nearest river line. Next, a coordinate transformation is performed to ensure that the raster has a resolution of 30 meters and uses the EPSG:4326 coordinate reference system, which is essential for the suitability of geospatial data. After the distance raster is generated, perform a Min-Max normalization analysis to transform the distance values into a more standardized range, followed by value inversion so that areas closer to the river have higher values. Finally, the results of this processing are saved as a raster for further analysis.

3.4.3 Runoff raster creation

Landcover Raster (Copernicus Global Land Cover Layers) taken within a 40 km radius of the research location is used in the Runoff Raster Creation (Fig 5.c), which aims to produce a map of potential water runoff. This raster has a resolution of 100 m [47], which is good enough to be used in flood analysis. It describes the land cover that affects the value of water runoff: forest, agricultural land, urban, etc. The runoff value is determined for each pixel based on the Runoff Coefficient Table [29], which provides different coefficients for different land cover types, reflecting how much rainwater will flow as runoff compared to what the soil absorbs. Furthermore, a coordinate transformation is carried out to ensure that the runoff raster has a resolution of 30 meters and uses the EPSG:4326 coordinate reference system, which is essential to maintain the consistency of geospatial data and facilitate integration with other data. After obtaining the runoff value, a Min-Max normalization analysis is carried out to transform the runoff value into a more standardized range, making it easier to interpret the data. The last step is to save the results of this processing as a raster.

3.4.4 Precipitation raster

The relationship between rainfall in upstream areas and the distance to the river of a location provides a score for the influence of rainfall at a location. Based on actual and forecast data on rainfall, the target month and year for analysis are first determined because rainfall can vary significantly over time. After that, the previously created river distance raster (the distance between the nearest rivers from each location) is read. To maintain the consistency of geospatial data, coordinate transformation is carried out to create a rainfall raster with a resolution of 30 meters using the EPSG:4326 coordinate reference system. The rainfall value is divided by 10 (to adjust the scale) and then multiplied by the distance to the river. The last step is to save the processing results in a raster (Fig 5.d).

3.4.5 Hazard raster

In Hazard Raster processing (Fig 5.e), the first step is determining the target month and year for the analysis because risk conditions can vary depending on a specific time. After that, read the previously generated elevation, river distance, runoff, and rainfall rasters, which provide essential information about the factors that affect the risk. The analysis is carried out by calculating the hazard value of each pixel by multiplying it by the weight value determined for each factor, which can include elevation, distance to river, runoff, and rainfall. The last step is to save the results of this processing as a raster.

4 Result and discussion

This section explains all the results that have been obtained, in the form of numbers, tables, and map visualizations.

4.1 Raster data processing

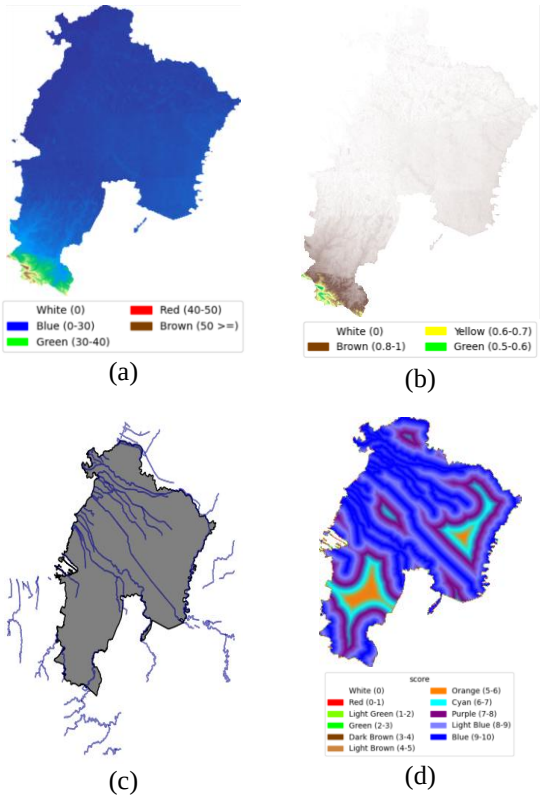
The raster calculation process is designed to produce several key parameters: elevation, distance to river, and runoff analysis. This process is continued by creating a raster to assess the impact of upstream rainfall on downstream areas based on total monthly rainfall and river distance.

The elevation raster describes the height of the land surface relative to sea level, starting with the raster merging process and continuing through resampling (Fig 6.a), truncation, and finally, normalization, which includes an inversion step (Fig 6.b). This is important because higher locations tend to have less impact on flooding.

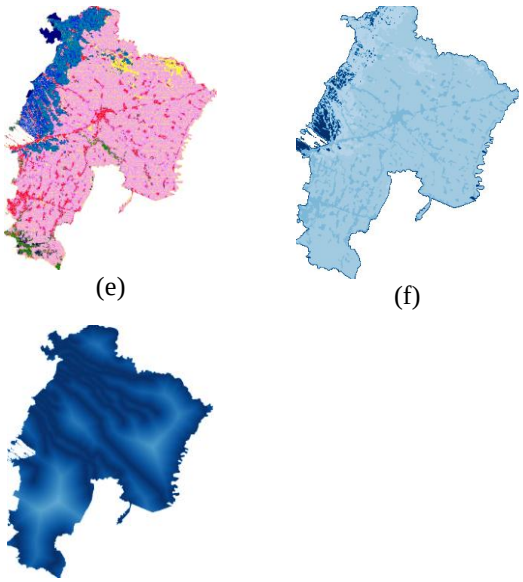
The distance to the river raster reflects the shortest distance from each pixel to the river. It starts with a base map depicting the river flow (Fig 6.c) and then calculates the shortest distance for each pixel using the Euclidean distance method (Fig 6.d). Pixels closer to the river are shown in bluer colors.

Precipitation, the impact of upstream rainfall on downstream areas, is calculated by involving the distance of the nearest river to each pixel (Fig 6.g).

The land cover raster is the raw data that provides information about the land cover at each pixel (Fig 6.e). The color code for the legend is according to the Discrete Classification in the Copernicus Global Land Service [48]. Then, the land cover value is converted into runoff (Fig 6.f) based on the runoff coefficient table [29]. In the raster visualization, a bluer color indicates more water, which is considered runoff.



(a) (b) (c) (d)



(e) (f) (g)

Figure 6: Raster Processing: a) Raw Elevation, b) Inverted Elevation, c) River Flow, d) River Distance, e) Land Cover, f) Runoff, g) Precipitation

4.2 River flood hazard weight

This study aims to assess river flood hazard weights using Artificial Neural Networks (ANN), offering advancements over traditional methods like the Analytic Hierarchy Process (AHP). By creating a flood and rainfall database and utilizing relevant raster data, ANN modeling calculates the weight of variables influencing flood hazards. This data-driven approach is expected to yield more accurate results and enhance the effectiveness of flood hazard information systems.

4.2.1 Flood database and raster data

A comprehensive database was constructed to develop a data-driven flood hazard weighting system. The first data includes flood events with the following structure: id, flood, region, year, date, information, and coordinates (Table 1). The second data comprises 11 years (2013-2023) of precipitation from the upstream region of Grobogan. Subsequently, raster data was collected. The obtained Raster Elevation data was integrated with river distance, land cover, and runoff information. The flood database was updated using raster data for the column's: runoff, river distance, precipitation, and elevation. Precipitation data was populated by querying Grobogan's upstream precipitation database.

Table 4: Flood database

id	f	date	loc	coord	prec p	dept h	ru no ff	distc	elv
1	1	2023-01-01	Betokan	-6.87, 110.63	149.54	55.68	0.38	0.56	4.48
2	1	2023-01-01	Bintoro	-6.89, 110.63	149.54	83.66	0.47	0.50	1.68
3	1	2023-01-01	Mangunjiwan …	-6.88, 110.59 …	149.54	48.40	0.33	0.50	5.21
15911	0	2013-04-10	Ngelo Kulon	-6.79, 110.68	21.76	0	0.6	0.46	3.60
15912	0	2013-04-10	Pecuk	-6.79, 110.71	21.76	0	0.51	0.54	4.84
15913	0	2013-04-10	Pasir	-6.79, 110.67	21.76	0	0.56	0.44	4.70

Table 4: Flood database (continued)

Temp	Dewp	Humd	Pressure	Wind Speed	Precipitation	datetime
24.71	23.28	18.13	99.16	2	0.32	2013110
24.7	23.18	18.01	99.12	2.14	0.28	2013111
24.62	23.09	17.94	99.11	2.25	0.23	2013112
…	…	…	…	…	…	…
27.01	25.33	20.45	99.49	1.48	0.59	2023123
26.78	25.26	20.39	99.46	1.61	0.64	2023122
26.55	25.16	20.26	99.4	1.71	0.7	2023123

4.2.2 Flood hazard model with ANN

The data is prepared before applying ANN. When a negative weight is found at the end of the process, the data preparation process is corrected by inverting the variables with negative weights. A negative weight indicates an inverse relationship between the input and target variables. Higher elevations and greater distances from the river are related to a reduced risk of flooding. This makes sense, as areas that are elevated or farther from the river are less prone to being impacted by floodwaters. The variables were defined as X1: elevation, X2: river distance, X3: runoff, X4: precipitation, and Y: flood depth (target). Outliers were removed using the Z-score method to ensure data quality. The dataset was split into training and testing sets with an 80:20 ratio. Next, the data was standardized to improve the model's performance.

The neural network model was constructed using a sequential architecture. The model included an input layer with 64 neurons and a ReLU activation function, followed by a dropout layer with a 20% dropout rate. This was followed by a hidden layer with 32 neurons and another dropout layer with a 20% dropout rate. The output layer consisted of a single neuron with a linear activation function. The model was compiled using the Adam optimizer with a learning rate 0.001 and MSE loss function. Early stopping was incorporated to prevent overfitting, monitor the validation loss with five epochs, and restore the best weights.

The model was then trained on the prepared dataset, and its performance was evaluated using MSE and MAPE.

The model achieved an optimal MSE of 0.0111, indicating a high level of accuracy in its predictions. Additionally, MAPE was 7.81%, reflecting the model's precision in forecasting relative to the actual values.

4.2.3 Weight

The weight calculation results show elevation: -0.0435, river distance: -0.0492, runoff: 0.0411, and precipitation: 0.0623. At this stage, negative weight results for elevation and river distance indicate that both variables negatively influence flood predictions. In the context of flood hazard weighting, this is considered invalid, so the data preparation is repeated. The data for both variables is inverted.

After inverting both variables and repeating the process, the output shows that elevation has the highest average weight of 0.2096, followed by river distance at 0.0899, precipitation at 0.0729, and runoff at 0.0124. After normalization, the elevation weight becomes 0.5499, indicating that elevation is the most significant factor in the flood hazard model. The distance from the river and rainfall also have a considerable influence, with weights of 0.2360 and 0.1913, respectively, while surface runoff has the least impact, with a weight of 0.0325. Thus, elevation is the most influential variable in this model, followed by the distance from the river and rainfall, while surface runoff has the most minor contribution.

4.3 Combination of triple ES and ANN

To forecast the precipitation, the first step is to group historical statistics on rainfall by year and month and then find the highest value for each variable. Then, we used Triple ES to make predictions about each variable. The data was split into two groups: training data (from 2011 to 2019) and testing data (2020–2024). Using L-BFGS-B optimization, weather factors like temperature, humidity, pressure, and rainfall are used for forecasting. Once the model is made, each variable's MSE and MAPE values are found by comparing the predicted results with the actual data. The model is then saved as Model 1.

Based on the test results, the optimal seasonal period is 48 months (4 years). Based on the test results, the optimal seasonal period is 48 months (4 years); the test results for all periods are reported in Table 2. Periods from 12 to 36 show increasing error values, but it turns out that the 48-season period shows the lowest error (MAPE: 32.93%). The 60 period failed because the training data was insufficient for training for two full seasons (120).

The initial forecasting results (Triple ES) show that temperature (2.15%), dewpoint (2.89%), humidity (4.0%), and surface pressure (0.12%) have very low error values (MAPE) and are suitable for use in the following prediction stage. Wind speed has an error that is relatively low (12.8%) and acceptable in time-series analysis. Precipitation has a very high error (66.51%), which is addressed in the following prediction stage.

To help improve prediction accuracy, separate variables were created: normalized and binned, with Min-Max normalization used for all variables except for precipitation, which used equal-width binning. The variables Temperature, Dewpoint, Humidity, Surface Pressure, and Wind Speed receive scores from normalization within the range of 0-1. The total precipitation variable also gets a score of 0-1 but as a result of binning into four categories: light rain (score: 0.25), moderate rain (0.5), heavy rain (0.75), and hefty rain (1). The data is divided into two parts: training data and testing data.

The next step with ANN is to achieve optimal results; several treatments are carried out to train the model. It starts with the use of a standard scaler for data normalization, the use of 2 hidden layers (the first hidden layer using 100 neurons, and the second hidden layer using 50 neurons), a maximum of 500 iterations, activation function: ReLU, and optimizer: ADAM. At the end of this activity, model 2 for prediction is saved. The model generated with ANN shows a MAPE evaluation result of 0.3585, which means an average difference of 32.93% between the model's predictions and the actual data. The mean squared error between the expected and actual values is represented by an MSE of 0.11, indicating a small prediction error, meaning the model's performance is quite good.

The final step is a combination of methods (Fig 7), using models 1 and 2 sequentially, to forecast variable X for several years using model 1 for testing purposes. Then, the prediction of the Y variable is based on the forecast results of the X variable. Final evaluation results of the

combination of methods in MSE and MAPE are obtained; MSE: 3.03 shows that the average square error is relatively low, and MAPE 0.3293 means that the forecast misses by 32.93% of the actual value). In the context of time-series forecasting, it isn't easy to obtain high accuracy compared to prediction techniques, so this value indicates a reasonable level of accuracy.

Table 5 presents a comparison between the Triple Exponential Smoothing (ES) method, which uses a seasonal period of 48 and achieves a Mean Absolute Percentage Error (MAPE) of 66.51%, and the Ensemble Method, which was tested with seasonal periods of 12, 24, 36, and 48. The optimal performance was observed with a seasonal period 47, resulting in a MAPE of 32.93%. It is important to note that a seasonal period 60 could not be applied as it failed to meet the requirement of at least two complete cycles.

Table 5: Evaluation

Metric	Triple ES (48)	Ensemble (12)	Ensemble (24)	Ensemble (36)	Ensemble (48)
MAPE	66.51%	35.85%	35.10%	36.73%	32.93%
MAE	79.44	1.28	1.23	1.30	1.20
MSE	12472.07	3.28	3.33	3.10	3.03
RMSE	111.68	1.81	1.83	1.76	1.74
R^2	0.29	0.51	0.50	0.53	0.54

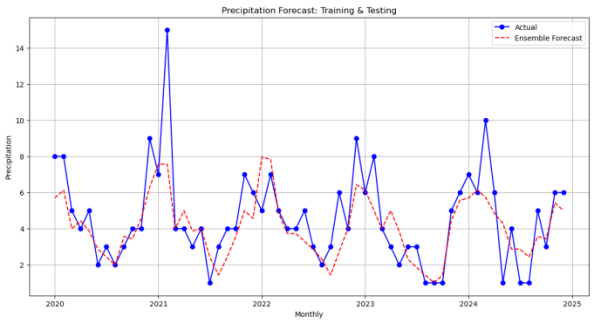


Figure 7: Precipitation forecast, training & testing

Once the model's accuracy is deemed sufficient, a forecast is made using this model for the next 20 years (2025-2045) in Fig 8. The results of this forecast are used for visualization purposes.

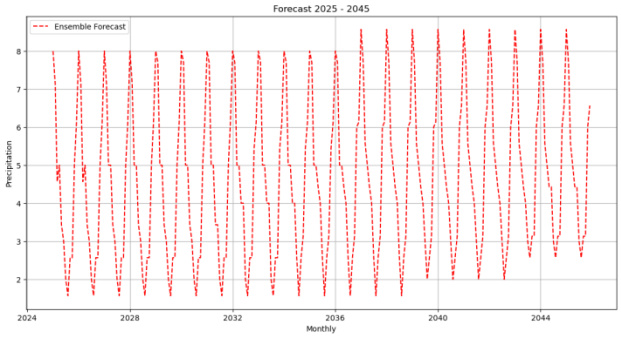


Fig 8: Precipitation forecast

4.4 Visualization in the form of a flood hazard map

Visualization in the form of a hazard map can be done using year and month parameters. Fig 2 shows the flood hazard calculation results with the variables elevation, river distance, runoff, and precipitation in a time-series manner. Because elevation, river distance, and runoff are relatively constant, rainfall provides the main difference in this time-series-based flood hazard research. When preparing all raster data, a transformation is performed to set the coordinate reference system (CRS) to EPSG:4326 with the desired resolution.

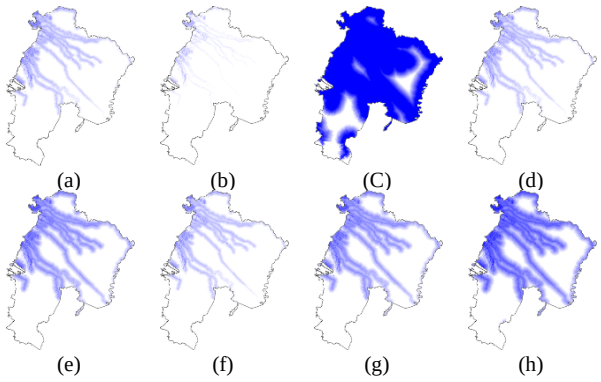
Elevation data was created by combining several rasters that form the Demak Regency and then clipped according to the regional boundaries. Because the elevation of the sea area is ignored, the empty values are changed to NaN, and the values are also set to 0 if they are negative (beneath the surface of the sea). The flood hazard weight calculation results indicate that the elevation variable's weight is negative. A negative weight for the elevation variable means that an increase in elevation tends to reduce the value calculated in the model. In the context of flood danger, the higher an area is, the less likely it is to be affected by floods. The maximum elevation value of 254.16 meters indicates that Demak has relatively low topography, making it more at risk of flooding, mainly since Demak is located in a coastal area with many rivers. The low elevation increases the likelihood of flooding during heavy rainfall or rising sea levels. Because the location of Demak Regency is generally close to the sea, scoring does not use height categories as usual for visualization purposes. Instead, it uses min-max normalization, which means the score ranges from 0 to 1 relative to the area. In performing this normalization, because negative elevation values (below sea level) are replaced with zero, the zero value is considered the minimum elevation value. The maximum elevation value is calculated; if NaN values are in the pixels, they will be ignored. Normalization is done by dividing each value by the maximum value that has been calculated. The normalized values are then inverted so that the highest value becomes the lowest and vice versa.

A distance transformation creates a raster for river distance, and the results are converted from degrees to meters. Like elevation, min-max normalization is also used to score for river distance. After normalization, the values are reversed so that the most significant value becomes the smallest and vice versa. Using inverse values in the score of the nearest distance to the river is related to the method of flood risk assessment. The closer an area is to the river, the higher the flood risk. By reversing the distance value, areas closer to the river will receive a higher score, indicating a more significant flood risk. Conversely, areas further from the river will receive a lower score, indicating a more negligible risk. The maximum distance to the river of 12,807.14 meters means that the farthest point in the analyzed area is approximately 12.8 km from the river.

Raster Land Cover was obtained from Copernicus Global Land Cover Layers: CGLS-LC100 Collection 3 [48], by

setting a coordinate point in Demak Regency and then defining a polygon with a radius of 50 km from the center point. Then, the Land Cover value is converted into Runoff according to the runoff coefficient table [23]. The runoff value is already in score form, so normalization, binning, or scoring is not needed again. A positive weight on this variable indicates that a higher runoff value means a higher risk of flooding. A higher runoff value means more surface water flows rather than being absorbed into the ground. This increases the potential flood risk because the ground surface cannot effectively absorb rainwater, causing water to flow and accumulate in certain areas. A high runoff value can be caused by various factors, such as excessive rainfall, impermeable soil, or land use changes like roads and buildings that increase the area of impermeable surfaces. Runoff forecasting uses double exponential smoothing to obtain the trend component and forecast results. However, the percentage change from the forecast results shows minimal annual variation, especially in the early years, with a slight upward trend in the following years; this may be due to the limited data. This increase in value is not significant enough to be considered a major factor in flood hazard assessment. Given the slight fluctuations and relatively stable trends, this runoff value is regarded as constant in the context of flood hazards, which is why this study only uses data from 2019, the most recent data available.

The use of Triple ES and NN causes the total rainfall forecast to produce a repeating pattern (Fig 9), the trend component remains but does not dominate, so there is no forecast of annual rainfall increase. The use of forecast data for creating flood hazard maps is done by providing the year and month as parameters, then the rainfall values are taken from the forecast data to form rainfall raster data. In determining the rainfall value for each pixel in Demak Regency as the downstream area, a multiplication is performed between the rainfall value of the upstream area (Grobogan) and the river distance that has been normalized and inverted. The closer to the river, the more the rainfall influence value is considered to be the same as the upstream area.



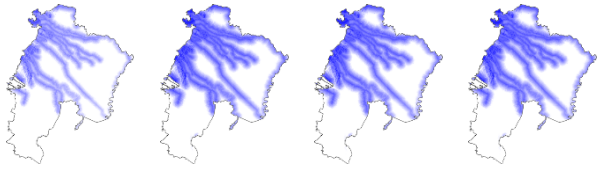


Figure 9: Flood hazard map, based on historical data and forecast results:
(a) 2012-02, (b) 2012-12, (c) 2021-02, (d) 2021-11, (e) 2026-01, (f) 2026-02, (g) 2030-01, (h) 2039-01, (i) 2042-02, (j) 2043-01, (k) 2044-01, (l) 2045-01

Flood hazards were historically high in 2021, particularly in February, when alarming risk figures were recorded, indicating the potential for significant environmental and community impacts. However, compared to the previous year, the flood probability has decreased significantly in 2022 and beyond, raising hopes that mitigation efforts are starting to bear fruit.

This study's projection analysis reveals a worrying upward trend in flood risk despite a decline in 2022. Flood risks are projected to increase from 2025 to 2025, with January and February of 2026 being the high-risk months. There is a possibility that the hazard will continue to grow as this trend is not limited to 2026 but will continue beyond (the projections of this study are limited to 2045). This suggests the need to consider future flood risks further and implement mitigation strategies to save communities and the environment from adverse impacts.

5 Conclusion

The conclusion drawn from this research is that the combination method of Triple ES and ANN is capable of predicting rainfall while considering influencing factors, with a MAPE of 32.93%.

Based on a 20-year forecast, flood risk can be estimated on a seasonal basis. Because of the use of seasonal components, the trend component is used only to support the seasonal component and does not represent the overall trend prediction (up and down).

For future research, it is recommended to use methods that only consider the trend component, such as Double Exponential Smoothing.

The flood danger in Demak Regency is not only from river factors but also from tidal flooding, so it is recommended to consider coastal flooding in future research.

Acknowledgements

This research is funded by the Ministry of Education, Culture, Research, and Technology, Directorate General of Higher Education, Research, and Technology of Indonesia. In accordance with the decision letter from the Directorate of Research, Technology, and Community Service Number 0459/E5/PG.02.00/2024.

References

[1] D. K. Hakim, R. Gernowo, and A. W. Nirwansyah, "Flood prediction with time series data mining: Systematic review," *Nat. Hazards Res.*, vol. 4, no. 2, pp. 194–220, Oct. 2023, doi: 10.1016/j.nhres.2023.10.001.

[2] I. Ardiansah *et al.*, "Integrated Streamflow Forecasting System: A Step Towards Smart Flood Management," *Inform.*, vol. 47, no. 9, pp. 109–121, 2023, doi: 10.31449/inf.v47i9.4890.

[3] S. Parvaze, J. N. Khan, R. Kumar, and S. P. Allaie, "Temporal flood forecasting for trans-boundary Jhelum River of Greater Himalayas," *Theor. Appl. Climatol.*, vol. 144, no. 1–2, pp. 493–506, Apr. 2021, doi: 10.1007/s00704-021-03562-8.

[4] Q. Yu, Y. Wang, and N. Li, "Extreme Flood Disasters: Comprehensive Impact and Assessment," *Water*, vol. 14, no. 8, p. 1211, Apr. 2022, doi: 10.3390/w14081211.

[5] B. Jo, "Peta Banjir Demak 2024, Jumlah Korban, & Update Kondisi Terkini," Tirto. Accessed: Mar. 30, 2024. [Online]. Available: https://tirto.id/peta-banjir-demak-dan-update-kondisi-terkini-gW6F

[6] DetikJateng, "Banjir Demak 2023 Rendam Seratusan Desa: Data Lokasi dan Penyebab," Detik. Accessed: Mar. 30, 2024. [Online]. Available: https://news.detik.com/berita/d-6497484/banjir-demak-2023-rendam-seratusan-desa-data-lokasi-dan-penyebab

[7] S. Afifah and D. R. Hizbaron, "Vulnerability assessment of residential buildings to tidal flood hazards in Sriwulan Village, Sayung District, Demak Regency," *E3S Web Conf.*, vol. 200, p. 01008, Oct. 2020, doi: 10.1051/e3sconf/202020001008.

[8] S. Dewi Novita, A. Alif Nor, T. Taryono, M. Muchamad Farid, and K. Dinda Nur Fadila, "Detection Of Flood Hazard Potential Zones By Using Analytical Hierarchy Process In Tuntang Watershed Area, Indonesia," *Geogr. Tech.*, vol. 19, no. 1/2024, pp. 61–78, Dec. 2023, doi: 10.21163/GT_2024.191.05.

[9] A. A. Atmaja, M. Asa, K. A. Alam, D. D. Putra, C. N. Yuanita, and C. Patricia, "Strengthening Resilience in the Face of Extremes: Lessons Learned from the 2024 Demak Flood," RDI Gobal.

[10] Mirza Hanif Al Falah, Tri Retnaningsih Soeprobowati, H. Hadiyanto, A. Rahim, Barra Madini Noor, and N. Permatasari, "Diatom Stratigraphy as a Flood Record in the Lower Tuntang River, Demak, Central Java," *Evergreen*, vol. 10, no. 1, pp. 272–282, Mar. 2023, doi: 10.5109/6781082.

[11] A. Purwanto, P. Paiman, and A. Sudiro, "The Use of Sentinel-2A Images to Estimate Potential Flood Risk With A Multi-Index Approach in The Mempawah Watershed," *Geosfera Indones.*, vol. 8, no. 1, p. 83, Apr. 2023, doi: 10.19184/geosi.v8i1.37156.

[12] K. Prasetyaningtyas, "Prakiraan Daerah Potensi Banjir Bulan Juli-September 2024." Accessed: Jun. 10, 2024. [Online]. Available: https://www.bmkg.go.id/iklim/potensi-banjir.bmkg?p=prakiraan-daerah-potensi-banjir-bulan-juli-september-2024&tag=&lang=ID

[13] Kompas, "BMKG Prediksi Banjir Bandang di Sumbar sampai 22 Mei, Imbau Warga Hindari Lereng Bukit." Accessed: May 14, 2024. [Online]. Available: https://nasional.kompas.com/read/2024/05/14/11021731/bmkg-prediksi-banjir-bandang-di-sumbar-sampai-22-mei-imbau-warga-hindari

[14] AntaraNews, "BMKG prediksi 12 daerah berpotensi alami banjir bandang kategori siaga." Accessed: May 15, 2024. [Online]. Available: https://www.antaranews.com/berita/2504189/bmkg-prediksi-12-daerah-berpotensi-alami-banjir-bandang-kategori-siaga#:~:text=Miming menyebutkan untuk periode tiga hari ke depan%2C,Barat%2C Kalimantan Selatan%2C Kalimantan Timur%2C dan Kalimantan Tengah.

[15] Tempo, "Aceh Flooding Subsides, BNPB Warns of Subsequent Downpour." Accessed: May 15, 2024. [Online]. Available: https://en.tempo.co/read/1867865/aceh-flooding-subsides-bnpb-warns-of-subsequent-downpour

[16] H. Cloke, "Why flooding is still so difficult to predict and prepare for," theconversation.com. Accessed: Jun. 15, 2024. [Online]. Available: https://theconversation.com/why-flooding-is-still-so-difficult-to-predict-and-prepare-for-126866

[17] Tempo, "Great Losses over 4 Years of Hydrometeorological Disasters: BNPB," Tempo. Accessed: Jun. 16, 2024. [Online]. Available: https://en.tempo.co/read/1689461/great-losses-over-4-years-of-hydrometeorological-disasters-bnpb

[18] A. Al-Fugara, A. N. Mabdeh, S. Alayyash, and A. Khasawneh, "Hydrological and Hydrodynamic Modeling for Flash Flood and Embankment Dam Break Scenario: Hazard Mapping of Extreme Storm Events," *Sustainability*, vol. 15, no. 3, p. 1758, Jan. 2023, doi: 10.3390/su15031758.

[19] M. Shariati, M. Kazemi, R. Naderi Samani, A. Kaviani Rad, M. Kazemi Garajeh, and N. Kariminejad, "An integrated geospatial and statistical approach for flood hazard assessment," *Environ. Earth Sci.*, vol. 82, no. 16, p. 384, Aug. 2023, doi: 10.1007/s12665-023-11077-w.

[20] N. Umar and A. Gray, "A water level prediction using ARMA and ARIMA models: A case study of the river Niger," in *GRASPA 2021: The International Environmetric Society*, 2021. [Online]. Available: https://pureportal.strath.ac.uk/en/publications/a-water-level-prediction-using-arma-and-arima-models-a-case-study

[21] W. M. Wong, M. Y. Lee, A. S. Azman, L. A. F. Rose, and F. P. T. dan Teknousahawan, "Development of Short-term Flood Forecast Using ARIMA," *Int. J. Math. Model. Methods Appl. Sci.*, vol. 15, pp. 68–75, 2021.

[22] E.-I. Koutsovili, O. Tzoraki, N. Theodossiou, and G. E. Tsekouras, "Early Flood Monitoring and Forecasting System Using a Hybrid Machine Learning-Based Approach," *ISPRS Int. J. Geo-Information*, vol. 12, no. 11, p. 464, Nov. 2023, doi: 10.3390/ijgi12110464.

[23] M.-J. Chang, I.-H. Huang, C.-T. Hsu, S.-J. Wu, J.-S. Lai, and G.-F. Lin, "Long-Term Flooding Maps Forecasting System Using Series Machine Learning and Numerical Weather Prediction System," *Water*, vol. 14, no. 20, p. 3346, Oct. 2022, doi: 10.3390/w14203346.

[24] A. Borucka, "Seasonal Methods of Demand Forecasting in the Supply Chain as Support for the Company's Sustainable Growth," *Sustainability*, vol. 15, no. 9, p. 7399, Apr. 2023, doi: 10.3390/su15097399.

[25] D. K. Hakim, R. Gernowo, and A. W. Nirwansyah, "Rainfall, Wind Speed, and Temperature Forecast

Using Triple Exponential Smoothing and Gradient Descent," *KnE Soc. Sci.*, Mar. 2024, doi: 10.18502/kss.v9i6.15311.

[26] N. Munier and E. Hontoria, "Shortcomings of the AHP Method," 2021, pp. 41–90. doi: 10.1007/978-3-030-60392-2_5.

[27] P. Li, "Design of AHP-Delphi Emergency Capability Evaluation Index System Model in Management Stage," 2022, pp. 1208–1214. doi: 10.1007/978-981-16-8052-6_170.

[28] J. Lin, C. Sreng, E. Oare, and F. A. Batarseh, "NeuralFlood: an AI-driven flood susceptibility index," *Front. Water*, vol. 5, Oct. 2023, doi: 10.3389/frwa.2023.1291305.

[29] CivilWeb, "Rational Method Runoff Coefficient." Accessed: Aug. 09, 2024. [Online]. Available: https://civilweb-spreadsheets.com/drainage-design-spreadsheets/runoff-and-rainfall-intensity-calculator-spreadsheet/rational-method-runoff-coefficient/

[30] R. B. Mudashiru, N. Sabtu, I. Abustan, and W. Balogun, "Flood hazard mapping methods: A review," *J. Hydrol.*, vol. 603, p. 126846, Dec. 2021, doi: 10.1016/j.jhydrol.2021.126846.

[31] N. I. Saikh and P. Mondal, "GIS-based machine learning algorithm for flood susceptibility analysis in the Pagla river basin, Eastern India," *Nat. Hazards Res.*, vol. 3, no. 3, pp. 420–436, Sep. 2023, doi: 10.1016/j.nhres.2023.05.004.

[32] A. B. Adeyemi and A. A. Komolafe, "Flood Hazard Zones Prediction Using Machine-Learning-Based Geospatial Approach in Lower Niger River Basin, Nigeria," *Nat. Hazards Res.*, Jan. 2025, doi: 10.1016/j.nhres.2025.01.002.

[33] M. Nhangumbe, A. Nascetti, S. Georganos, and Y. Ban, "Supervised and unsupervised machine learning approaches using Sentinel data for flood mapping and damage assessment in Mozambique," *Remote Sens. Appl. Soc. Environ.*, vol. 32, 2023, doi: 10.1016/j.rsase.2023.101015.

[34] Z. Demissie, P. Rimal, W. M. Seyoum, A. Dutta, and G. Rimmington, "Flood susceptibility mapping: Integrating machine learning and GIS for enhanced risk assessment," *Appl. Comput. Geosci.*, vol. 23, p. 100183, Sep. 2024, doi: 10.1016/j.acags.2024.100183.

[35] H. El-Bagoury and A. Gad, "Integrated Hydrological Modeling for Watershed Analysis, Flood Prediction, and Mitigation Using Meteorological and Morphometric Data, SCS-CN, HEC-HMS/RAS, and QGIS," *Water*, vol. 16, no. 2, p. 356, Jan. 2024, doi: 10.3390/w16020356.

[36] H.-J. Wang, R. Merz, S. Yang, and S. Basso, "Inferring heavy tails of flood distributions through hydrograph recession analysis," *Hydrol. Earth Syst. Sci.*, vol. 27, no. 24, pp. 4369–4384, Dec. 2023, doi: 10.5194/hess-27-4369-2023.

[37] E. Lechowska, "Approaches in research on flood risk perception and their importance in flood risk management: a review," *Nat. Hazards*, vol. 111, no. 3, pp. 2343–2378, Apr. 2022, doi: 10.1007/s11069-021-05140-7.

[38] A. Singh, D. Dawson, M. Trigg, and N. Wright, "A review of modelling methodologies for flood source area (FSA) identification," *Nat. Hazards*, vol. 107, no. 2, pp. 1047–1068, Jun. 2021, doi: 10.1007/s11069-021-04672-2.

[39] S. Sandiwarno, "Penerapan Machine Learning Untuk Prediksi Bencana Banjir," *J. Sist. Inf. Bisnis*, vol. 14, no. 1, pp. 62–76, Jan. 2024, doi: 10.21456/vol14iss1pp62-76.

[40] F. Karim, M. A. Armin, D. Ahmedt-Aristizabal, L. Tychsen-Smith, and L. Petersson, "A Review of Hydrodynamic and Machine Learning Approaches for Flood Inundation Modeling," *Water*, vol. 15, no. 3, p. 566, Feb. 2023, doi: 10.3390/w15030566.

[41] B. K. Panigrahi, S. Das, T. K. Nath, and M. R. Senapati, "An Application of Data Mining Techniques for Flood Forecasting: Application in Rivers Daya and Bhargavi, India," *J. Inst. Eng. Ser. B*, vol. 99, no. 4, pp. 331–342, Aug. 2018, doi: 10.1007/s40031-018-0333-9.

[42] M. Nayak, S. Das, and M. R. M. R. R. Senapati, "Improving Flood Prediction with Deep Learning Methods," *J. Inst. Eng. Ser. B*, vol. 103, no. 4, pp. 1189–1205, Aug. 2022, doi: 10.1007/s40031-022-00720-y.

[43] I. N. da Silva, D. Hernane Spatti, R. Andrade Flauzino, L. H. B. Liboni, and S. F. dos Reis Alves, *Artificial Neural Networks*. Cham: Springer International Publishing, 2017. doi: 10.1007/978-3-319-43162-8.

[44] J. Thioulouse, S. Dray, A.-B. Dufour, A. Siberchicot, T. Jombart, and S. Pavoine, *Multivariate Analysis of Ecological Data with ade4*. New York, NY: Springer New York, 2018. doi: 10.1007/978-1-4939-8850-1.

[45] M. Akkaya, "Vector Autoregressive Model and Analysis," in *Handbook of Research on Emerging Theories, Models, and Applications of Financial Econometrics*, Cham: Springer International Publishing, 2021, pp. 197–214. doi: 10.1007/978-3-030-54108-8_8.

[46] F. Kronthaler, *Statistics Applied With Excel*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2023. doi: 10.1007/978-3-662-64319-8.

[47] M. Buchhorn, M. Lesiv, N.-E. Tsendbazar, M. Herold, L. Bertels, and B. Smets, "Copernicus Global Land Cover Layers—Collection 2," *Remote Sens.*, vol. 12, no. 6, p. 1044, Mar. 2020, doi: 10.3390/rs12061044.

[48] M. Buchhorn *et al.*, "Copernicus Global Land Service: Land Cover 100m: collection 3: epoch 2015: Globe," Sep. 2020, *Zenodo*. doi: 10.5281/zenodo.3939038.

Enhanced Neural Differential Distinguisher for Speck32/64 Using Attention Mechanisms and Multi Ciphertext Inputs

Xue Jiang[1,2*], Min Li[1], Makulov Kaiyrbek[3], Valerii Lakhno[2], Sahun Andrii[2]
Email: xjiang1283@163.com, limin@bbc.edu.cn, kaiyrbek.makulov@yu.edu.kz, lva964@nubip.edu.ua,
[1]School of Computer and Information Engineering, Bengbu University, Bengbu 233030, China
[2]Department of Computer systems, networks and Cybersecurity, National University of Life and Environmental Sciences of Ukraine, Kyiv 03041, Ukraine
[3]Department of Computer Science, Caspian University of Technology and Engineering, Almaty 050151, Kazakhstan
E-mail: a.sagun@nubip.edu.ua
[*]Corresponding author

Keywords: EfficientNet, person-job matching measure, deep learning, artificial intelligence, convolutional neural network

Received: December 24, 2024

In recent years, neural network-based differential distinguishers have demonstrated significant advantages in accuracy and effi-ciency over traditional differential distinguishers in symmetric cipher differential analysis. However, when dealing with ciphers involving a higher number of encryption rounds, neural network differential distinguishers still struggle to accurately identify ci-phertext pairs. To address this issue, this study proposes a neural network differential distinguisher model based on attention mechanisms and optimized ciphertext input structures. Specifically, the model first innovates the residual structure within the attention mechanism to maximize the weight of highly discriminative features, enhancing the feature extraction capability of the improved model. Secondly, a multi-scale convolution method is employed, integrating the network structure ideas of RegNet, with the addition of convolutional branches and optimization of activation functions, which further enhances the model's feature ex-traction capability. Finally, a multi-ciphertext input pattern is introduced to improve the input data information, and random key encryption is applied to the input ciphertext structure to construct multi-feature information representations of the ciphertext and encryption functions. The results from 5-8 rounds of experiments on Speck 32/64 indicate that the proposed new neural distinguisher can significantly improve discrimination accuracy to a maximum of 1.65%. On this basis, we carried out an optimization study on the construction method of the multi-ciphertext-pair dataset. The new dataset can increase the accuracy of the distinguisher by 49.16% compared to that of the single-ciphertext-pair case, and can extend the number of attack rounds from 7 to 8.

Povzetek: Raziskava predlaga izboljšan nevronski diferencirnik za Speck32/64, ki s pozornostjo in večvhodnimi šifratnimi pari poveča kvaliteto klasifikacije in število razpoznavnih krogov.

1 Introduction

At the 2019 US Secret Conference, Gohr [1] introduced a novel cryptanalysis approach using a deep learning neural network. Targeting the block cipher Speck, the strategy constructed a differentiator using a neural network, which aimed to exploit machine learning pattern recognition capabilities to enhance cryptanalysis beyond traditional methods. This groundbreaking research ignited significant interest, inspiring numerous cryptographic researchers to delve deeper into the field. Building on Gohr's work, Baksi [2] modified the input data structure from a single ciphertext pair to multiple pairs sharing the same differential value. The application of this approach to large-block ciphers such as ASCON and KNOT yielded impressive results. Researchers have compared the performance of multilayer perceptron (MLP) [3], convolutional neural networks (CNN) [4], and long short-term memory (LSTM) [5] networks for differential analysis and concluded that MLP is the most effective, while the other architectures have limitations in accuracy and training speed.

Aayush Jain [6] further refined Baksi's research by maintaining the original input data structure while optimizing the MLP network. Additionally, following M. Wang's [7] findings, the optimal input difference for the PRESENT cipher was incorporated as a fixed input difference, which resulted in improved differential analysis accuracy for three to five rounds of the PRESENT cipher. Emanuele Bellini [8] integrated cryptographic algorithm characteristics into the neural network architecture to provide prior knowledge. The proposed neural network differentiator comprised two components: a time distinguisher and a feature extractor. The input ciphertext pair was divided into four equal segments, with each segment processed by two dense layers of 32 neurons. This approach aimed to independently extract features from each ciphertext block, minimizing inter-block interactions and aligning more closely with the cryptographic algorithm's structure. In the second part of

their research, the authors employed an MLP network for prediction. They introduced a novel neural network distinguisher to conduct a differential analysis on 4–7 rounds of TEA and RAIDEN encryption algorithms, surpassing the performance of three traditional differential distinguishers. Subsequent research has primarily focused on enhancing the neural network's acquisition of prior cryptographic knowledge. For instance, Lijun Lyu [9] integrated mixed-integer linear programming (MILP) to construct a neural network differential distinguisher. By pre-determining the required difference value (δ) using MILP, the authors processed reduced ciphertext and extracted ciphertext pairs with the difference δ as the neural network input, resulting in more regular input data. HengChuan Su [10] transitioned from independent differential pairs to polyhedral differential pairs, utilizing polyhedral differences to establish closer connections between input data, thereby enabling the neural network to extract more ciphertext features. In 2021, Adrien Benamira [11] analyzed Gohr's cryptanalysis research, and by shifting the input from differential pairs to differential values and dynamically adjusting these values during testing, three key conclusions were drawn: (1) a neural network distinguisher's performance is directly proportional to its acquired cryptographic information; (2) the performance of binary differential discriminators is influenced by the penultimate round's differential distribution; (3) while a neural network structure cannot be entirely replaced by other machine learning algorithms, incorporating local network modifications using machine learning techniques can enhance differential analysis accuracy. Benamira's study marked the first exploration of the working principles of neural network distinguishers. In 2022, Hayato Kimura [12] continuously cracked the low-pass cipher algorithm, iteratively refining network parameters based on experimental results to realize white-box attacks. Unlike previous research focusing on improving neural network structure and input data, Kimura's work emphasized interpretability by examining the neural network's decision-making process from a cryptographic perspective. Additionally, neural network differential distinguishers have achieved significant results in the differential analysis of various lightweight encryption algorithms [13-15].

However, there are noticeable issues in the existing research. For instance, the neural network structures used are still relatively simple, and most studies only perform basic parameter tuning on the networks. There is limited research on modifying the neural network structures by incorporating prior knowledge of cryptographic algorithms. The feature information provided by single or multiple fixed differential ciphertext pairs is limited, and the current input data structure fails to offer more feature information to the neural network, which restricts the accuracy of the network in distinguishing between ciphertext pairs.

To address these issues, this paper proposes a differential cryptanalysis model based on an attention mechanism and residual structure. The goal is to enhance the discriminative capabilities of neural network differential distinguishers, broaden the diversity of input data, and strengthen the model's learning and generalization abilities. We optimize the neural network structure by designing an attention-based residual module to enhance feature extraction capabilities and improve the model's discriminative power. We also expand the structure of input data by extending single ciphertext pairs to multiple concatenated ciphertext pairs, combined with encryption and decryption using random keys. This optimizes the input ciphertext structure, enhances the network's ability to learn ciphertext features, and improves the effectiveness of multi-round encryption algorithms. The results from 5-8 rounds of experiments on Speck32/64 indicate that the proposed distinguisher model achieves an improvement in accuracy of 0.16% for individual ciphertext pairs, and 0.85% for multiple ciphertext pairs. The number of rounds recognized can be extended to eight.

The specific contributions of this paper are as follows:

(1) Design of an Attention-based Residual Improvement Scheme: We propose a novel neural network distinguisher model that combines multi-layer convolutions and convolutional branches, exploring the impact of different network structures on the distinguisher's accuracy. This significantly enhances the network's feature extraction capabilities in complex encryption environments.

(2) We propose an innovative model for a multi-scale convolutional enhancement scheme that integrates RegNet's multi-scale convolution method and network design principles. By utilizing stacked 3x3 convolutions and adding convolutional branches, we investigate the impact of these modifications on the accuracy of the distinguisher. This approach significantly enhances the model's feature extraction capabilities.

(3) Development of a multi-ciphertext pair input structure-based encryption method: by utilizing random keys for one-round encryption, we achieve diversity in input data, strengthening the neural network's learning capabilities regarding ciphertext features. This significantly improves the model's generalization capabilities and training efficiency.

The structure of the paper is as follows: Section 2 introduces the fundamentals of the Speck32/64 cipher system, differential analysis, and the design and operation of differential distinguishers. Section 3 delves into the optimization of neural network distinguishers, including a comparative analysis of various neural network architectures, optimization of network parameters, and improvements in ciphertext input format. Section 4 presents the experiments and analysis. Finally, Section 5 provides the conclusion and directions for future work.

2 Preliminaries

2.1 Speck32/64

Speck [16] is a lightweight block cipher introduced by Ray Beaulieu et al. from the National Security Agency (NSA) in June 2013. This study focuses on Speck 32/64, which employs a 32-bit block size and a 64-bit key. The algorithm's core is a combination of modular addition,

shift, and bitwise XOR operations. The encryption process for Speck 32/64 is shown in Figure 1. The internal structure of this cipher comprises two 16-bit blocks: a left ciphertext block (Li) and a right ciphertext block (Ri). The subkey for the i-th round is denoted as Ki. The left ciphertext block is right-shifted by 7 bits and then modularly added to the right ciphertext block, followed by an XOR operation with the current round's subkey to produce a new left ciphertext block. The left-shifted right ciphertext block is XORed with this new left block to generate a new right ciphertext block. This process iterates for 22 rounds, resulting in the final ciphertext.

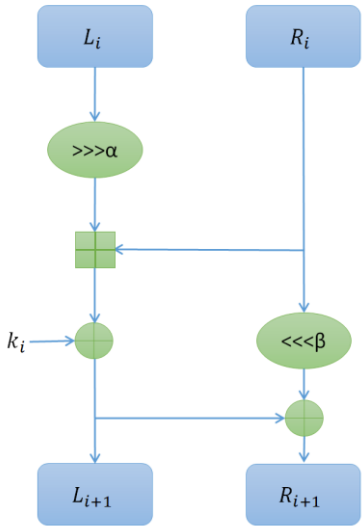


Figure 1: The round function and key schedule algorithm of Speck 32/64.

In one round of Speck cipher encryption, the input data are denoted as (L_i, R_i), and the output data are denoted as (L_{i+1}, R_{i+1}). The encryption process can be described as follows:

$$L_{i+1}=((L_i >>> \alpha) + R_i) \oplus k_i \quad (1)$$

$$R_{i+1} = (R_i <<< \beta) \oplus L_{i+1} \quad (2)$$

where $\oplus$ represents modular addition, $<<<, >>>$ represent bitwise cyclic shifts to the left and right in the version of Speck 32/64, α = 3, and β = 7.

2.2 How differential cryptanalysis and differential distinguishers work

Differential cryptanalysis is a potent technique for breaking block ciphers. The ability of a block cipher to resist this attack is a critical measure of its overall security. The core principle of differential cryptanalysis involves exploiting the non-uniform probability distribution of ciphertexts resulting from specific input differences [17]. By analyzing these patterns, an attacker can gather information about the key, significantly reducing the search space for potential keys.

Introduced by Biham [18] and Shamir in 1990, differential cryptanalysis is a plaintext attack that differentiates encrypted ciphertext from random data. The method hinges on identifying high-probability differential paths within the cipher's structure and constructing a differential distinguisher based on these paths. Due to its effectiveness against iterative cryptosystems, differential cryptanalysis is a standard tool for analyzing block ciphers. As a result, it is a fundamental metric for evaluating cipher security, and it has made significant contributions to the field of cryptanalysis and cryptographic security.

Differential cryptanalysis is a chosen-plaintext attack technique that exploits the statistical properties of block ciphers [19]. A differential distinguisher identifies high-probability differential patterns within the encryption algorithm, distinguishing valid ciphertext pairs from random ones. This information is used to filter potential key values. Specifically, a distinguisher for $\gamma - 1$ rounds can differentiate valid ciphertext pairs from random ones, enabling a key recovery attack on the full γ-round cipher.

A differential divider refines the distinguisher's output by identifying ciphertext pairs that align with a high-probability differential propagation path. This maximizes the number of "correct pairs" while minimizing the number of "wrong pairs", thereby reducing the key search space. The distinguisher's accuracy is crucial; higher accuracy leads to fewer false positives, a faster key search, and a more effective cryptanalysis.

Recent advancements in deep learning have led to the development of deep learning-based cipher distinguishers capable of achieving high accuracy. These distinguishers can be used to improve subsequent cryptanalytic efforts.

2.3 Differential analysis based on deep learning

Deep learning [20] is a machine learning technique predicated on artificial neural networks (ANNs) that process and analyze data by mimicking the interconnected structure and function of the human brain. It excels in handling complex tasks such as image and speech recognition, as well as natural language processing. Within this domain, convolutional neural networks (CNNs) constitute a critical branch specifically designed for image data processing. CNNs extract image features through convolutional, pooling, and fully connected layers. Convolutional layers employ convolutional kernels (filters) to scan input images and extract local features, while pooling layers reduce dimensionality and computational load, enhancing model generalization.

In 2015, He Kaiming et al. introduced the ResNet model [21], a convolutional neural network architecture incorporating skip connections. The residual module comprises two or more convolutional layers and a skip connection, directly summing the input and output to transmit the shallow network output to deeper layers, mitigating the vanishing gradient problem. Figure 2 illustrates the ResNet network structure. Despite its advancements, ResNet still presents opportunities for enhancing recognition accuracy, computational efficiency, and model parameter count. This study incorporates numerous residual structure design principles into the optimization of the neural network differential distinguisher's architecture.

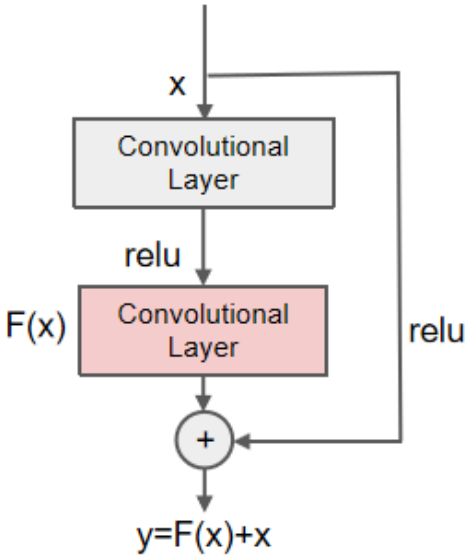


Figure 2: ResNet model structure

In 2019, Gohr pioneered the integration of deep learning with differential cryptanalysis. This study proposes a binary classification method for analyzing ciphertext pairs, building upon existing techniques for Speck encryption. Focusing on the input differential of (0x0040/0000), the goal is to distinguish between genuine and random ciphertext pairs. Real pairs originate from plaintext pairs with a specified difference, while random pairs stem from arbitrary plaintext pairs. When applied to Speck 32/64, this method surpasses traditional approaches in accuracy. Its applicability extends beyond Speck to other lightweight ciphers.

Gohr's neural network architecture, illustrated in Figure 3, comprises four modules: input, initial convolution, residual, and prediction modules. The input module feeds ciphertext data to the initial convolution module, structured as a ResNet network with 1 × 1 convolution kernels for feature extraction. The residual module enhances feature extraction using ten two-convolution neural networks and residual connections to minimize information loss. The prediction module employs multiple fully connected layers to map input features to output labels, classifying pairs as genuine or random and ultimately determining accuracy.

3 Methodology

As shown in Figure 3, the neural network-based differential distinguisher is composed of four modules: the input module (Module 1), initial convolution module (Module 2), residual module (Module 3), and prediction module (Module 4). This study focuses on optimizing and improving the network structure of the input module and the residual module to enhance the classification accuracy of the neural network-based differential distinguisher.

In this section, the structure and advantages of the network model proposed in this study are described in detail. The overall network structure is shown in Figure 4. In the process of improving the neural network differential distinguisher, the focus is primarily on optimizing the residual module and input data structure to enhance the model's distinguishing accuracy and performance. To improve the network architecture, attention-based [22] improvements were introduced to the residual module, enhancing feature extraction capabilities. This allows the neural network to better focus on important features, thus improving accuracy when distinguishing ciphertext pairs. Additionally, integrating RegNet's multi-scale convolution method into the neural network differential discriminator enhances feature extraction capabilities by incorporating convolution branches within the network architecture. By concatenating multiple ciphertext pairs and encrypting with a random key for one round, the input data are extended from single ciphertext pairs to multiple concatenated ciphertext pairs, and differential values are introduced with one round of random key encryption. This improvement not only increases the diversity of input data but also enhances the neural network's ability to learn ciphertext features, thereby improving the model's generalization ability and training efficiency. Based on multiple ciphertext pairs, further optimizations were made to accommodate higher rounds of the Speck encryption algorithm, improving the overall distinguishing accuracy. These improvements, achieved through network structure and input data optimization, significantly enhance the neural network differential distinguisher's performance in analyzing high-round cryptographic algorithms.

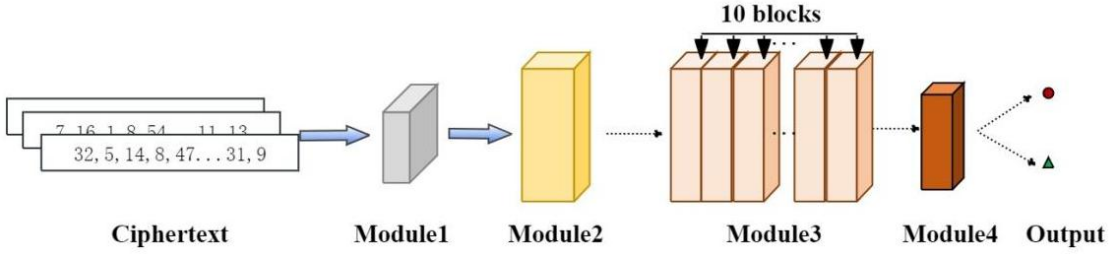


Figure 3: Gohr neural network

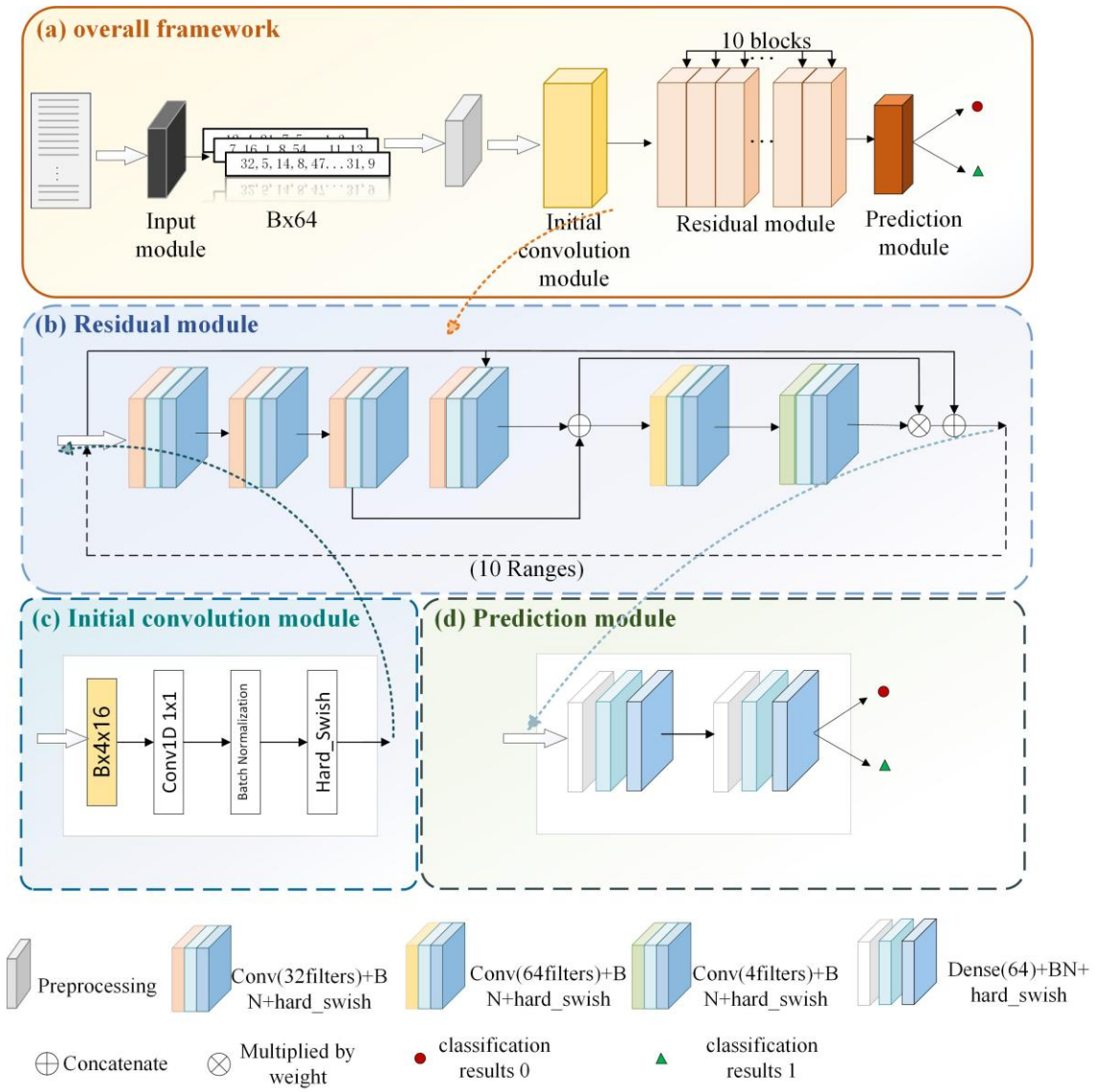


Figure 4: The overall structure of B-C3-HSwish.

3.1 Improvement of network architecture

Based on the original Gohr neural network differential distinguisher, we constructed a more accurate neural network differential distinguisher tailored for Speck 32/64 encryption reduced to 5–7 rounds. The specific improvements are detailed in Sections 3.1.1 and 3.1.2, where the differential classification accuracy for the 5–7 rounds of the Speck encryption algorithm is tested to validate the optimal improvement strategy. In Section 3.2, after determining the optimal structure for the distinguisher, the input ciphertext structure is further optimized to enhance the performance of the distinguisher.

3.1.1. Improvement of residual module based on channel attention

Typically, differential neural distinguishers are constructed using multiple cascaded residual structures [23]. The input of each residual structure is added to the input of the next, which helps reduce the risk of overfitting and improves the model's generalization capability. Meanwhile, the SENet attention mechanism aids the model in better utilizing information across different channels, allowing the network to adaptively adjust the importance of various channels and enhance the focus on significant features, thereby improving the classification performance [24].

In this study, we conducted extensive experimental investigations on the network width and depth of the residual module, aiming to determine the optimal number of filters, the optimal number of residual towers, and the optimal size of convolutional kernels. This was done so that the differential distinguisher could optimally acquire ciphertext feature information and achieve the highest possible distinguishing accuracy. However, after an in-depth analysis of the experimental results, we found that these adjustments did not significantly enhance the model's performance.

Through comprehensive and systematic experimental analysis and comparison, we identified the optimal residual module and its internal residual tower structure, as detailed below. Based on experimental

testing, we first selected a residual tower comprising three one-dimensional convolutional layers, with each layer having a convolutional kernel size of 3×3. Additionally, we integrated an attention mechanism module, combining the channel attention extraction structure with the residual tower. After the residual tower learned from the ciphertext data, the outputs from the convolutional layers within the tower were concatenated. This transformed the two-dimensional data (16, 32) into three-dimensional data (16, 32, n). Subsequently, the data was pooled to form a tensor of shape (1, 1, n), which was then passed through a convolutional neural network for further learning and feature extraction. The resulting weights were multiplied with the original concatenated data. The adjusted channels were summed to restore the data to its original two-dimensional form, while maintaining the overall data volume throughout the process. The network structure is shown in Figure 5.

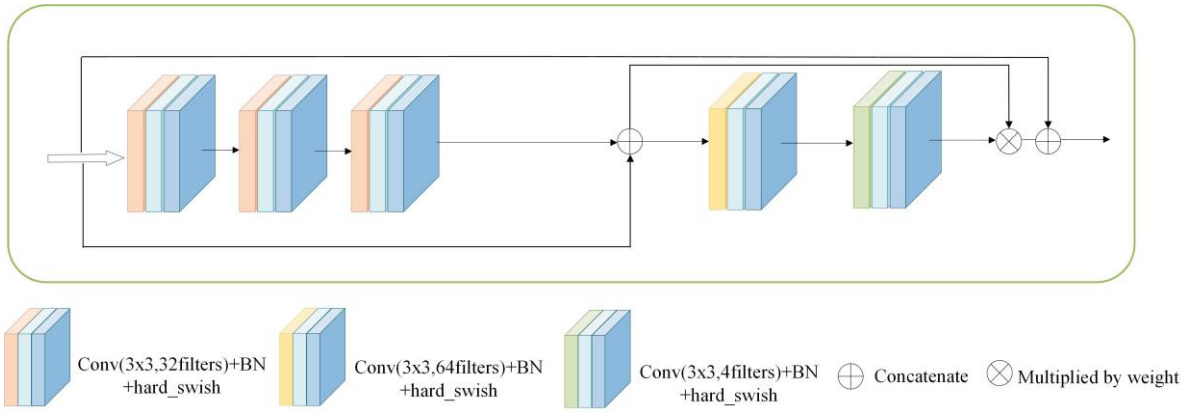


Figure 5: residual tower structures.

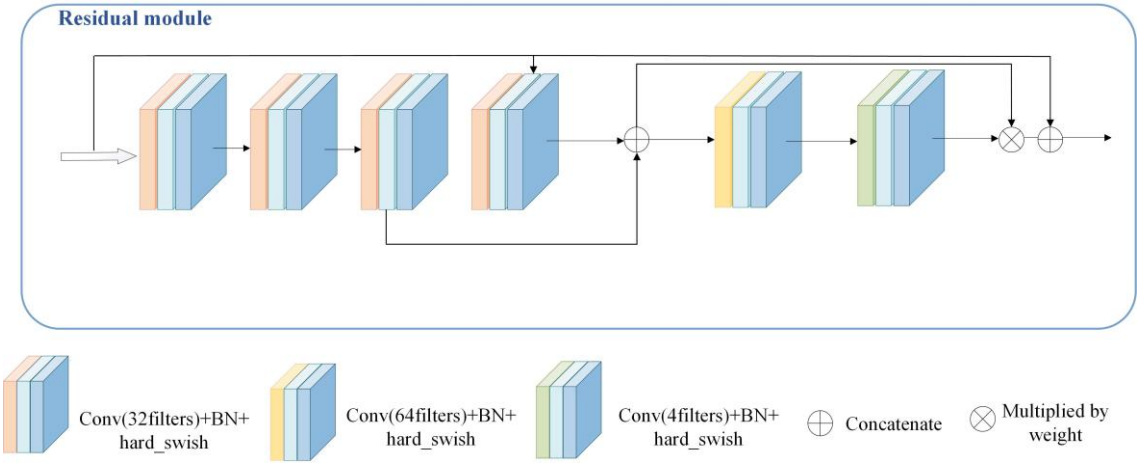


Figure 6: Branch-CNN3.

3.1.2. Structure of feature enhancement module based on convolution branch

RegNet is an efficient convolutional neural network architecture renowned for its systematic design and uniform network structure. It facilitates multi-scale learning through diverse building blocks, effectively enhancing the depth and breadth of feature extraction, while also improving learning efficiency through parameter optimization [25]. In this study, we integrate the multi-scale convolution approach of RegNet into a traditional neural network differential distinguisher, using only two stacked 3x3 convolutional layers for feature extraction. The effect of stacking two 3x3 convolutions is equivalent to the receptive field of a single 5x5 convolution. Thus, we adopt a multi-scale convolution strategy, incorporating RegNet's network design principles by adding convolutional branches to the network structure of the differential distinguisher. Additionally, we modify the activation functions to further optimize the feature extraction quality of these branches, thereby enhancing the distinguisher's feature extraction capabilities. The detailed architecture is shown in Figure 6.

3.2 Improvement of the input module

The experiments in the previous section demonstrated that modifying the neural network differentiator architecture can significantly enhance differential discrimination accuracy. However, it was also observed that optimizing the neural network structure alone is insufficient for achieving higher-round differential attacks. To further improve model accuracy and overall performance, this section focuses on optimizing the input dataset. By manipulating the dataset structure, the neural network can extract more ciphertext information and structural characteristics of the encryption function. This leads to

higher-quality, more diverse input data, enhancing the model's generalization ability, training efficiency, and overall performance.

Previous research (i.e., [26, 27]) has proposed to change the input mode from a single ciphertext pair to the input of multiple ciphertext pairs concatenated together. Experiments have shown that if the input mode is changed from inputting a single ciphertext pair at a time to inputting multiple concatenated ciphertext pairs, the differential distinguisher will learn more feature information, resulting in a significant improvement in the discrimination accuracy. In this paper, we improve on this idea by using multiple ciphertext pairs, namely p1, p2, ⋯, p32. After encrypting them for one round with a random key, c1, c2, ⋯, c32 are obtained. The differential values of these multiple ciphertext pairs and the differential values after one-round encryption (d1, d2, ⋯, d32) are concatenated and input into the neural network differential distinguisher. This model is named RKMP (Random Key Multi-Cipher Pairs), and is shown in Figure 7.

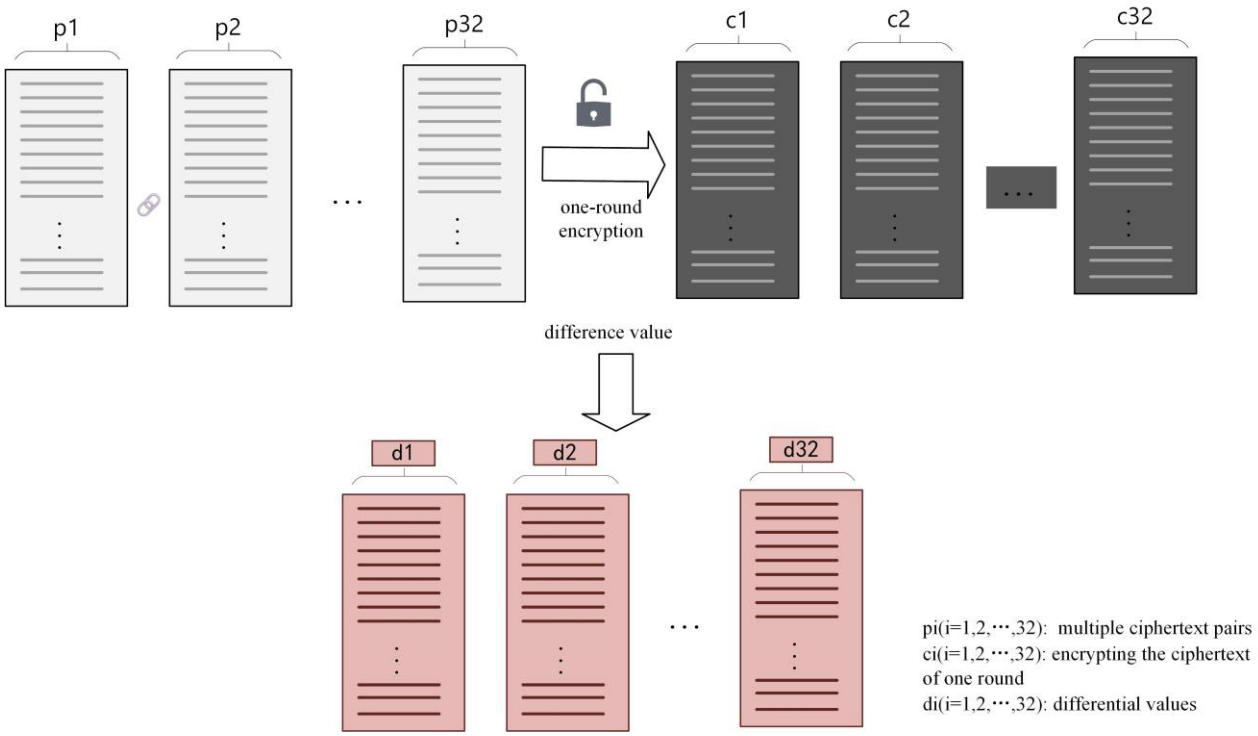


Figure 7: New input structure RKMP.

4 Experiments

4.1. Dataset

To ensure a fair and consistent experimental environment, the dataset and hyperparameters remain fixed throughout the study. Additionally, the neural network's random seed parameter is stabilized to mitigate the impact of potential floating-point variations on the results.

Network parameters: The neural network was trained on the training set for 200 epochs. The batch size was set to 5000. The Adam algorithm with default parameters in Keras was used to optimize the cross-entropy loss function, with a small penalty for L2 weight regularization (regularization parameter c = 10^{-5}). The learning rate uses a cyclical learning rate, with the learning rate l_i for $epoch_i$ set to $l_i = \alpha + \frac{(n-i) \bmod (n+1)}{n} \cdot (\beta - \alpha)$, where α = 10^{-4}, $\beta = 2 \cdot 10^{-3}$, n = 9 [28]. At the end of each epoch, the obtained network is saved, and the best network is evaluated based on the test set.

Data generation: A fixed random seed Linux random number generator was used to generate the required key, and the sizes of the training and test sets were set to 10^7 and 10^6, respectively. The fixed differential ciphertext pair was obtained by encrypting the plaintext pair with a difference of (0x0040, 0x0000) for n rounds, while the random ciphertext pair was encrypted with uniformly distributed plaintext pairs. The fixed-difference cipher was marked with a Y label of «1», and the random ciphertext was marked with a Y label of «0».

Feature Extraction: The network takes two ciphertexts (C_L^{r1}, C_R^{r1}) and (C_L^{r2}, C_R^{r2}) as inputs, which were fed into the network in the form of $(C_L^{r1}, C_R^{r1}, C_L^{r2}, C_R^{r2})$. It predicts whether the ciphertext pair conforms to the initial difference (0x0040, 0x0000). In the initial convolutional layer, a single-layer convolution operation with a kernel size of 1 is used to extract features from the input ciphertext matrices. The purpose of this step is to mimic the XOR operation in cryptographic computations. Therefore, this convolutional layer learns the convolution

of the four-bit pairs that are XORed in cryptographic operations, thereby extracting their features.

Training cost: According to the above basic training plan, in a batch size of 5000 cases, a single GTX 3090 graphics training epoch takes about 90 s. Therefore, a complete training cycle can be finished in less than a day.

4.2. Improvement of residual module

4.2.1. Improvement of residual module based on attention idea

To investigate the effect of the number of convolutional layers inside the residual tower on the performance, we designed three different residual structures (S1, S2, and S3) containing two, three, and four 3*3 one-dimensional convolutional layers, respectively. These structures are shown in Figure 8 (a), (b) and (c). We added the attention mechanism to these basic structures to form three new structures, SeNet-S1, SeNet-S2, and SeNet-S3, as shown in Figure 8 (d), (e), and (f), respectively. The above six residual structures were tested for differentiation against five rounds of Speck ciphertext pairs; the results are shown in Table 1.

Table 1: The accuracy of different distinguishers with the improved model

Network Structure	Network Depth	Accuracy
S1	10	92.95%
S2	10	92.95%
S3	10	92.92%
SeNet-S1	10	93.01%
SeNet-S2	10	93.04%
SeNet-S3	10	93.00%

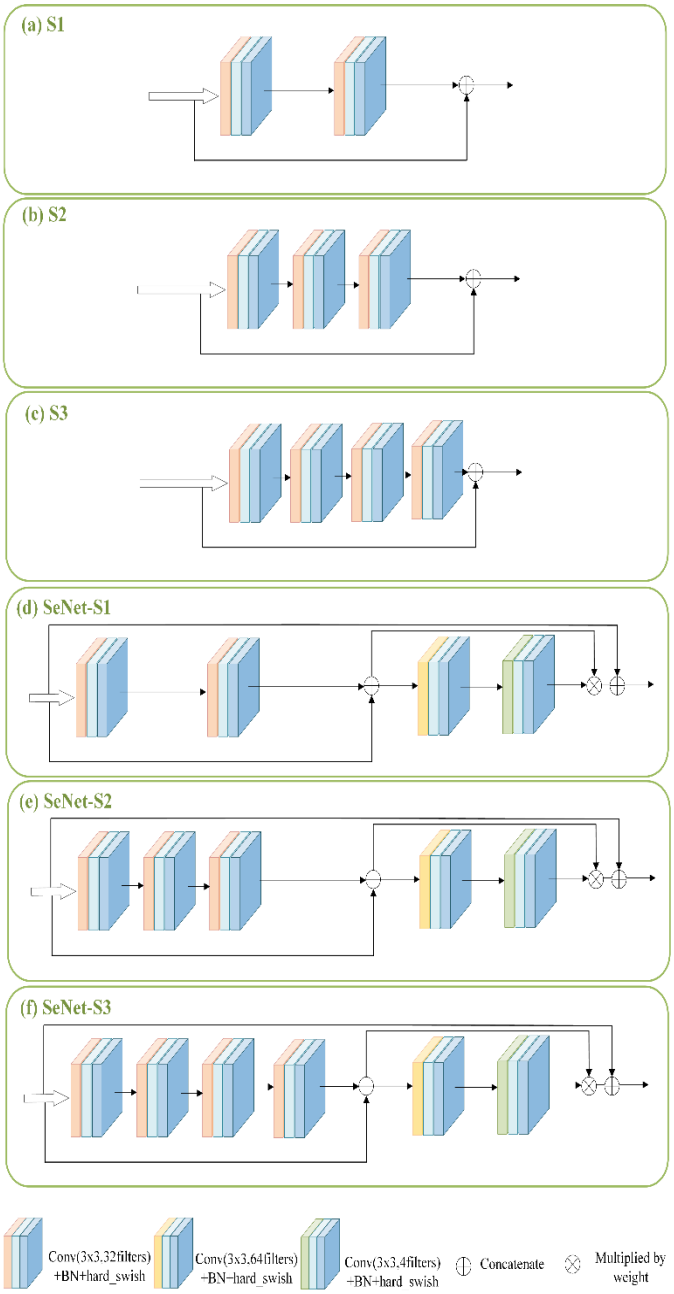


Figure 8: Six residual structures.

As shown in Table 1, SeNet-S2 demonstrates optimal performance in convolution operations. By enhancing feature representation, introducing automatic weight learning, optimizing gradient flow, reducing feature redundancy, integrating local features, and improving network stability, the model significantly outperforms its predecessors. Experimental results indicate that a three-layer convolution strikes an ideal balance between feature extraction, selection, gradient propagation, redundancy reduction, local feature integration, and model flexibility, surpassing both two-layer and four-layer configurations. This equilibrium enables the model to effectively capture complex features while maintaining computational efficiency, resulting in superior classification accuracy.

4.2.2. Structure of feature enhancement module based on convolution branch

The authors experimented with different weight learning structures and neuron counts, as summarized in Table 2.

Table 2: The accuracy of different distinguishers with the improved model

Network Structure	Add a Branch	The Structure Used for Weight Learning	Number of Neurons per Layer	Accuracy
NoBranch-MLP1(SeNet-S3)	NO	MLP	64, 3	93.04%
NoBranch-MLP2	NO	MLP	32, 3	93.01%
NoBranch-CNN1	NO	Convolution	64, 3	93.07%
Branch-MLP1	YES	MLP	64, 4	93.09%
Branch-CNN1	YES	Convolution	128, 4	93.09%
Branch-CNN2	YES	Convolution	256, 4	93.07%
Branch-CNN3	YES	Convolution	32, 4	93.11%

Table 3: Comparison of accuracy of different parameters

Activation Function	Normalization	Accuracy
ReLU	BatchNorm	93.11%
PReLU	BatchNorm	93.06%
Swish	BatchNorm	93.14%
Gelu	BatchNorm	93.14%
ELU	BatchNorm	93.06%
Selu	BatchNorm	92.99%
LeakyReLU (0.3)	BatchNorm	92.96%
Hard_swish	BatchNorm	93.16%
Hard_swish	LayerNorm	92.91%
Hard_swish	InstanceNorm	92.63%

The experimental results in Table 3 demonstrate that incorporating branch structures significantly enhances convolutional network performance, particularly for smaller neuron counts (e.g., 32 and 4). The convolutional networks consistently outperformed the MLPs in these experiments. Increasing network width proved more effective than depth in capturing input data characteristics, leading to improved model performance. Wider networks achieved higher representational capacity with fewer layers, mitigating the gradient vanishing and training challenges associated with deep networks. The shared convolution kernel in the difference distinguisher's convolutional layers effectively detected patterns across the entire input, enhancing model generalization and feature recognition. Consequently, branch addition emerges as a valuable optimization strategy.

4.3. Improving the activation function

Activation functions and normalization are crucial components in neural networks, enhancing expressivity, prediction accuracy, stability, and convergence speed [29]. Building upon the improved distinguisher in Section 3.2, this section investigates activation functions, normalization methods, and their neural network applications. Residual structures typically aim for numerical results within a specific interval in the "residual" branch. However, using the ReLU activation function at the end of this branch leads to non-negative, increasing "residual" values, potentially impacting representational power. To address this, the authors adjusted the activation function's placement or replaced ReLU with Swish, LeakyReLU, or other alternatives. Table 3 presents specific test results for five-round Speck encryption ciphertext pairs.

Normalization can improve the stability and convergence speed of the model and reduce the risk of overfitting. After an experimental comparison and analysis, it is found that the activation technic BatchNorm used in the distinguisher in this section is the optimal solution at present. Therefore, BatchNorm is still used in the distinguisher in this section, and the specific test process is shown in Table 3.

This section focuses on enhancing the neural network differential distinguisher by refining the activation function and normalization operation within the initial convolution module, residual module, and prediction

module. Employing activation functions such as Hard_swish and GeLU in place of the ReLU function yields superior outcomes for discrimination. Furthermore, the normalization method significantly impacts the performance of the Hard_swish activation function, with the activation technic BatchNorm outperforming LayerNorm and other alternatives. Consequently, the authors replace the ReLU function with Hard_swish in the initial convolution module, residual module, and prediction module of the Branch-CNN3 architecture while retaining BatchNorm. Figure 9 illustrates the overall structure of this distinguisher.

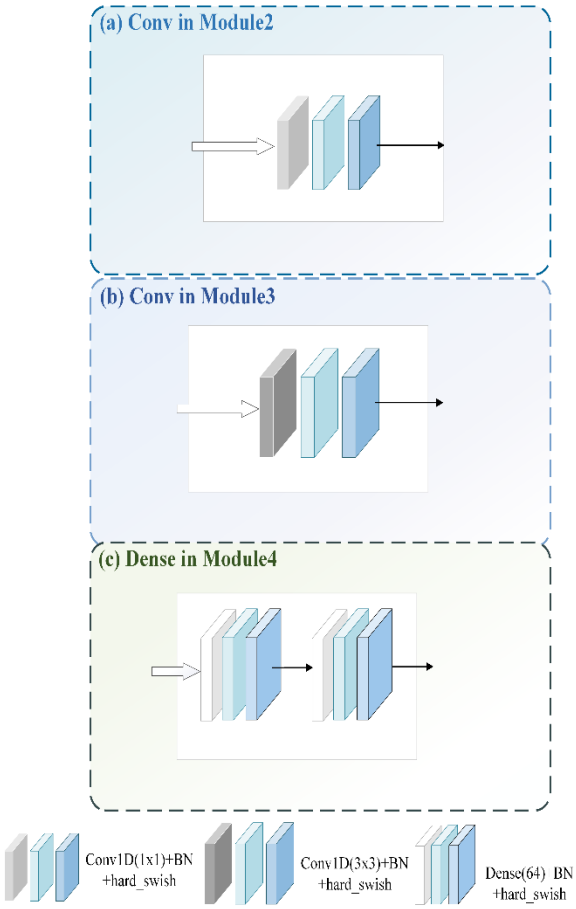


Figure 9: Activation function versus normalization improvement for Branch-CNN3.

Table 4: Comparison of accuracy of various differentiators

Round of Speck 32/64	Neural Distinguisher	Accuracy
5	CSYY22 [14]	92.6%
	Gohr [1]	92.9%
	B-C3-HSwish	93.2%
6	CSYY22 [14]	78.4%
	Gohr [1]	78.8%
	B-C3-HSwish	79.0%
7	CSYY22 [14]	60.7%
	Gohr [1]	61.6%
	B-C3-HSwish	61.7%

4.4. Comparative analysis of the accuracy of different distinguishers

At this point, the overall differential discriminator is completed and named B-C3-HSwish. The detailed structure diagram is shown in Figure 8. It was used to differentiate between five and seven rounds of the Speck encryption algorithm. A comparison of B-C3-HSwish with existing distinguishers is presented in Table 4.

In the 5-7 rounds of the Speck 32/64 encryption discrimination experiment, compared with the Gohr and CSYY22 models, the B-C3-HSwish model exhibited an accuracy advantage in different rounds. In the fifth round, the rate of improvement was approximately 0.32% compared to the Gohr model and approximately 0.65% compared to the CSYY22 model. In the sixth round, these rates were approximately 0.25% and 0.77% respectively. In the seventh round, they were approximately 0.16% and

1.65% respectively. It can be observed that under the conditions of an increasing number of encryption rounds, greater discrimination difficulty, and generally high accuracy levels, the B-C3-HSwish model can consistently maintain its advantage, thus demonstrating its effectiveness as a Speck 32/64 encryption discriminator.

4.5. Improvement of the input module

We conducted a pairwise test experiment to validate the performance of the novel ciphertext structure RKMP (random key multi-cipher pairs) in conjunction with the new splitter B-C3-HSwish, as detailed in Table 5. The results indicate that the B-C3-HSwish distinguisher outperformed the Gohr distinguisher across various round numbers, substantiating the claim that the RKMP ciphertext structure combined with the B-C3-HSwish distinguisher constitutes an optimal pairing. Despite a decrease in accuracy at higher rounds, this combination still provides effective discrimination.

Table 5: Tests of B-C3-HSwish distinguisher combined with the new ciphertext structure RKMP

Round of Speck 32/64	Ciphertext Structure	Neural Distinguisher	Accuracy
7	single-ciphertext pair	Gohr	61.6%
	single-ciphertext pair	B-C3-HSwish	61.7%
	random key multi-cipher pairs	Gohr	91.25%
	random key multi-cipher pairs	B-C3-HSwish	92.03%
8	single-ciphertext pair	Gohr	Unrecognizable
	single-ciphertext pair	B-C3-HSwish	Unrecognizable
	random key multi-cipher pairs	Gohr	63.01%
	random key multi-cipher pairs	B-C3-HSwish	63.32%

5 Conclusions

The focus of this study was to explore how neural networks can be utilized to replace traditional differential distinguishers for ciphertext pair classification. Current neural network-based differential distinguishers exhibit several significant limitations, primarily characterized by a low classification accuracy and a limited number of distinguishable encryption rounds. To address these issues, this study concentrated on two key aspects: the design of the neural network architecture for the differential distinguisher and the design of the input ciphertext structure. Building on extensive experience and existing neural network structures, the authors optimized the network architecture by improving residual modules based on channel attention, multi-scale convolutions, and activation functions in the distinguishers. We also propose a new input data structure by optimizing the input ciphertext structure, enabling the neural network to capture more ciphertext features and encryption structure information. These improvements led to the successful design of a more performant differential distinguisher, achieving an accuracy of up to 92.03% for distinguishing seven rounds of the Speck 32/64 block cipher. Furthermore, the number of distinguishable rounds was extended to eight, with an accuracy of 63.32%. The simulation results validate the effectiveness and superiority of the deep learning-based differential distinguisher design proposed in this paper for the Speck 32/64 cipher system.

In the field of cryptography, where information technology is developing rapidly and data security is of paramount importance, the security assessment of cryptographic algorithms forms the core of data security. The Speck series of cryptographic algorithms are widely used in the Internet of Things (IoT) and embedded systems due to their high efficiency and adaptability to resource-constrained environments. Taking IoT devices as an example, many sensor nodes use Speck 32/64 algorithms to encrypt and transmit data to prevent theft and tampering. However, if the algorithms' security is not secure, attackers may analyze the number of encryptions rounds to crack the encryption, resulting in privacy leakage and illegal access to data, so improving the accuracy of distinguishing between the encryption rounds of Speck 32/64 will help researchers to accurately assess the security of the encryption rounds and find loopholes, thus supporting security hardening of the devices. In embedded systems, such as smart meters and smart home control centers, which also rely on this algorithm and are closely connected to daily life, cracking the encryption will lead to serious consequences, such as confusion in energy management and threat to home security. Therefore, it is of great significance to accurately distinguish the Speck 32/64 encryption rounds to ensure the safe operation of embedded systems and maintain order. The accuracy of the distinction achieved in this study can provide a more reliable basis for the security assessment of the relevant systems in the practical applications and promote the development and improvement of the data security guarantee technology in the real world.

Although this research has made significant progress in designing differential distinguishers based on neural networks, some limitations remain. First, this study primarily focuses on the Speck 32/64 block cipher, and its applicability to other encryption algorithms has not yet been validated. Second, despite improvements in classification accuracy, performance may still be limited when dealing with higher rounds or more complex encryption algorithms. Finally, issues related to computational efficiency and resource consumption in practical applications require further optimization, and future research could explore several directions. First, the model's applicability should be tested across a broader range of encryption algorithms to verify its generalizability; second, the network architecture should be further optimized to enhance its ability to distinguish higher-round encryption; additionally, computational efficiency should be improved to make the model suitable for large-scale real-world applications, which is crucial; lastly, adaptive mechanisms should be explored that allow the model to automatically adjust to different encryption scenarios, which could significantly increase its practical value.

Acknowledgement

This study was supported by the Bengbu University 2024 Key Natural Science Research Project Bengbu University 2024 Key Natural Science Research Project (grant number 2024ZR03zd), and Bengbu University 2024 Natural Science Research Project (grant number 2024ZR03).

References

[1] Gohr, A. (2019). Improving attacks on round-reduced speck32/64 using deep learning. In Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part II 39, 150-179. Springer International Publishing. https://doi.org/10.1007/978-3-030-26951-7_6

[2] Baksi, A., Breier J., Chen Y., Dong X. (2021). Machine learning-assisted differential distinguishers for lightweight ciphers. Design, Automation & Test in Europe Conference & Exhibition (DATE), 176-181. https://doi.org/10.23919/DATE51398.2021.9474092

[3] Tolstikhin, I. O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung J, et al. (2021). Mlp-mixer: An all-mlp architecture for vision. Advances in neural information processing systems, 1857, 24261-24272. https://doi.org/10.48550/arXiv.2105.01601

[4] Chua, L. O. (1997). CNN: A vision of complexity. International Journal of Bifurcation and Chaos, 7(10), 2219-2425. https://doi.org/10.1142/S0218127497001618

[5] Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. Neural computation, 31(7), 1235-1270. https://doi.org/10.1162/neco_a_01199

[6] Jain, A., Kohli, V., & Mishra, G. (2020). Deep learning based differential distinguisher for lightweight cipher PRESENT. Cryptology ePrint Archive. 2020/846, https://ia.cr/2020/846

[7] Wang, M. (2008, June). Differential cryptanalysis of reduced-round PRESENT. First International Conference on Cryptology in Africa. 40-49.

[8] Bellini, E., & Rossi, M. (2021). Performance comparison between deep learning-based and conventional cryptographic distinguishers. In Intelligent Computing: Proceedings of the 2021 Computing Conference, 3, 681-701. Springer International Publishing. https://ia.cr/2020/953

[9] Lyu, L., Tu, Y., & Zhang, Y. (2022, May). Improving the Deep-Learning-Based Differential Distinguisher and Applications to Simeck. In 2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 465-470. https://doi.org/10.1109/CSCWD54268.2022.9776036

[10] Su, H. C., Zhu, X. Y., & Ming, D. (2021). Polytopic attack on round-reduced simon32/64 using deep learning. In Information Security and Cryptology: 16th International Conference, Inscrypt 2020, Guangzhou, China, December 11–14, 2020, Revised Selected Papers, 12612, 3-20. Springer International Publishing. https://doi.org/10.1007/978-3-030-71852-7_1

[11] Benamira, A., Gerault, D., Peyrin, T., & Tan, Q. Q. (2021). A deeper look at machine learning-based cryptanalysis. In Advances in Cryptology–EUROCRYPT 2021: 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17–21, 2021, Proceedings, Part I 40, 12696, 805-835. Springer International Publishing. https://doi.org/10.1007/978-3-030-77870-5_28

[12] Kimura, H., Emura, K., Isobe, T., Ito, R., Ogawa, K., & Ohigashi, T. (2022, June). Output prediction attacks on block ciphers using deep learning. In International Conference on Applied Cryptography and Network Security, 248-276. Cham: Springer International Publishing. https://ia.cr/2021/401

[13] Hou, Z., Ren, J., & Chen, S. (2021). Improve neural distinguisher for cryptanalysis. Cryptology ePrint Archive. https://ia.cr/2021/1017

[14] Chen, Y., Shen, Y., Yu, H., & Yuan, S. (2023). A new neural distinguisher considering features derived from multiple ciphertext pairs. The Computer Journal, 66(6), 1419-1433. https://ia.cr/2021/310

[15] Rajan, R., Roy, R. K., Sen, D., & Mishra, G. (2022). Deep Learning-Based Differential Distinguisher for Lightweight Cipher GIFT-COFB. In Machine Intelligence and Smart Systems: Proceedings of MISS 2021, 397-406. Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-16-9650-3_31

[16] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., & Wingers, L. (2015). The SIMON and SPECK lightweight block ciphers. 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), 1-6. https://doi.org/10.1145/2744769.2747946

[17] Sabonchi, A. K. S., & Akay, B. (2020). Cryptanalysis of polyalphabetic cipher using differential evolution algorithm. Tehnički Vjesnik, 27(4), 1101-1107. http://dx.doi.org/10.17559/TV-20190314095054

[18] Biham, E., & Shamir, A. (1991). Differential cryptanalysis of DES-like cryptosystems. Journal of Cryptology, 4, 3-72. https://doi.org/10.1007/BF00630563

[19] Beyne, T., & Rijmen, V. (2022). Differential cryptanalysis in the fixed-key model. In Annual International Cryptology Conference, 13509, 687-716. Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-15982-4_23

[20] Deng, L., & Yu, D. (2014). Deep learning: methods and applications. Foundations and trends in Signal Processing, 7(3–4), 197-387. https://doi.org/10.1561/2000000039

[21] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444. https://doi.org/10.1038/nature14539

[22] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, 7132-7141. https://doi.org/10.1109/CVPR.2018.00745

[23] Wang, Z., Plaxco, K. W., & Makarov, D. E. (2007). Influence of local and residual structures on the scaling behavior and dimensions of unfolded proteins. Biopolymers: Original Research on Biomolecules, 86(4), 321-328. https://doi.org/10.1002/bip.20747.

[24] Li, W., Liu, K., Zhang, L., & Cheng, F. (2020). Object detection based on an adaptive attention mechanism. Scientific Reports, 10(1), 11307. https://doi.org/10.1038/s41598-020-67529-x

[25] Schneider, N., Piewak, F., Stiller, C., & Franke, U. (2017). RegNet: Multimodal sensor registration using deep neural networks. In 2017 IEEE intelligent vehicles symposium (IV), 1803-1810. IEEE. https://doi.org/10.1109/IVS.2017.7995968

[26] Chen, Y., & Yu, H. (2021). A New Neural Distinguisher Model Considering Derived Features from Multiple Ciphertext Pairs. IACR Cryptol. ePrint Arch., 2021, 310. https://ia.cr/2021/310

[27] Hou, Z., Ren, J., & Chen, S. (2021). Improve neural distinguisher for cryptanalysis. Cryptology ePrint Archive. https://ia.cr/2021/1017

[28] Smith, L. (2017). Cyclical Learning Rates for Training Neural Networks. 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE. https://doi.org/10.48550/arXiv.1506.01186

[29] Wu, Z., Yu, H., Zhang, L., & Sui, Y. (2023). The Adaptive Quadratic Linear Unit (AQuLU): Adaptive Non-Monotonic Piecewise Activation Function. Tehnički Vjesnik, 30(5), 1469-1485. https://doi.org/10.17559/TV-20230614000735

Machine Learning-Based Analysis of Research Policy Impacts on Academic Performance Metrics

Xingfeng Zhao, Yanjie Wang*
[1]Handan Vocational College of Science and Technology, Handan City, Hebei Province, 056046, China
E-mail：03-01-11@163.com
*Corresponding author

Keywords: artificial intelligence, machine learning, research policy, citation influence, feature engineering

Received: Januar 28, 2025

Artificial Intelligence (AI) has significantly transformed research policy analysis by enabling data-driven decision-making and predictive analytics. Research policies play a crucial role in shaping academic and institutional research outcomes, influencing funding allocation, citation impact, and overall research productivity. This study employs Machine Learning (ML) techniques to examine the effect of policy, outcome, and control factors on research performance. The dataset, sourced from an online repository on GitHub, underwent preprocessing steps, including handling missing values and feature selection to identify the most influential variables. The study applies Linear Support Vector Classifier (LinearSVC) to predict three key research metrics: research productivity, citation impact, and innovation index. Experimental results demonstrate that LinearSVC achieved an accuracy of 97% (F1-score: 0.96, AUC-ROC: 0.62) for research productivity prediction, 94% (F1-score: 0.93, AUC-ROC: 0.93) for citation impact, and 95% (F1-score: 0.95, AUC-ROC: 0.96) for the innovation index. These findings highlight the robustness of AI-driven analysis in evaluating research policies and optimizing institutional strategies. By leveraging ML models, this study provides insights into policy effectiveness, enabling institutions to refine their research strategies for improved scientific innovation and societal advancement.

Povzetek: Raziskava z uporabo LinearSVC in strojnega učenja analizira vpliv raziskovalnih politik na produktivnost, citatni vpliv in inovacije.

1 Introduction

The advancements in Artificial Intelligence (AI) have transformed society by improving various spheres of life such as healthcare, enhancing communication, and driving economic growth. The growth in technology is very much dependent on research and development. Defining better research policies is essential for fostering innovation, defining living standards and addressing societal challenges. All governments try to define research policies that can help to allocate resources efficiently, guide research priorities, and promote collaboration across disciplines and borders. A *research policy* is a set of guidelines, principles, and frameworks that govern how research is conducted, funded, and applied within an organization, institution, or country. The policy defines areas for research which can be beneficial for society ensuring ethical concerns sharing with validate source and content of research study, and ensuring that research priorities meet the economic, social, and national goals [1]. It may also cover areas like funding allocation as the proper use of money should be ensured to solve top societal problems.

In the current era of AI, the use of data analytics is being used to carry out data in terms of three perspectives: descriptive data analytics, perspective data analytics and predictive data analytics. The data analytics include the latest technologies of exploratory data analysis as well as using machine and deep learning algorithms for prediction of the various outcomes that may help to define future policies [2]. In research, both policy and the data used are interlinked. More precisely, the research depends on the quality of its data and the policy of that research in a specific field. Research policy has a variety of mediated effects on policy over a range of time periods [3]. More academic research usually has longer-term effects that impact the presumptive worlds of policymakers, whereas commissioned research strives for more immediate, short-term effects [4].

However, it's also important to recognize how these cultures overlap, especially when it comes to the classifications of researchers and policymakers as well as career mobility between them [5]. As the research landscape grows more complicated, there is an increasing demand for policies that can adapt to new challenges, optimize the use of funding, and ensure that all research areas, including emerging and interdisciplinary fields, are adequately supported [6]. Machine learning (ML) techniques, with their ability to analyze and collect vast amounts of data for research policy as shown in fig 1 to uncover hidden patterns, offer a promising solution to address these challenges and transform the way research policies are evaluated to address the limitations of traditional research evaluation methods [7]. These techniques are already making strides in various sectors, including healthcare, education, and environmental

management, where large-scale data integration is critical for informed decision-making [8]. By identifying patterns of excellence and areas for improvement, AI methods contribute to more effective decision-making, ultimately fostering a more equitable research environment. This study primarily aligns with the Secondary Data and Field Experimentation Setups research policy methods. The analysis relies on structured datasets, utilizing machine learning models to assess policy impacts, making it a clear example of secondary data usage. Additionally, the study involves an experimental setup where various classifiers and feature selection techniques are applied to optimize model performance in policy evaluation. This aligns with field experimentation methodologies, as the research not only analyzes existing data but also tests different computational approaches to improve predictive accuracy and derive actionable insights for policymakers.

1.1 Research question

The study's objectives and methodology are used to define better defined research questions. Based on the above, by investigating the role of artificial intelligence (AI) and machine learning (ML) techniques in assessing the effect of research policies, this study examines their effects on the academic and institutional outcomes. Some of the research questions which the study will attempt to answer are:

1) How different research policy variables (for instance funding budget, policy duration etc.) affect research productivity, citation impact and innovation index?

Techniques of machine learning models are used to examine the correlation between the outcomes of research and the policy variables. The results show that higher amounts of funding and longer time in place of the policy have a positive influence on research productivity and citation impact. Furthermore, STEM fields policies have a higher innovation index than the social science one.

2) According to which machine learning models do research policies and their effectiveness evaluate most accurately predictions?

Five ML models were evaluated (LinearSVC, GaussianNB, KNN, RidgeClassifierCV, and ExtraTreeClassifier) and it was found that LinearSVC had the highest accuracy (97% by means of research productivity, 94% by means of citation impact, and 95% by means of innovation index). The reason for the superior performance of LinearSVC is that it can deal with high dimensional data as well as separate policy influence well.

3) What are the key influential factors which drive the research performance and how they vary across institutions or regions of different countries.

The study first uses feature selection techniques which rank the factors funding amount, researcher experience level and collaboration level as the most important drivers of the research performance. Private institutions are more oriented in terms of innovation indices, whereas research-intensive regions are more productive and more impactful from the perspective of citation.

1.2 Research contributions

In this research study, our main aim is to explore various features that act as independent variables to check their impact on the dependent which are directly linked with research outcomes including research productivity, citation impact and research innovations. The research productivity refers to the number of research publication during the research project period; citation impact refers to the average number of citations per paper in the period and innovation is associated with the innovations, products or technology that resulted based on the research projects. For descriptive analytics, exploratory data analysis and data visualization methods have been used. For prescriptive analytics and to predict the outcome variables, we have taken various relevant features and the statistical feature ranking methods such as Gain Ratio and Information Gain have been used.

Figure 1: Methods to collect data for research policies

For predictive analytics, the machine learning algorithms such as Linear Support Vector Classifier also known as LinearSVC, Gaussian Naïve Bayes, Ridge Classifier, ExtraTreeClasifier and k-nearest neighbors (kNN) have been applied. The data is prepared, preprocessed and split using holdout methods. The evaluation has been carried out using standard classification evaluation measures of accuracy, f-measures, and Receiver Operating Characteristics (ROC) Curves. The main Contribution of this study as follows:

- Analysis of the influence of various factors based on policy, outcome and control variables, providing insights for policy impacts on academic research performance.
- The application of diverse ML models to classify and predict research outcomes such as Research Productivity, Citation Impact, and Innovation Index.

- Features ranking by employing various feature engineering methods to identify top key factors in driving research outcomes.
- Proposal of a robust framework for predicting research outcomes metrics, with highest accuracy of 97% using SVM classifier for criteria of predicting high, moderate and low impact of research productivity and its influential impact in terms of citations as well.

The rest of the paper organization as follows: Section 2 presents the background knowledge of relevant literature in field of academic research outcomes. Section 3 shows the comprehensive details of applied methodology including experimental setup. Section 4 shares the analysis of results along with discussion. Section 5 provides the summary of paper in conclusion form with future directions.

2 Related work

The recent advancements in research at global level have attracted the researchers to focus on various research policies and analyzing their impact. An important discussion in research policy studies & their impact in recent years has focused on whether similarities or differences between national research policies, as well as the extent to which these policies have changed in a similar way over the recent few decades. Summary of existing studies shown in table I.

Factors influencing research performance have been categorized into three primary classifications: individual, institutional, and research self-efficacy. Individual factors encompass personal attributes of faculty members, such as age, gender, educational background, academic rank, workload, field of specialization, and research experience. These characteristics shape how researchers engage with their academic roles and contribute to their scientific output [9]. Among these factors, the influence of gender, age, and academic position on publication rates has been examined by suggesting that academic position significantly outweighs such factors in its impact on research productivity. Seniority within the academic hierarchy often correlates with greater access to resources, collaborations, and opportunities for publication [10]. Research performance has been demonstrated as a crucial factor among universities linked to research productivity, emphasizing the importance of research training, sufficient salaries, and efficient work habits. These elements collectively highlight the systemic and institutional supports necessary for fostering high domain experts to follow research policies [11]. A comparative analysis of higher education institutions across six countries highlighted a notable survey in faculty research productivity within the Arab countries. This region's relatively low research output was attributed to a combination of institutional, cultural, and systemic factors, underlying the importance of tailored strategies to address regional challenges and promote academic excellence [12]. This collectively underscores the multifaceted nature of factors influencing research performance and offers pathways for targeted interventions.

Another study highlighted the integration of ML for policy-making, demonstrating how ML models, like doubly robust estimators, mitigate biases when estimating treatment effects in large-scale policy scenarios [13]. Another emphasized for predicting societal outcomes and showcased its potential in assessing education policy impacts, where non-linear predictors improved forecasting accuracy by over 30% compared to traditional econometric models [14]. Several studies explored the role of research policies and funding in shaping academic outputs. Analysis shows that funding mechanisms influences citation-based performance and research productivity, which directly impacts the diffusion of new scientific knowledge in society. Specifically, models linking funding to citation growth reveal that well-supported research is more influential factor for research productivity [15]. Citation impact evaluation methods highlighted the relationship between publication metrics and academic impact. Using fi-score, as an unbiased measure for self-citations and external influences, complement traditional metrics like the H-index for evaluation of academic productivity [16]. Societal research evaluation extends beyond academia, assessing societal benefits such as contributions to public health and environmental sustainability [17]. However, further research is needed to address the inherent risks of bias, transparency, and reproducibility while advancing theoretical and methodological approaches to ensure equitable outcomes for all researchers.

Table 1: Summary of prior work

Sr. No	Ref	Year	Model	Results (%)	Dataset	Features
1	[9]	2021	SVM, AI-driven Governance System	Acc: 70	Survey data from American research universities.	Impact factor, grants
2	[11]	2021	LR, DT, SVC	Acc: 80	Interviews, Meetings	Average count, mean of citation count
3	[12]	2022	Bibliometric analysis, SPSS	Pre: 79	Arab literature (documents published between 2006–2015)	Citation counts before and after the Arab Spring
4	[13]	2022	XGB, RF, SVM	Acc: 85	scholarly research articles on sustainable finance	Citation Index Impact Factors

5	[14]	2020	SVM, KNN	F1: 73	Research publications funded Canadian researchers (2000–2018).	Textual Features
6	[15]	2022	SVM, NB	Acc: 79	Academic history of Brazilian researchers	Case studies and Surveys
7	[17]	2020	CNN, SVM	Pre: 77	Policies and practices data in environmental and sustainability education	Institution, Funding, Collaborative Index
8	[18]	2024	KNN, Adaboost	Acc: 89	Transactional data	Citation Impact, Funding, Impact Factor

3 Proposed research methodology

In this section, proposed research methodology, as shown in fig 2, has been explored by applying feature engineering techniques and various ML models training apply on dataset, evaluated using standard parameters of classification measures such as accuracy, F1-Score and AUC-ROC.

We develop a structured bias assessment framework to better guarantee fairness and see potential biases in our study. To ascertain if bias exists, one conducts Exploratory Data Analysis (EDA) such that they examine dataset composition to reveal composition, and whether there's any underrepresentation of events. Fairness metrics as accuracy, precision, recall, F1–score, and AUC-ROC is used for bias diagnosis and enables a quantitative measure of disparities in model performance. Hence, we optimize the model fairness by setting right values of hyperparameters and balanced learning across different groups to mitigate bias. Moreover, Information Gain and Gain Ratio are used to analyze feature contribution in order to enhance interpretability and model transparency by confirming if a feature has an impact on model decision. Fairness evaluation dashboards and MLOps continue to monitor bias continuously, so as fairness can be tracked over time and necessary adjustments can be taken to keep the model reliable and fair. Use of this structured approach makes our study more robust, and introduces it to the practices of fairness aware machine learning.

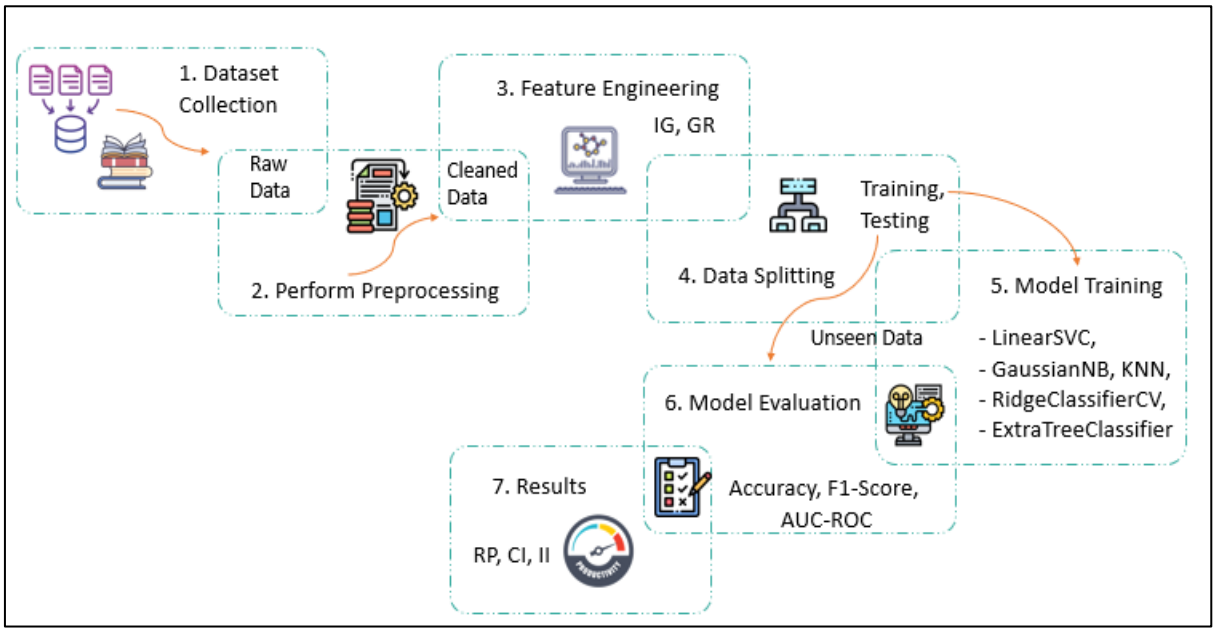


Figure 2: Framework showing steps of of proposed research methodology

3.1 Preprocessing

Preprocessing is a critical step in the data quality as well as improving the model's result's performance. Finally in this study, Min-Max Scaling has been applied to normalize feature values within a predefined range so that there is consistency and there are no biases in the training of the model. Information Gain and Gain Ratio were used for feature selection that will associate and retain the most important attributes to reduce the dimensionality and

computational complexity while keeping the key data patterns. The tabular dataset, data cleaning was done to remove inconsistencies and outliers and thus to have a structured and reliable dataset that can be analyzed. Additionally, the missing values were taken care of using appropriate imputation techniques to avoid information loss while keeping the integrity of the dataset. Together, these preprocessing techniques strengthen the robustness of the dataset, making the reliability of subsequent modeling marches and the probability of a reliable overall accuracy and generalizability of the study's findings.

3.2 Feature engineering

Feature engineering is the process of transforming, selecting and creating informative attribute from raw data to improve accuracy, enhance interpretability of machine learning models. Information Gain (IG) is used for data analysis, measuring the reduction in uncertainty in machine learning models. First calculate uncertainly variables, split data into subset, calculate entropy of variables and then calculate information gain of each feature. Feature extraction using information gain to rank feature and select top ranked features. This technique is effective in handling high dimensional data and easy to interpret. Furthermore, Gain Ratio (GR), used for data analysis, handles multi values attributes under splitting ratio of 0 and 1, to improve feature selection stability. It ensures that selection is not based only on those features that have a high number of distinct values and provides a fairer way to carry out feature selection for cases where the decision tree algorithm will be used [18]. The method is particularly suitable for the use with the large data that can have various features.

$$GR = Information\ gain/Split\ information \quad (1)$$

GR and IG are necessary metrics to use with decision trees for selecting attributes. When splitting a dataset on a particular attribute, Information Gain does whatever measurement is defined as Information Gain, which is usually what outcome expected, namely how much the entropy of the resulting datasets decreases. It also finds the most informative features for classification. The entropy of a dataset D is given by equation 2:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i) \quad (2)$$

p_i is the probability that an arbitrary tuple in D belongs to class C_i, $|C_i, D|/|D|)$ is the estimated probability.

After splitting the dataset D using attribute A into v partitions D_i, the entropy for the partitions is given by equation 3:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j) \quad (3)$$

To address the attrition issue caused by having many values for some attributes, Information Gain is normalized by Gain Ratio. It is calculated as in 4:

$$Gain(A) = Info(D) - Info_A(D) \quad (4)$$

However, Information Gain tends to favor attributes with many unique values, which may not always be optimal. To counteract this bias, the Gain Ratio (GR) is introduced, defined as in 5:

$$GR(A) = \frac{Gain(A)}{SplitInfo(A)} \quad (5)$$

where $SplitInfo(A)$ computed as using equation 6.

$$SplitInfo(A) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \log_2\left(\frac{|D_j|}{|D|}\right) \quad (6)$$

These equations ensure a robust and unbiased feature selection process, improving the interpretability and efficiency of classification models.

Information Gain (IG) and Gain Ratio (GR) were used for feature selection to improve the models toward the predictive capability. The measurements of entropy reduction from splitting data on specific attribute combination is what IG can do best, and GR is able to normalize a metric for dealing with values of a categorical variable with more than one. Computational limitations and the concerns of interpretability in policy-based decision making implied that other methods like Recursive Feature Elimination (RFE) or Mutual Information were not used. In the study, a holdout validation was employed with the dataset split into 80% training, 20% testing for generalization performance assessment. This supports the use of AI driven methodologies in research policy evaluation to create data for policymakers to implement funding strategies more efficiently as well as institutional policies.

3.3 AI-based applied models

To evaluate the effectiveness of various ML techniques in predicting research outcomes, several classification models were applied to explore the performance of models, highlighting their strengths and ability power for classifying research-related outcomes that helpful in decision making for optimizing policy and research innovation.

3.3.1 Gaussian Naïve Bayes (GNB)

GNB is a features independent classifiers use for probabilistic based theorem and effective for high dimensional data. For computing, this uses the following formula that shows in equation (7) and (8).

$$P(y|x) = \frac{P(x|y) * P(y)}{P(x)} \quad (7)$$

$$P(x|y) = \frac{1}{(\sqrt{(2\pi\sigma\text{^}2)})} * \frac{exp(-((x-\mu)\text{^}2)}{(2\sigma\text{^}2))} \quad (8)$$

3.3.2 K-Nearest Neighbors (k-NN)

KNN is a lazy method of learning that makes prediction based on nearest neighbors use for non-linear relationship and handles categorical features, while K-NN is a not parametric classifying technique. A data point gets assigned a class that reflects the majority class of its k-nearest neighbors. The KNN compute show in (9).

$$d(m,n) = \sqrt{\sum_{i=1}^{k}(mi - ni)2} \quad (9)$$

3.3.3 Extra tree classifier

An approach to ensemble learning called Extra Trees (Extremely Randomized Trees) mixes predictions from several unpruned decision trees. This model is effective for

nonlinear relationship and fast training model. It leverages random splits for features, in opposition to Random Forests. Based on variants of Gini index equations, the data points are split using the equation that achieves the highest value, ensuring optimal decision tree construction. The equation of extra tree classifier shows in (10).

$$\Delta G = G(D) - (\frac{|D|}{|DL|}G(DL) + \frac{|D|}{|DR|}G(DR)) \quad (10)$$

3.3.4 Linear SVC

A linear classifier called Linear SVC produces the hyper plane that reduces the distance between classes [19]. This model is fundamental in ML and used for handling high dimensional data and robust noise. Linear SVC objective function show in equation (11) and (12).

$$y = w\hat{}T\ x + b \quad (11)$$

Subjected to.

$$L(w,b) = \sum(max(0, 1 - y_i(w\hat{}T\ x_i + b))) \quad (12)$$

3.3.5 Ridge classifier CV

To prevent over fitting, the Ridge Classifier regulates a linear model. By separating data into training and testing sets, cross validation analyzes model performance. This model is used for reducing over fitting, handling high dimensional data and imbalanced data. It penalizes the magnitude of the coefficients with the goal to minimize the squared error. Ridge classifier objective function show in equation (13).

$$L(w,b) = \sum(y_i - (w\hat{}T\ x_i + b))\hat{}2 + \alpha * ||w||\hat{}2 \quad (13)$$

3.4 Hyperparameter settings

A holdout validation approach was applied, splitting the dataset into 80% training and 20% testing to evaluate the model's generalization ability. The models were trained using LinearSVC, GaussianNB, KNN, RidgeClassifierCV, and ExtraTreeClassifier, with hyperparameter tuning performed. The hyperparameter tuning process was conducted on LinearSVC, the best-performing model, to optimize its parameters. The following table II summarizes the tuned hyperparameters:

TABLE I. HYPERPARAMETER SETTINGS

Parameter	Values Tested	Optimal Value
penalty	l1, l2	l2
loss	hinge, squared_hinge	squared_hinge
C (Regularization)	0.001, 0.01, 0.1, 1, 10	1
max_iter	500, 1000, 5000	1000

These settings helped LinearSVC achieve the highest accuracy for predicting research productivity. Table III defines the variable description used in methodological equations for deeper understanding Performance Evaluation Measures.

4 Experimental setup

In the experimental setup section, discuss dataset, the process and setting used to carry out the experiments and evaluate the models for impact of research policy.

4.1 Dataset

For this study, the dataset was sourced from an open-source repository on GitHub so that it can be as transparent and reproducible as possible. The file consists of 4,000 entries with policy, outcome and control variables used for policy evaluation. Preprocessing was carried out on the dataset by handling the missing values in the numerical features using mean imputation and the categorical features using mode imputation. Policy impact, funding allocation and institutional effectiveness are preferred factors considered in the dataset, and therefore it serves as a solid foundation for analysis regarding the effect of research policies on productivity and innovation.

For data analysis, the dataset is divided into three categories policy variables, outcome and control variables. The policy variables also consist of five columns include funding amount in dollar, policy duration in year, target audience that category into three values universities, private research centers and public research institutions, priority areas categorically into STEM, health and social sciences and policy types that divided into three categories include grant, tax incentive and loans. The second categories are outcomes that consists of four columns include research productivity that tell about numbers of papers, citations impact shows the average citations per paper, innovation index includes, collaboration level and research quality score, region that consists of diverse regions including Asia, Africa, Europe, North America and Oceania with institution type that is private and public, researcher experience level consist of senior and junior.

4.2 Performance evaluation measures

Performance evaluation matrix is used for comparing performance of different models, model improvement and comparison with baselines. Use multiple metrics for comprehensive evaluation such as accuracy, f1-score, and AUC-ROC [20].

4.2.1 Accuracy

It shows the proportion of cases that are accurately expected and Proportion of accurately expected instances % of total instances, as in eq (14).

$$Accuracy = \frac{(TP+TN)}{TN+FN+FP+TP} \quad (14)$$

4.2.2 F1-score

The F1-Score manages the trade-off between precision and recall by taking the harmonic mean of both, as in eq (15). When the distribution of class is unbalanced, the F1-Score is especially helpful.

$$F - score = 2 * \frac{(\frac{TP}{(TP+FP)} * \frac{TP}{(TP+FN)})}{(\frac{TP}{(TP+FP)} + \frac{TP}{(TP+FN)})} \quad (15)$$

4.2.3 AUC-ROC

AUC-ROC measures the ability of a model to predict and plot true positive rate against the false positive rate and used to robust class imbalance, comparability of different models.

Table 2: All symbols with abbreviations

Symbols	Abbreviations
μ	mean
$(\sigma)^2$	variance
x	feature value
y	class label
m	Query point
n	A neighboring point
k	Number of features
$G(D)$	Impurity measure.
$DL\ and\ DR$	Left and right subsets after the split
$\|D\|$	Total number of samples.
C	regularization parameter
w	weight vector
b	bias term
TP	True positive
TN	True negative
FP	False positive
FN	False negative

All the above performance evaluation metrics are essential for evaluating the performance of classification models, especially in scenarios where class distribution and the cost of errors vary.

5 Results analysis

In this section, we examined the results through feature engineering techniques and machine learning models to analyze research policies impact for the purpose of classifying the various research outcomes based on the selected features.

5.1 Exploratory data analysis

The correlation matrix as shown in fig 3 provides a visual representation of the relationships between numerical variables in the synthetic dataset. The figure presents the correlation matrix that shows the insightful analysis of the relationships between some research policy factors. The heatmap shows — visually — correlations coefficients between the variables, where -1 to 1 is the value and red shades represents the highest positive correlations, and the blue shades is highest negative or negative correlations. A couple of observations of interest is the 'strong positive correlation' (0.58) between Policy Duration and Citation Impact, that longer interventions can facilitate higher citation impact. Research Productivity is also moderately related to Funding Amount (0.48) and Citation Impact (0.25), which implies that funding of some kind is essential to achieving high research output. Curiously, the Collaboration Level and Research Productivity have a moderate positive correlation (0.48) revealing the significance of the collaborative nature of academia. Priority Area, however, has weak or negligible correlations with almost all the variables indicating that it has relatively poor direct impact on most of the research success metrics. Thus, findings highlight the role of funding allocation, collaboration and policy longevity in determining whether an uncommitted fund dedicated to research has an impact and promotes the effectiveness of research. Overall, the correlation matrix reveals several notable relationships between the variables in the synthetic dataset by defining moderate positive and weak negative correlations.

The pair plot shown in fig 4 provides a comprehensive visual overview of the relationships between numerical variables in the synthetic dataset. The dataset displays a wide range of funding amounts, from approximately 0 to 20 million, which suggests that the dataset is not artificially constrained and encompasses a variety of funding scenarios. Pair plot visualization shows all relationships among various policies and outcome variable over several regions. Individual variable distribution has also been shown in the diagonal elements, with patterns such as skewness in funding amount and collaboration level, which indicate that these variables may need to be normalized or transform. The scatter plots demonstrate the relationships between various variables, including funding amount and research productivity as well as citation impact, and have positive correlations. This is in line with the existing literature, which indicates that research funds greater publication output and research citation impact. However, their relationships with funding and innovation index, besides collaboration level and research quality score, exhibit more dispersed distribution, which suggests the influence on the result beside funding. In general, the figure summarizes what patterns are to be looked for in future regression or classification analysis to quantify the efficacy of policy variables in influencing research performance.

As shown in fig 5, the boxplots that are presented display the distribution of numerous research-related variables including the amount of funding, the duration of the policy, the productivity of research, the impact of the citation, the success rate of funding, the innovation index, collaboration level, and research quality score. The distribution of the funding amount is very high skewed with a few very large outliers, indicating that while most of the research projects receive small amount of funds, some projects are funded by few times more compared to most of them. Policy duration is close to a normal distribution— it means that it is consistent in policy timelines. The compact IQR for both research productivity and citation impact indicate that most values fall in a narrow range, but outliers indicate a variation of productivity across researchers. As you can see it is very skewed meaning the funding success rate can be very low while a percentage makes it through to pass. Looking at the innovation index, it shows a greater spread meaning innovation levels of different research outputs vary. Research projects at

collaboration level show a right skewed distribution, i.e. there are research projects with extensive collaboration, a larger number of low collaborations. The last one involves research quality scores, which show a quite narrow distribution with some extreme values, meaning that most research is of a quality appropriate to the benchmark, but some projects excel or regress far more than others. However, these insights are useful when it comes to optimizing the policy, as they assist in discerning the impact of funding, collaboration, and duration of the policy on research productivity and quality.

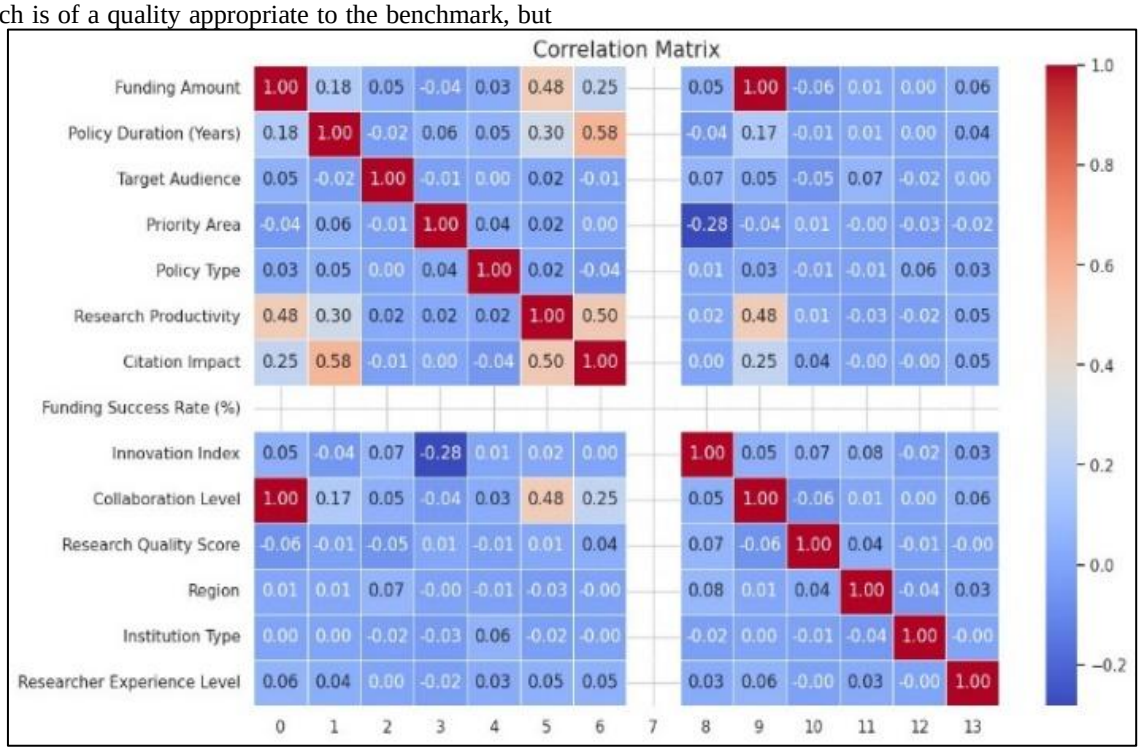


Figure 3: Analysis of correlation matrix for variable relationship

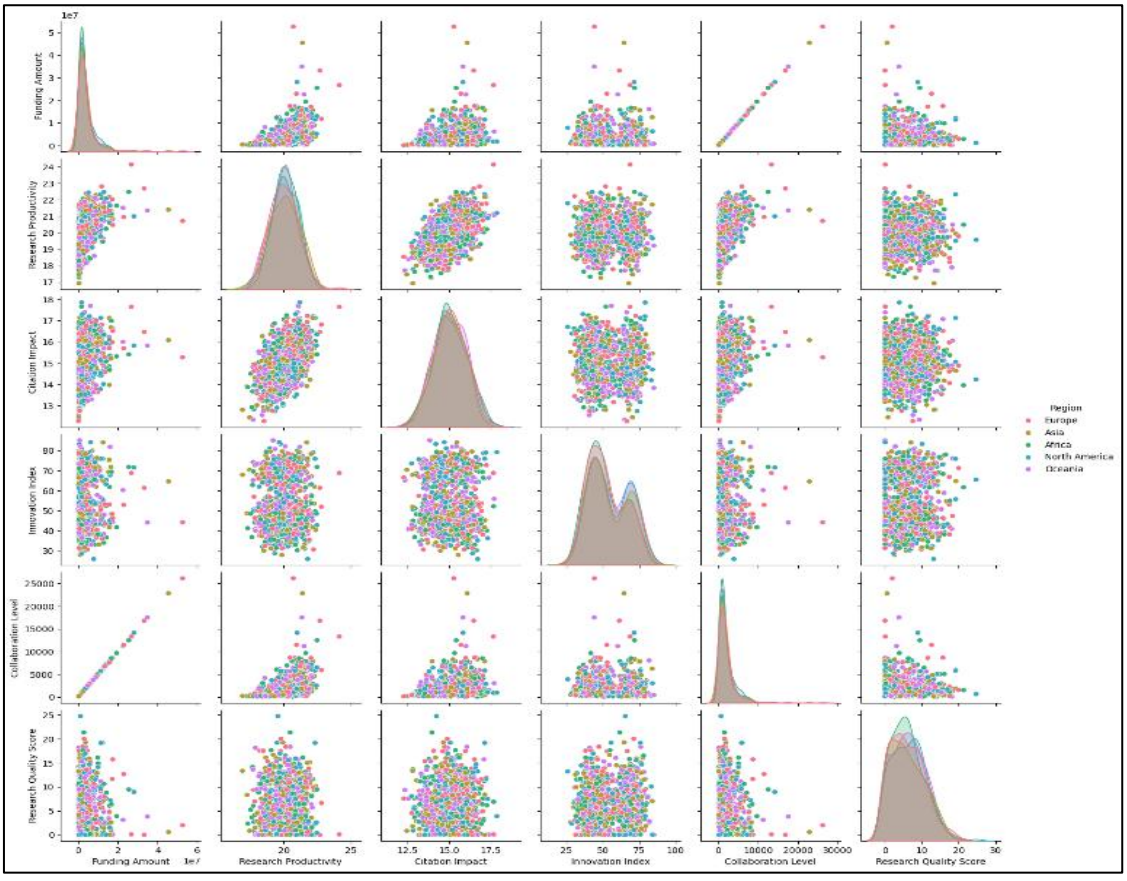


Figure 4: Analysis of dataset using pair plot visualization

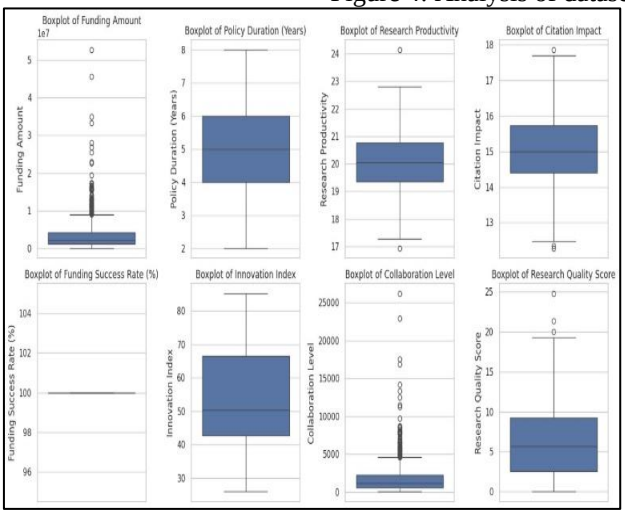


Figure 5: Summary of distribution of numerical variables

Fig 6 depicts the Funding Amount and the Innovation Index, while splitting the data by Institution Type and the researcher's Experience level. The resultant scatter plot shows the relationship between funding amount and the innovation index using the data points split by institution type (private versus public) and researcher experience level (junior versus senior). Most data points tend to occur at lower levels of funding indicating that most of the institutions are not getting much in terms of finances. Although innovative institutions have lower funding than others, several still show very high innovation indices and innovation is not solely a question of financial investment. The distribution of the innovation scores looks quite similar between public institutions (orange) and private institutions (blue), and senior researchers (x marked points) are associated with higher scores generally. However, at larger dimensions of funding, there is increased variability of innovation outcomes with some highly funded institutions as well as others with lower innovation indices. This can be interpreted to imply that in addition to funding, culture research, collaboration and policy frameworks may also contribute to generate innovation.

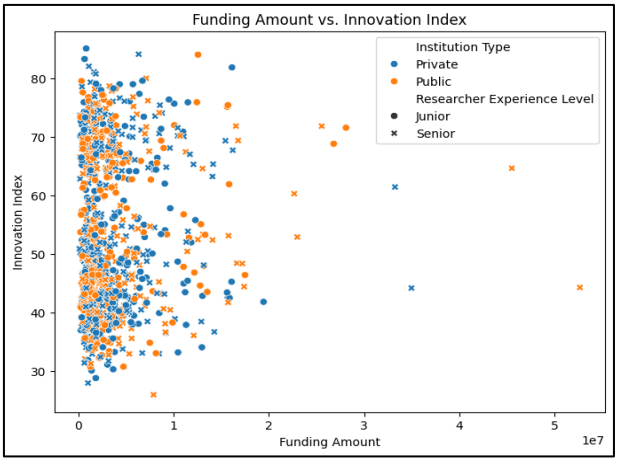


Figure 6: Funding amount and the innovation index across institution type

Fig 7 visualizes the distribution of Research Productivity across five global regions: European continent, East Asian, African, the American and Australian continent. The distributions suggest that median productivity within regions is slightly different and varies slightly as far as spread is concerned. Europe and North America being marginally higher and the distance between the median and mean are narrow suggesting that the researchers are more consistent in their productivity. On the other hand, Asia and Africa presented broader dispersion which may imply higher variability in the productivity ranges for these two regions. Oceania remains comparable to the distribution witnessed in Europe and North America hence moderate candor. In general, this map allows us to identify differences in regional performance and distribution of high and low research outputs, although research productivity is quite comparable between various regions.

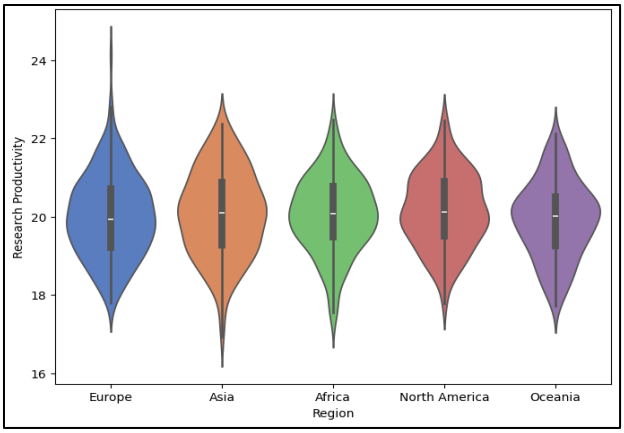


Figure 7: Distribution of research productivity across regions

Fig 8 shows trends of innovation over time, the relationship between different research indices and the co-authorship networks for researchers grouped by experience and institutional affiliation. The first visualization, a density plot, examines the distribution of the Innovation Index across three policy types: Grant, Loan, and Tax Credit. This is justified by the fact that the data has a bimodal pattern for all the policies at around 45 and 70. However, the Tax Incentive policy presents a steeper slope in the lower peak, which means that it encourages innovative outcomes in the lower range score. Regarding the distribution of the densities with the articles, we observe slightly higher densities centered around the first upper peak, which is approximately 70, thus the loan policy may enhance the improvement of innovation performance more than grants.

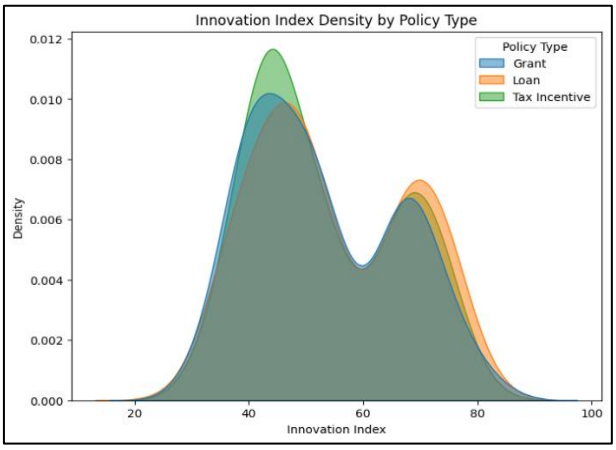


Figure 8: Trends of innovation index by policy type

Fig 9 shows how Collaboration Levels of Junior and Senior researchers differ between Public and Private entities. The results of junior researchers indicate that several collaboration level clusters are at low levels, with few points hitting elevated levels, thus illustrating that inconsistent collaboration possibilities exist for early-year researchers. On the other hand, the results presented for Senior researchers show a greater variability, with the relatively higher end of collaboration scoring even more frequently represented, especially in the category of public institutions with some of the outliers tipping the 20,000 marks. This further brings out the key factors in Research Collaborations which include experience and institutional support where senior researchers with connections take advantage of these.

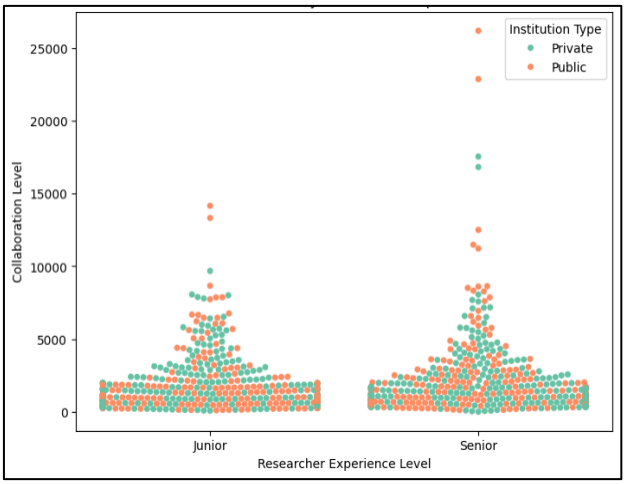


Figure 9: Collaboration level by research experience level

5.2 Analysis of results using various AI based feature engineering techniques

Feature selection methods including IG and GR were employed to determine the relevant factors in predicting top features that assist in classification by providing the measure of the ability to improve the state of uncertainty to identify which features and variables are most influential so that the target labels can be accurately predicted.

5.2.1 Policy Variables

Among the policy variables, detail analysis of results has been defined using table IV, revealing that funding amount showed the highest information gain (0.089) and gain ratio (0.134), showing the most influential factor in determining research outcomes. This suggests that the amount of funding plays a significant role in influencing metrics such as productivity and impact. Policy duration also has a measurable, though smaller, influence with an information gain of 0.011, showing that longer or shorter policy periods have some impact on research results. This outcome may indicate that, beyond funding amount and duration, other policy factors are less influential in determining research success across the examined metrics.

Table 3: Result analysis of policy-based variables

Policy Variables	Info Gain	Gain Ratio
Funding Amount	0.089	0.134
Policy Duration	0.011	0.002
Priority Area	0.001	0.001

5.2.2 Control Variables

Among the control variables, in table V, region and institution type exhibit similar levels of influence with information gains of 0.010 and 0.011, respectively, indicating that the geographic location and type of institution (private or public) have a small but measurable effect on research outcomes. Researcher Experience Level has a very low information gain (0.002), implying that the experience level of researchers contributes little to the variance in the outcome metrics in comparison to other factors.

Table 4: Result analysis of control-based variables

Control Variables	Info Gain	Gain Ratio
Region	0.010	0.002
Institution Type	0.012	0.001
Research Experience	0.002	0.002

5.2.3 Outcome variables

In outcome variables, the most crucial feature research productivity with information gain of 0.525, indicating as a strong predictor of overall research impact, as shown in table VI. Citation impact follows with an information gain of 0.070, suggesting that it also plays a role in understanding the effectiveness of research policies, though to a lesser extent than productivity. Innovation index shows very low information gain (0.002), indicating that it may be less directly influenced by the factors analyzed, or that it captures different aspects of research outcomes not directly tied to the measured policy and control variables. Research quality score shows no information gained about 0.001, suggesting that it may not

be effectively captured or influenced by the policy and control variables.

In summary, funding amount and research productivity are the top-ranking features based on information gain, indicating that they are the most influential variables in determining research success. Control variables like region and institution type have modest influence. These findings suggest that while funding and productivity are critical factors. The findings emphasize the importance of funding allocation and indicating that policies could be optimized by prioritizing funding support to enhance productivity, as well as by considering regional and institutional contexts.

Table 4: Result analysis of outcome-based variables

Outcome Variables	Info Gain	Gain Ratio
Research Productivity	0.525	0.001
Citation Impact	0.070	0.005
Innovation Index	0.002	0.001
Research Quality Score	0.001	0.001

6 Discussion and comparison with prior work

To classify the research findings, the features were further explored using machine learning models including LinearSVC, ExtraTreeClassifier, GaussianNB, KNeighborsClassifier, and RidgeClassifierCV. The models' performance was assessed using accuracy and F1-score where possible, based on three outcomes, as comprehensive results were displayed in table VII. This study clearly helped to identify which policies are most recommended and insightful for high quality research and innovation.

Table 5: Results of all three metrics using ML Models

Research Productivity			Citation Impact		Innovation Index	
Models	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**
GaussianNB	0.89	0.90	0.91	0.91	0.91	0.91
KNeighbors Classifier	0.87	0.85	0.80	0.78	0.84	0.84
RidgeClassifierCV	0.88	0.86	0.71	0.61	0.85	0.85
ExtraTreeClassifier	0.90	0.93	0.94	0.94	0.82	0.82
LinearSVC	**0.97**	**0.96**	**0.94**	**0.93**	**0.95**	**0.95**

The findings also show that LinearSVC perform better prediction accuracies than other models in outcome measures. As for research productivity, LinearSVC scored an accuracy of 97% and an F1-score of 96%: the data clearly shows its efficiency in differentiating between the research categories. Similar improvement was equally observed in the innovation index where LinearSVC achieved at equal score with both measures, accuracy and F1-Score of 95%, signifying its efficiency in predicting the degree of adopted innovation prompted by research policies. Moreover, with the citation impact, using ExtraTreeClassifier achieved accuracy and f1-score of 94% equally as the most efficient classifier among the four, although its prediction accuracy was equivalent to LinearSVC at about 94% and an F1-Score of 93%. For research productivity, the model was useful with an accuracy of 90%. The ExtraTreeClassifier emerges as a fundamentally accurate approach in implementing ensemble learning models for complicated datasets and policy implications. Furthermore, GaussianNB as a much simpler model also had high performances concerning all the outcome variables, as with citation impact overall accuracy and F1 score of 91%. For research productivity, it achieved 89%. This makes the model effective while at the same time, it is complex and can be readily interpreted in cases where interpretability of a model is a necessity. On the other hand, the performance of models such as KNeighborsClassifier and RidgeClassifierCV has shown to be middle level. When the KNeighborsClassifier was used, a score of 87% for research productivity, 80% for citation impact and 84% for innovation index were obtained, revealing the limitations in effectively classifying outcome variable. This model struggled the most, particularly for citation impact, where its accuracy was 71% and its F1-score dropped to 61%, suggesting that it may not be well-suited for datasets with complex, nonlinear relationships. As comparative analysis on the base of accuracy measure has been shown in fig 10.

The contribution of this study is in how ML techniques can be used in the evaluation of research policies using policy, outcome, and control variables. Compared to other classifiers, LinearSVC has performed well due to its robustness of dealing with high dimensional categorical and numerical data, its ability to efficiently discover the important factors that affect research productivity, citation impact, and innovation index. For linear decision boundaries adapted well in high dimensional spaces, and a probabilistic approach to generalize categorical and continuous data were given by GNB, while LinearSVC was chosen due to its effectiveness. To enable interpretability and interpret feature importance estimation, ETC was selected as an ensemble method. Ensemble techniques such as Random Forest and Gradient Boosting Machines were considered to further enhance the model diversity and nonlinear pattern. Finally, it is widely known that these methods are robust with regards to complex, heterogeneous datasets and aggregate multiple weak learners to prevent overfitting. Further work may be possible with these advanced models to provide better predictive performance and generalizability.

The ML approach is more accurate and powerful than prior studies which used traditional statistical models in relating research policies to their different outcomes. An underperformance of some models may be explained by the characteristics of the used dataset, such as the irregularity of the funding success rates and the varying policy effectiveness across regions. Furthermore, biases in data collection such as differences in research priorities, institutional funding structures, and the way policy is implemented across regions may affect generalizability of

the model. A relatively lower AUC-ROC (research productivity 0.62) indicates that factors beyond calculating the dataset may not be captured entirely, for instance, external factors like researcher motivation and institutional support may not be captured. Future research concerning the first two problems mentioned should focus on ensemble learning approaches and domain adaptation techniques, since these will mitigate biases, enhance model reliability and improve predictive performance in research policy evaluation.

To ensure contextual relevance, the results of this study should be interpreted within the broader framework of research policy optimization. It is demonstrated that funding amount is highly correlated with research productivity, which suggests that the intended financial support indeed plays qualitatively critical roles in improving publication rates. This is consistent with the existing literature on policy that pays close attention to funding in stimulating high impact research. The research suggests that policy duration and priority areas shape the impact of citation, so long-term and strategically targeted formulation of policy is the way to impactfully study the research. This stands as support for previous studies that show that investment in targeted funding initiatives is associated with sustained impacts in academics. The results indicate the impact of institutional type and researcher experience in determining research quality and collaboration. In accordance with previous studies on the institutional research culture and its effect on product quality, public institutions and senior researchers have higher innovation indices.

Building on the ML based ranking of influential factors, the study contributes with an empirical basis of improving policies for research. LinearSVC is predictive (97% in the case of research productivity, 94% for the citation impact, and 95% for the innovation index) and thus, supports that it is fair to use AI-driven policy evaluations. Correlation analysis as shown in fig 3, supports what is expected from policy literature, namely equitable funding distribution, long term planning and supportive institutional environment, which lead to success of research. Although, some surprising result shown, such as the moderate AUC ROC scores in fig 11,12, and 13 for some variable indicate that data driven policy assessment still needs be further refined. Such insights from future research may also include other socio-economic and interdisciplinary factors. This study utilizes AI for developing a framework for data-based decision making for evaluation of research policy and contributes to guide institutions and funding agencies to maximize the research benefits for society by maximizing available resources.

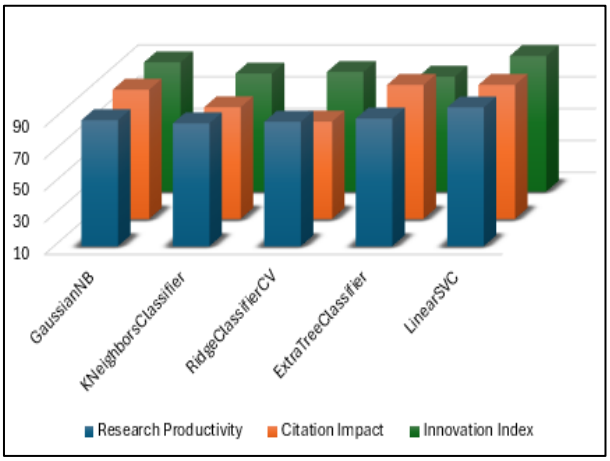


Figure 10: Analysis of results in term of accuracies

In conclusion, the findings show that there is 95% average accuracy for predicting the research outcomes from the policy variables using LinearSVC and ExtraTreeClassifier models, and overall, with single model accuracy among three categories LinearSVC achieves the highest accuracy of 97%. Other relatively basic models that have showed usefulness include GaussianNB, but these are progressively being dominated by more complex models in cases where complex data are needed, in this case to predict the effect of research policies, more complex models demonstrated better performance. The Receiver Operating Characteristic (ROC) curves for the three outcome variables demonstrate the effectiveness models in classifying these key research metrics. For Research Productivity as shown in Fig 11, the ROC curve shows that LinearSVC outperforms all other models, achieving the highest AUC of 0.62, indicating its robustness in capturing productivity patterns. GaussianNB follows closely with an AUC of 0.83, showcasing its simplicity yet effective classification capabilities. However, models like KNeighborsClassifier and ExtraTreeClassifier show lower AUC scores of 0.50 and 0.65, respectively, highlighting potential challenges in handling this outcome's complexity.

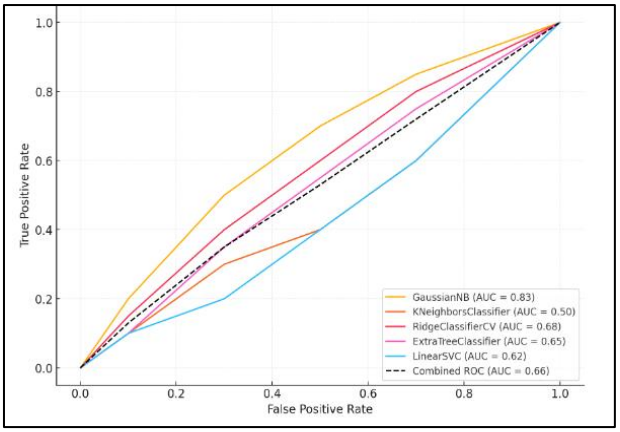


Figure 11: Analysis of ROC-AUC combined model outcomes using research productivity metrics

For Innovation Index as shown in fig 13, the ROC curve reveals that LinearSVC achieves an exceptional AUC of 0.96, making it the most effective model for this outcome. GaussianNB and RidgeClassifierCV also perform well, with AUCs of 0.92 and 0.88, respectively. In the case of Citation Impact as shown in fig 12, LinearSVC again leads with an AUC of 0.93, followed by GaussianNB and ExtraTreeClassifier with AUCs of 0.89 and 0.87, respectively. These results reflect the models' ability to effectively classify citation influence levels. Meanwhile, KNeighborsClassifier and RidgeClassifierCV, with AUCs of 0.67 and 0.61, show relatively weaker performance, indicating a need for improvement in handling this dataset's intricacies. However, KNeighborsClassifier and ExtraTreeClassifier, with AUCs of 0.84 and 0.81, exhibit moderate effectiveness in distinguishing innovation levels. These findings highlight the value of advanced predictive tools in improving research policy framework and evaluation to bridge the need to select an appropriate training model to get useful and reliable information.

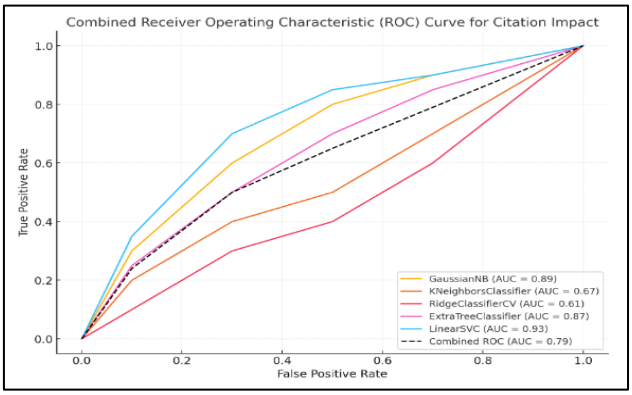


Figure 12: Analysis of ROC-AUC combined model outcomes using citation impact metrics

In the proposed study, use of SVC reached overall accuracy of 97%, which is still a major leap from previous studies, as comparative analysis shown in table VIII. For example, [11], used three models LR, SVC, and DT, and obtained accuracy was 80%. Similarly, [13] used XGBoost (XGB), Random Forest (RF), SVM classifiers, with higher improvement of the accuracies by 85%.

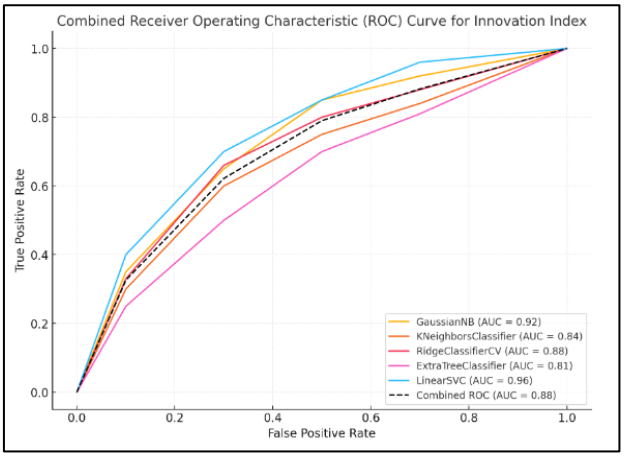


Figure 13: Analysis of ROC-AUC combined model outcomes using innovation index metrics

Overall, AUC-ROC values show the performance of the different classifiers in predicting the research productivity, citation impact and radicalness of innovation. Compared to other models, models with LinearSVC always serve as the best performing model as evidenced by the best AUC scores 0.93 in citation impact and 0.96 on innovation index. As LinearSVC performs so well, we might assume that the decision boundary found by the model was so good for the dataset structure as it is effective at dealing with high dimensional feature spaces. Contrarily, models such as KNeighborsClassifier and RidgeClassifierCV achieve lower AUC values especially in research productivity (0.50 and 0.68, respectively), due to their lack of a possibility to fit the complex patterns of data. GaussianNB performs reasonably well on all metrics (AUC = 0.89 for citation impact and 0.92 for innovation index), probably because it handles categorical and continuous variable in a probabilistic manner. Model performance varies depending on feature engineering, hyperparameter tuning, dataset characteristics and this points to the importance of them in being able to classify. However, [15] obtained only 79% accuracy by applying DT and Naive Bayes (NB) Technique. The increased performance improvement for our proposed approach underlines the advantage of our model and feature selection method as a new baseline for measuring the flow-through effect of research policies by employing machine learning.

Table 6: Comparison with existing studies

Sr. No	Ref	Top Model	Results
1	[11]	LR	80%
2	[13]	RF	85%
3	[15]	DT	79%
4	[18]	KNN	89%
5	**Proposed**	**SVC**	**97%**

7 Conclusion and future work

The evaluation of research policies and their impact on various outcome variables based on synthetic dataset, defining a critical evidence-based decision-making in field of academia and innovation ecosystems. By leveraging ML approaches and feature engineering techniques, this study explores various policies and contextual factors in shaping research success. Our findings reveal that ML model such as LinearSVC as the top-performing classifier across all research outcome variables, with the highest accuracy of 97% for Research Productivity. This demonstrates the model's robustness in handling diverse research policy datasets and its suitability for predictive analysis. Among the features, Funding Amount emerged as the most influential policy variable with the highest IG (0.089) and GR (0.134), underscoring the significant role of financial support for acquiring successful research outcomes. This study bridges the gap in understanding how policy design impacts research outcomes by providing a systematic, data-driven framework. Moving forward, additional

features such as technology adoption, and societal impact metrics could provide a comprehensive view of research outcomes using advanced techniques like Explainable AI to enhance the interpretability of models and ensure more transparency in policymaking, offering deeper insights into sustainable policy design.

References

[1] L. Vesnic-Alujevic, S. Nascimento, and A. Pólvora, "Societal and ethical impacts of artificial intelligence: Critical notes on European policy frameworks," *Telecommunications Policy*, vol. 44, no. 6, p. 101961, Jul. 2020, doi: https://doi.org/10.1016/j.telpol.2020.101961.

[2] Y. K. Dwivedi *et al.*, "'So what if ChatGPT wrote it?' Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy," *International Journal of Information Management*, vol. 71, no. 0268-4012, p. 102642, Aug. 2023, doi: https://doi.org/10.1016/j.ijinfomgt.2023.102642.

[3] O. V. Spivakovsky, S. A. Omelchuk, V. V. Kobets, N. V. Valko, And D. S. Malchykova, "Institutional Policies On Artificial Intelligence In University Learning, Teaching And Research," Information Technologies and Learning Tools, vol. 97, no. 5, pp. 181–202, Oct. 2023, doi: 10.33407/itlt.v97i5.5395.

[4] E. Lett, E. Asabor, S. Beltrán, A. M. Cannon, and O. A. Arah, "Conceptualizing, Contextualizing, and Operationalizing Race in Quantitative Health Sciences Research," *The Annals of Family Medicine*, vol. 20, no. 2, pp. 157–163, Jan. 2022, doi: https://doi.org/10.1370/afm.2792.

[5] G. MacIntyre, N. Cogan, A. Stewart, N. Quinn, M. O'Connell, and M. Rowe, "Citizens defining citizenship: A model grounded in lived experience and its implications for research, policy and practice," *Health & Social Care in the Community*, vol. 30, no. 3, Jun. 2021, doi: https://doi.org/10.1111/hsc.13440.

[6] P. M. Krafft, M. Young, M. Katell, K. Huang, and G. Bugingo, "Defining AI in Policy versus Practice," in Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, in AIES '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 72–78. doi: 10.1145/3375627.3375835.

[7] M. K. Hasan Chy and O. Nana Buadi, "Role of Machine Learning in Policy Making and Evaluation," International Journal of Innovative Science and Research Technology (IJISRT), pp. 456–463, Oct. 2024, doi: 10.38124/ijisrt/IJISRT24OCT687.

[8] K. Amarasinghe, K. T. Rofdolfa, H. Lamba, and R. Ghani, "Explainable machine learning for public policy: Use cases, gaps, and research directions," Data Policy, vol. 5, Feb. 2023, doi: 10.1017/dap.2023.2.

[9] R. Wang, J. Li, W. Shi, and X. Li, "Application of Artificial Intelligence Techniques in Operating Mode of Professors' Academic Governance in American Research Universities," Wirel Commun Mob Comput, vol. 2021, 2021, doi: 10.1155/2021/3415125.

[10] B. D'Ippolito and C.-C. Rüling, "Research collaboration in Large Scale Research Infrastructures: Collaboration types and policy implications," Res Policy, vol. 48, no. 5, pp. 1282–1296, 2019, doi: https://doi.org/10.1016/j.respol.2019.01.011.

[11] B. Álvarez-Bornstein and M. Bordons, "Is funding related to higher research impact? Exploring its relationship and the mediating role of collaboration in several disciplines," J Informetr, vol. 15, no. 1, p. 101102, 2021, doi: https://doi.org/10.1016/j.joi.2020.101102.

[12] B. Ibrahim, "Arab Spring's effect on scientific productivity and research performance in Arab countries," Scientometrics, vol. 117, no. 3, pp. 1555–1586, 2018, doi: 10.1007/s11192-018-2935-z.

[13] S. Kumar, D. Sharma, S. Rao, W. M. Lim, and S. K. Mangla, "Past, present, and future of sustainable finance: insights from big data analytics through machine learning of scholarly research," Ann Oper Res, 2022, doi: 10.1007/s10479-021-04410-8.

[14] A. Ebadi, S. Tremblay, C. Goutte, and A. Schiffauerova, "Application of machine learning techniques to assess the trends and alignment of the funded research output," J Informetr, vol. 14, no. 2, p. 101018, 2020, doi: https://doi.org/10.1016/j.joi.2020.101018.

[15] J. A. V Tohalino and D. R. Amancio, "On predicting research grants productivity via machine learning," J Informetr, vol. 16, no. 2, p. 101260, 2022, doi: https://doi.org/10.1016/j.joi.2022.101260.

[16] M. Umer *et al.*, "Scientific papers citation analysis using textual features and SMOTE resampling techniques," *Pattern Recognition Letters*, vol. 150, pp. 250–257, Oct. 2021, doi: https://doi.org/10.1016/j.patrec.2021.07.009.

[17] Lyutov, A., Uygun, Y. & Hütt, MT. Machine learning misclassification networks reveal a citation advantage of interdisciplinary publications only in high-impact journals. *Sci Rep* **14**, 21906 (2024). https://doi.org/10.1038/s41598-024-72364-5

[18] Fernandes, C., Ferreira, J.J., Raposo, M.L. *et al.* The dynamic capabilities perspective of strategic management: a co-citation analysis. *Scientometrics* **112**, 529–555 (2017). https://doi.org/10.1007/s11192-017-2397-8

[19] F. K. Alarfaj, H. Ahmad, H. U. Khan, A. M. Alomair, N. Almusallam, and M. Ahmed, "Twitter Bot Detection Using Diverse Content Features and Applying Machine Learning Algorithms," Sustainability (Switzerland), vol. 15, no. 8, Apr. 2023, doi: 10.3390/su15086662.

[20] M. Ahmed, H. U. Khan, S. Iqbal, and Q. Althebyan, "Automated Question Answering based on Improved TF-IDF and Cosine Similarity," in 2022 Ninth International Conference on Social Networks Analysis, Management and Security (SNAMS), 2022, pp. 1–6. doi: 10.1109/SNAMS58071.2022.10062839

Text Segmentation in Online Health Consultation Using Multi Layer Perceptron with Sentence Embedding and Sentence Features

Yunianita Rahmawati, Daniel Siahaan*, Diana Purwitasari
Department of Informatics, Faculty of Intelligent Electrical and Informatics Technology, Institut Teknologi Sepuluh Nopember (ITS), Surabaya 60111, Indonesia
E-mail: 7025211024@student.its.ac.id, daniel@if.its.ac.id, diana@its.ac.id
*Corresponding author

Keywords: aspects of doctor-patient communication, OHC, MLPSentFeat, sentence feature, text segmentation

Received: September 3, 2025

The six aspects of doctor-patient communication are universal and relevant to various medical contexts and are the basis for structuring and analyzing doctors' answer texts in online health care. Identifying each aspect of communication in the doctor's answers is important to ensure effective communication. There are three critical challenges in segmenting the doctor's answers, i.e. limited public dataset, extreme imbalance data, and implicit semantic variations between aspects. This study proposed a novel approach to text segmentation, focusing on communication aspects, particularly doctor-patient interactions. Unlike classical segmentation methods, which relied on topic similarity, the proposed method segmented text based on the communicative function of each sentence, offering a more accurate reflection of the interaction structure in Online Health Consultation (OHC). The model developed, MLPSentFeat, integrated sentence features directly into its architecture and demonstrated effectiveness in handling data with highly imbalanced label distributions without the need for synthetic data or complex balancing techniques.

Sentence features were formed using a combined approach: the Likelihood Ratio (LR) algorithm filtered words relevant to a label, and Information Gain (InfoGain) selected the most informative features. Experimental results showed that integrating these sentence features significantly enhanced the model's sensitivity to variations in linguistic structure, especially in recognizing non-standard sentence types, such as questions, which were prevalent in the information-gathering aspect of medical communication.

The best model produced, MLPSentFeat+DS2+Features3, was optimized to achieve the lowest segment error percentage of 8.18%, despite a slight performance decline in some labels after optimization. The use of text normalization, along with appropriate data size, cleanliness, and alignment of sentence features with the sentences' semantic structure, proved crucial in preventing overfitting. The MLPSentFeat model was successfully applied across various domains, demonstrating cross-domain adaptability, including in doctor's answer in online medical consultation systems, with potential for further development to identify questions with diverse sentence structures.

Povzetek: Študija predstavlja nov pristop k segmentaciji besedil zdravnikovih odgovorov v spletnih zdravstvenih posvetih, ki temelji na komunikacijski funkciji stavkov in z modelom MLPSentFeat učinkovito obravnava neuravnotežene podatke ter izboljšuje prepoznavanje različnih vidikov komunikacije zdravnik–pacient.

1 Introduction

The public increasingly used online health consultation (OHC) services as a source of fast and practical health information [1][2]. The main attraction lay in the interaction between the doctor and the patient, primarily through the patient's questions and answers. The consultation text included the user's question, the doctor's answer, the consultation title, the doctor's name, and the time. The OHC service classified these consultation texts based on disease topics. Doctors' answers were prepared based on personal preferences, intuition, and the context of the questions received, and without standard structural guidance, giving rise to variations in content and appearance (layout). Doctors' answers were not structured in terms of text content; for example, they repeated discussions of a topic in different parts or conveyed topics that were not always relevant to the patient's questions and disease topics [3]. Sometimes, the medical communication delivered was also less effective [4]. Irregularities in layout also occurred, such as using paragraph breaks unrelated to a particular topic or combining several topics in one paragraph, which caused the main explanation to lack focus. OHC doctors' answers tended to be short, with an average of only 20 to 25 sentences [5], causing unclear topic boundaries, which indicated weaknesses in the text structure [6]. For communication to be more effective, the doctor's answers needed to be structured following aspects of doctor-patient communication that had been proven to support the

therapeutic relationship and increase patient satisfaction and compliance [7][8]. Furthermore, to ensure that the doctor's responses were well-structured according to the existing layout, these responses should have been divided into smaller segments. This segmentation aimed to make the text more readable and comprehensible [6], as well as to enable readers to more easily locate relevant topics without having to read the entire text [9]. Segmentation of doctors' answers was carried out at the sentence level because the text length was relatively short, so that topic changes could occur at the sentence level. Acceptance of text-based communication differed from oral communication, so users might have experienced difficulties in interpreting written text that was not well structured [10].

This research used the six aspects of doctor-patient communication formulated by Ann King [11]: fostering the relationship, gathering information, providing information, making decisions, responding to emotions, and enabling disease- and treatment-related behavior. These six aspects were universal and relevant to various medical contexts, thus serving as the basis for structuring and analyzing doctors' answer texts in OHC. Identifying each aspect of communication in the doctor's answer was important to ensure effective communication, where patients felt respected and understood [12], and played an essential role in health services [13][14]. The information-gathering aspect helped patients reflect on their medical problems, even though doctors did not always receive an immediate response. One form of reflection was asking questions so the patient could understand their condition more deeply [15]. Providing information was related to patient satisfaction with communication, which increased when doctors were willing to discuss the patient's medical condition openly [16]. Responding to emotions allowed doctors to convey information with empathy and provide hope [16]. Fostering the relationship built trust between patients and doctors [16][17]. Decision-making enabled doctors to recommend medical treatment to patients, especially in cases requiring immediate care [18]. The enabling disease- and treatment-related behavior aspect provided practical understanding to patients regarding how to alleviate or cure the disease they were experiencing [19]. Even though not all aspects appeared equally in every doctor's answer, this research still identified all of these aspects to evaluate the quality of communication in OHC.

This research faced several challenges in segmenting doctors' answer texts. First, although the number of labeled sentences in the dataset was relatively large, the total amount of medical consultation data available in each category was relatively limited. This limitation in the amount of medical consultation data resulted in a restricted variation in the information that could be learned by the model. Doctors' answer responses tended to follow similar patterns, which restricted the diversity of information in the data, causing the model to focus on highly similar patterns within the response texts. Therefore, despite the large number of labeled sentences, the limited variation in the medical consultation data became the primary factor leading to overfitting in the classification model. Second, the distribution of doctors' answer sentences was highly unbalanced, where aspects such as gathering information and responding to emotion (especially expressions of sympathy and empathy) were minimal, while the providing information aspect dominated. The OHC service did not provide an interface for users to answer follow-up questions from doctors in consultation sessions, so doctors limited the collection of information from users. In addition, doctors rarely expressed sympathy and empathy due to their difficulty in dealing with uncertainty in understanding patients' emotions and expressing empathy [10]. Third, this research formulated text segmentation targeting six aspects, each of which reflected an aspect of doctor-patient communication. In addition, existing text segmentation methods generally did not provide semantic labels, so the coverage of communication aspects in each segment was unclear [3][20][21]. Fourth, the structure of neural network models, such as MLP, had several hidden layers and neurons that needed to be optimized using a metaheuristic optimization algorithm so that classification performance improved [22][23][24][25]. Variations in wording and sentence forms in doctors' answer texts, as well as similarities between classes, reduced the effectiveness of simple models. Therefore, proper parameter tuning became crucial to improve the accuracy, generalization, and efficiency of model training.

In classical segmentation, one segment contained sentences whose topics were consistent. However, in segmentation based on communication aspects, the similarity of segments was determined not by the topic, but by the communicative function of the sentence. Therefore, even though the topic changed (for example, from 'itch' to 'flu'), if both sentences functioned as aspects of providing information, they were still considered to be in one segment.

The models that addressed the text segmentation task consisted of unsupervised and supervised learning. Unsupervised models included clustering, similarity-based, and probabilistic generative models [26]. Text segmentation used probabilistic generative models such as topic modeling [20][27][28][29][30], which allowed the identification of "hot topics" in documents [27]. This study formed segments based on doctor-patient communication aspects, which were not discussed explicitly in the doctor's answers. Text segmentation could also be performed using clustering models, in which segments or segment topics were formed based on the words discussed in the text [21][31][32][3]. However, since the six aspects of doctor–patient communication were not explicitly addressed in the doctors' responses, clustering models were less suitable for segmenting the text based on these aspects. Text segmentation using similarity-based methods such as TextTiling [33], C99 [34], GraphSeg [35][36][3], Content Vector Segmentation (CVS) [37][38][3], allowed the formation of segments by measuring the similarity between text units and was applied to long data [33]. The text content of a doctor's answer in a communication aspect could cover various topics, and differences could be observed between sentences. In addition, doctors' response texts tended to be

shorter, consisting of only 20 to 25 sentences [5] or one or two paragraphs.

Text segmentation used supervised learning based on certain aspects, for example, the sentiment analysis aspect with SEGBOT [26] and the doctor-patient communication aspect using the Hierarchical Temporal Convolutional Network (HTCN) [39]. The SEGBOT model was not designed to segment based on specific aspects of communication, such as doctor-patient communication, but rather to detect natural boundaries in text discourse structures based on semantic transitions between sentences. The HTCN model allowed for the segmentation of doctors' answer topics based on communication aspects and was applied to perform sentence-level segmentation. However, each aspect required many sentences. In addition, Wiguna's research only discussed sentence classification based on communication aspects but did not regroup dialogue into complete segments based on communication flow. This research shared a similarity with Wiguna's research in that it segmented the text of doctors' answers based on communication aspects. At the same time, the difference lay in the imbalance of sentence distribution, especially in the aspects of gathering information and responding to emotions (specifically sympathy and empathy). This difference represented an analysis gap that could be exploited to improve segmentation performance in this research.

The solution proposed in this research was to develop a text segmentation model using MLPSentFeat. This research aimed to form small segments of doctors' answers in OHC services and provide segment information. This model was designed through four main stages. First, at the pseudo-labeling stage, the training data were expanded by applying pseudo-labeling techniques based on voting classifiers from various machine learning algorithms and using the k-fold cross-validation algorithm to divide the training and testing data. Second, in the classification stage, MLPSentFeat was applied to the doctor's answers at the sentence level. Third, in the text segmentation stage, segments of doctors' answers were formed based on the classification results, labels, and annotations. Fourth, in the MLP optimization stage, the best MLP configuration was obtained, including the optimal number of hidden layers and neurons. This process was conducted by comparing various metaheuristic optimization algorithms and adjusting the MLP parameters: hidden layers, neuron range, population size, and repetition size. The best optimization model was then applied to various datasets.

The novelty of this study encompassed several methodological and empirical contributions. First, this research introduced a text segmentation approach based on specific communicative aspects—particularly those found in doctor–patient interactions—where segment similarity was determined not by topical relevance, but by the communicative function of each sentence. This departed from traditional segmentation methods, which typically constructed segments based on thematic coherence. Second, the MLPSentFeat model demonstrated the capability to classify data with highly imbalanced label distributions, where certain labels contained only a few dozen samples while others contained thousands, without the need to augment the data with synthetic (dummy) samples.

Moreover, the MLPSentFeat model operated efficiently and did not require high computational resources. Third, the model offered a novel architectural combination by integrating an MLP with explicitly constructed sentence features, enabling it to capture sentence patterns from labels with limited training instances. Fourth, sentence feature construction was achieved through a hybrid algorithm that combined LR and InfoGain, where LR filtered label-specific relevant words and InfoGain selected the most informative features. Fifth, the explicit integration of sentence features into the MLPSentFeat architecture was shown to improve the model's sensitivity to linguistic structural variation, particularly in recognizing non-standard sentence forms such as interrogative expressions, which were dominant in the gathering information aspect of doctor–patient communication. Sixth, as a minor novelty, this study also identified sentence forms in doctors' answers to selectively exclude certain words from the stopword list, thereby treating them as important features during the preprocessing stage, particularly in the stopword removal process.

This research focused on segmenting doctors' answers in OHC services based on aspects of doctor-patient communication. This segmentation not only contributed to improving users' understanding and access to information but also could be used by NLP developers for chatbot development, by OHC services to standardize answers, and by doctors to improve the quality of interactions with patients. This research contributed as follows:

- This research expanded the training data with pseudo-labeling using a voting classifier from several machine learning methods.
- This research proposed a text segmentation model using an MLP-based sentence embedding and sentence features for segmenting doctors' answers and annotating segments based on aspects of doctor-patient communication.
- The authors optimized the model by tuning its parameters to determine the optimal number of hidden layers and neurons for the MLP method after conducting an experiment using the initial parameters.

The structure of the remaining sections was organized as follows: Section II described relevant scientific work. Section III outlined the methodology used. Section IV presented the results and discussion. Section V summarized the conclusions of the research.

2 Related work

The related work provided a brief review of the research that was used as a reference in this study. This short review was divided into several subchapters, including the application of sentence forms in doctors' answers, the identification of aspects of doctor-patient communication, text segmentation, pseudo-labeling methods, and optimization methods.

Table 1: The form of the doctor's answer sentence is based on the doctor-patient communication.

Communication Aspect	**Sentence Form**	**Example of a doctor's answers sentences**	**Role, Responsibilities, and Abilities of Doctors**
Fostering the relationship (F1)	Phatic sentence [119]	'*Halo* (Hello)'. '*Sekian informasi dari saya* (That was the information from me)'.	The doctor greeted the patient and closed the consultation.
		'*Mohon maaf atas keterlambatan respon kami* (We apologized for our late response)'	The doctor admitted the error and expressed regret.
		'*Pertama-tama, saya hendak mengonfirmasi terlebih dahulu* (First of all, I wanted to confirm first)'	Doctors developed open communication skills.
		'*Jika sulit menghubungi dokter, silakan gunakan layanan chat di Play Store* (If it was difficult to contact a doctor directly, use the chat service in the Play Store)'	Doctors were involved in building partnerships.
Gathering information (F2)	Ordinary interrogative sentences [120][121]	'***Apakah** anda tidak mengalami menstruasi selama 3 bulan **?*** (Had you not had your period for 3 months?)'	Doctors asked open-ended questions to patients.
Providing information (F3)	Declarative sentences [122]	'*Bila hasilnya positif, maka keterlambatan menstruasi anda disebabkan oleh kehamilan* (If the result was positive, then your delayed menstruation was caused by pregnancy)'	Doctors shared patient information.
	Rhetorical interrogative sentences [120]	'*Baca laman artikel ini mengenai **Kenapa Bisa Terjadi Sakit Perut Sebelah Kanan ?*** (Read this article about **Why Can Right Side Abdominal Pain Occur ?**)'	
	Incomplete sentence [123]	'*Infeksi saluran kemih* (Urinary tract infection)'	
Decision-making (F4)	Command sentence (imperative form) [121]	'*Anda harus melakukan pemeriksaan ke dokter kandungan* (You had to have a check-up with a gynecologist)'	Doctors created collaborative action plans.
	Negative imperative sentences [124]	'*Bila keluhan berlanjut **jangan** ragu memeriksakan diri ke dokter* (If the complaint persisted, **do not** hesitate to see a doctor))'	
	Rhetorical interrogative sentences [120]	'*Dokter akan mengetahui **apakah** anda menderita penyakit ini berdasarkan hasil pemeriksaan* (The doctor would know whether you had this disease based on examination results)'	
	Declarative sentences [122]	'*Anda dapat menggunakan asuransi umum sehingga meringankan biaya prosedur tersebut* (You could use general insurance that covered the cost of the procedure)'	Doctors agreed with patients.
Responding to emotions (F5)	Empathy-expression sentence [125]	'*Saya mengerti kekhawatiran yang Anda rasakan* (I understood the concern you had. You were not alone. We felt the same)'	Doctors expressed empathy toward patients.
	Sympathy-expression sentence [126]	'*Sebelumnya kami turut prihatin terhadap kondisi yang menimpa ibu Anda* (We expressed our condolences regarding the condition that befell your mother)'	Doctors expressed sympathy for patients.
	Wish-expression sentence [121]	'***Semoga** ibu Anda lekas sembuh* (I hoped you got better soon)'	Doctors provided emotional support.
Enabling disease- and treatment-related behavior (F6)	Command sentence [121]	'*Perbanyak minum air untuk mencegah dehidrasi* (Drink more water to have prevented dehydration)'	Doctors changed patients' health behaviors or provided treatment-related advice.
	Negative imperative sentences [121]	'***Jangan** berikan jus dan minuman bersoda* ((**Don't** give juice and fizzy drinks)'	
	Rhetorical interrogative sentences [120]	'*Amati **apakah** diare tetap berlangsung lebih dari 1 minggu* (Observe whether diarrhea persisted for more than 1 week)'	

Table 2: Review of the methodology for identifying aspects of doctor-patient communication regarding the doctor's answers.

Study	Model	Embedding	Dataset	Research Gap	Contribution
Rahmawati et al. [5]	Multiclass classification with one-hot encoding. Compares LR, MLP, GPT-2, GRUs, FNN, LSTM, RNN, CNN, and BiLSTM.	Sentence embedding (SBERT)	Doctor's answers from the Alodokter website were from 502 data points with 11223 sentences. It was in Indonesian.	Used a small amount of data. Sentence features were extracted by selecting words with high frequency in each class.	Proposed MLPSentFeat. Alodokter dataset (13799 data). Sentence features were extracted from the keywords in each communication aspects. Feature reductions.
Zhafirah et al. [36]	Semantic Relatedness Graph (SRG).	Word embedding (IndoBERT)	Doctor's answers from the Alodokter website was 15,000. It was in Indonesian.	Segmentation results were not as good.	Segmenting doctor's answers using classification techniques.
Wiguna et al. [39]	HTCN, BiLSTM, LSTM, and CNN.	Word embedding (Word2Vec, FastText, GloVe)	The number of doctor's answers from the SehatQ website (KlikDokter, Alodokter) was 15,000. It was in Indonesian.	It had not been tested using sentence embedding. The amount of labeled data is small.	Using Sbert. Added 2000 labeled data from 42,226 sentences.
Rahmawati et al. [3]	Modeling of LDA topics, clustering (kMeans), and text segmentation (Graphseg, CVS).	Word embedding (TF-IDF)	Doctor's answers from the Alodokter website includes 500 data points with 8576 sentences. It was in Indonesian.	Used a small amount of data and had poor segmentation.	Use classification techniques based on aspects of doctor–patient communication to form segment and annotations.
Juanita [87]	Multi-label Relevance and Label Powerset classification by RF, Adaboost, KNN, and MLP.	Word embedding (Word2Vec, TF-IDF)	Doctor's answers from the Alodokter website includes 500 data points, and Steadyhealth includes 500 data points. They were in Indonesian and English.	Used a small amount of data. It had not been tested using sentence embedding.	Added labeled data by 2000 with 42,226 sentences. Using SBERT.

2.1. Use of sentence forms in doctors' answers

This section described the sentence forms in the doctor's answer for each aspect of doctor-patient communication, which were used as the basis in designing the model, such as preprocessing settings and sentence features. Identifying the doctor's answer sentence for the aspect of doctor-patient communication referred to the doctor's role, responsibility, and ability in each communication aspect [11]. The sentence form referred to the types of sentences used in each aspect of communication. The results of identifying the form of the

Placed the question word in the middle of the sentence and could end with or without a question mark. Declarative sentences conveyed statements from the doctor, while incomplete sentences represented structurally incomplete sentence forms. Command sentences were used to direct patients to perform certain actions, whereas negative imperative sentences were intended to prevent specific actions, typically indicated by the use of the word '*jangan* (do not)'. In addition, doctors

Doctor's answer sentence based on the elements of doctor-patient communication were shown in Table 1. Doctor-patient communication used different sentence forms, adjusted to the doctor's roles, responsibilities, and abilities. In addition, the writing style of doctors in the OHC service tended to use the same sentence form when explaining information, namely declarative sentences and rhetorical interrogatives.

Phatic sentences were used to maintain social relations between doctors and patients, as illustrated in Table 1. Ordinary interrogative sentences were written with a question word at the beginning and ended with a question mark—for example, starting with '*apakah* (had)' and ending with '?'. Rhetorical interrogative sentences also expressed sympathy, empathy, and prayers for the patient.

Some sentence forms were only used for one aspect of communication. Here are some examples. Phatic sentence forms (sentences used to build communication) were only used in the "fostering the relationship" aspect because doctors needed to develop relationships with patients. Sympathy, empathy, and wish sentence forms (sentences expressing hope or prayer) were only used in the "responding to emotion" aspect because doctors

needed to acknowledge the patient's emotions (sympathy), understand the patient's feelings (empathy), and give hope for healing or pray for the patient (wish). Incomplete sentences were only used in the "providing information" aspect, which meant that only this aspect contained such sentence forms.

Other sentence forms were used in several aspects of communication. 1) Various forms of interrogative sentences with various uses in the aspects of gathering information, providing information, decision-making, and enabling disease- and treatment-related behavior. This sentence structure was frequently used in various aspects of communication, indicating that physicians often used interrogative sentences. Question sentences were used to obtain answers or confirm previously known information. Interrogative sentences generally required a question word or a question mark (?), although the mark might not have been used in some types of interrogative sentences. Interrogative sentences in the aspect of gathering information were used to obtain information from patients, while in other aspects were used to confirm information. 2) Various forms of command sentences were used in the aspects of decision-making and enabling disease- and treatment-related behavior. Doctors instructed patients to take action related to treatment in both aspects. 3) News sentences were used to provide information and make decisions. Doctors provided health information in the form of news sentences in both aspects.

Doctors provided general information in the aspect of providing information, while doctors conveyed information about medical procedures in the decision-making aspect.

2.2. Identify aspects of doctor-patient communication

Research on identifying aspects of communication between doctors and patients from the doctor's answers (Table 2) had evolved through various approaches, including text classification and segmentation methods. Multiple models were explored, ranging from multilabel and multiclass classification to machine learning and deep learning techniques. These methods had been extensively applied to enhance text analysis in health communication. Additionally, while most previous studies employed word embedding, a few utilized sentences embedding. The datasets used in several prior studies originated from the Alodokter platform, which provided a relatively small volume of data.

2.3. Text Segmentation method

Text segmentation referred to the process of dividing documents into smaller units, known as segments. These segments could consist of topics, words, sentences, lines, passages, or characters. The type of segment was typically determined by the analysis goal and target specification. Document segmentation supported text analysis by offering: 1) reduced and more coherent data chunks than complete documents, and 2) well-defined units for analysis and data access [40]. Text segmentation was commonly applied for purposes such as: 1) topic segmentation, which extracted topically coherent segments, and 2) discourse segmentation, which extracted Elementary Discourse Units (EDUs) and other discourse structures—covering both topical and linear segmentation [41]. Segment boundaries were typically marked by paragraph breaks [33].

Previous studies performing aspect-based segmentation included Wiguna, who segmented doctor's answers based on aspects of doctor–patient communication using the HTCN method. The HTCN method segmented texts by leveraging dilated causal convolution, allowing the model to capture temporal sentence relationships accurately, maintain sequential dependencies, and manage long-range dependencies without gradient vanishing issues. Li addressed segmentation of topics and discourse in sentiment analysis of film reviews using the SEGBOT method. SEGBOT used BiRNN for contextual understanding, RNN for decoding, and a pointer network to determine segment boundaries based on end-position probabilities. Zhu [42] applied aspect-based segmentation to Chinese restaurant reviews using the PRanking method, where segmentation occurred by scoring texts according to aspects and separating segments at score transitions. Hananto [20] conducted segmentation of online customer reviews using TopicDiff-LDA for automatic annotation based on tourism aspects. This method detected topic distribution changes using LDA to determine segment boundaries. These prior studies served as references for aspect-based segmentation research.

Research on segmentation in the health domain included Chowdhury [43], who developed the MedTextSeg algorithm for segmenting sections in medical reports. Eisenstein [44] proposed a Bayesian algorithm based on lexical cohesion, which did not require labeled data, and applied it to medical textbooks. Ganesan [45] segmented clinical texts by identifying headers, footers, and key sections such as allergies or chief complaints using Logistic Regression (LR). Apostolova [46] employed SVM to automatically classify report content into semantic sections. Edinger [47] implemented rule-based approaches to search specific segments in clinical texts, with compatibility for engines like Lucene and Essie. Sadoughi [48] utilized LSTM for detecting section boundaries in clinical dictations. Ginter [49] combined HMM and LSA to segment and annotate electronic patient records. These efforts provided a foundation for health-related text segmentation studies.

Applications of text segmentation spanned several domains, such as chatbot Q&A in manufacturing consulting services [50], prediction of workplace accident severity [51], and automated assessment in Chinese medical education [52]. Other applications included analysis of "infarction" in Japanese medical literature [53], construction of Chinese EMR dictionaries [54], model extraction from knowledge graphs for manufacturing planning [55], self-learning evaluation in science education [56], structured extraction of medical documentation [57], negation detection in French clinical records [58], comparison of medical corpora [59], and scene segmentation in narrative fiction [60].

2.4. Pseudo-labeling method

This study employed a classification model to automatically label unlabeled data. Pseudo-labeling used a voting classifier algorithm derived from multiple classification models, namely, LR, SVM, Decision Tree (DT), K-Nearest Neighbor (KNN), Random Forest (RF), Naïve Bayes (NB), Gradient Boosting Machine (GBM), Extreme Gradient Boosting (XGBoost), Light GBM (LGBM), and Categorical Gradient Boosting (CatBoost).

The architectures of XGBoost, CatBoost, DT, LR, and RF referred to Abbas's study [61], which proposed the EnsCL-CatBoost model to improve accuracy in software requirements classification using unlabeled data. The EnsCL-CatBoost model was compared with XGBoost, CatBoost, DT, LR, and RF, and yielded the best results. XGBoost was an efficient and fast model for building gradient-boosted decision trees using parallel tree boosting techniques, which improved accuracy in regression, classification, and ranking tasks. CatBoost handled categorical features without One-Hot Encoding, managed missing data, reduced overfitting, and achieved high prediction accuracy with a fast-training process.

DT handled nonlinear relationships, was easy to interpret, and was effective for feature selection without requiring data distribution assumptions. LR performed well in classifying software requirements by handling binary and multi-class problems and providing measurable prediction probabilities. RF was suitable for handling complex software data, preventing overfitting, and improving accuracy through bagging techniques and out-of-bag error validation.

The architectures of LGBM, SVM, and NB referred to Alzamzami's research [62], which developed a sophisticated LGBM model for general sentiment analysis on informal short texts. It comprehensively evaluated various classification algorithms and proposed features. LGBM demonstrated the ability to manage high-dimensional data and address data imbalance issues. NB served as an effective probabilistic classifier, particularly in word frequency-based sentiment analysis. SVM was effective in text classification by selecting an optimal hyperplane that maximized the margin between classes. In multi-class scenarios, a one-versus-all approach with a linear kernel was employed to obtain the best results.

The GBM architecture referred to He [63], who conducted aspect-based sentiment analysis using meta-based self-training methods (MSM). The GBM-BERT method utilized unlabeled data. GBM enabled each part of the task-specific tower to filter irrelevant information and combine helpful information from other components, thereby improving the model's overall performance.

The KNN architecture was based on Li's study [64], which proposed the augmented retrieval K-nearest neighbors (KRA) method for a deep learning-based text classification model.

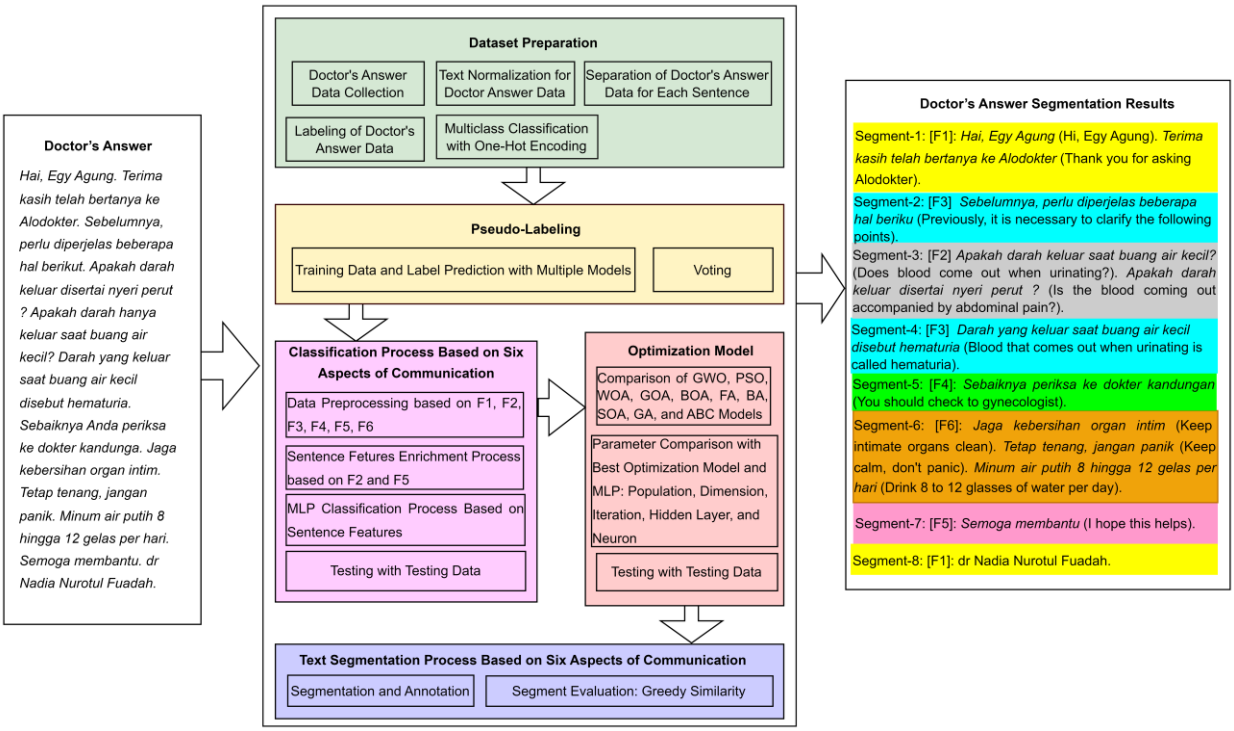


Figure 1: General architecture of MLPSentFeat method.

KNN leveraged data similarity to enhance prediction performance and helped language models understand context by referencing similar training examples.

These models provided multiple advantages, several of which had been applied to unlabeled and short text data. Therefore, the authors used these references as a foundation for the pseudo-labeling process on doctor's answer data, which typically consisted of longer sentences.

The classifier voting process referred to Elsaeed's research [65], which detected fake news on social media by combining the classification results of multiple models

using voting classifiers. Hard voting improved prediction accuracy by aggregating the outputs of several models and selecting the final label based on the majority vote per instance. The authors adopted this voting classifier approach to integrate the results from multiple models and determine the best label for each sentence. This ensemble learning strategy, also highlighted in Elsaeed's work, aimed to improve predictive performance. However, prior studies had not combined these classification methods for labeling unlabeled data, suggesting a novel contribution in up-to-date research modeling.

2.5. Optimization methods

Previous studies modified various optimization algorithms to optimize the weights and biases of MLP methods across different domains [66][67][23][68][69]. In addition, the number of hidden layers and neurons in MLP could be optimized using data mining approaches [70]. The references indicated that no prior study compared optimization algorithms specifically for tuning the number of hidden layers and neurons in the MLP method.

Studies also compared optimization algorithms in various domains [71][72][73][74][[75]. However, these studies did not explore optimization algorithms for MLP-based optimization.

This study performed optimization by tuning the number of hidden layers and neurons in the MLP model through the comparison of several metaheuristic algorithms—aimed at identifying optimal parameters through objective evaluation [76]. The compared algorithms included Grey Wolf Optimizer (GWO), Particle Swarm Optimization (PSO), Whale Optimization Algorithm (WOA), Grasshopper Optimization Algorithm (GOA), Butterfly Optimization Algorithm (BOA), Firefly Algorithm (FA), Bat Algorithm (BA), Seagull Optimization Algorithm (SOA), Genetic Algorithm (GA), and Artificial Bee Colony (ABC). The architectural foundation of each algorithm referred to previously established research.

The GWO architecture referred to [77], which mimicked the leadership hierarchy and hunting behavior of gray wolves (Canis lupus). PSO architecture referred to [78], which was inspired by the flocking behavior of birds and fish. The model integrated artificial life (A-life), group behavior theory, and evolutionary computing. WOA architecture, based on [79], simulated the bubble-net hunting strategies of humpback whales. GOA architecture, based on [80], mathematically modeled locust herd movement to solve optimization problems. BOA architecture referred to [81], which emulated butterflies' mating and foraging behavior using olfactory-based guidance for global optimization. FA architecture, based on [82], was inspired by the flickering patterns of fireflies and employed rules of attraction based on brightness. BA architecture, based on [83], was inspired by the echolocation behavior of microbats to detect prey and navigate in darkness. SOA architecture referred to [84], which simulated seagull migration and attack behaviors for improved search space exploration. GA architecture, based on [85], modeled natural evolution through selection, crossover, and mutation with solutions represented as binary gene chromosomes. ABC architecture referred to [86], which modeled the division of roles in artificial bee colonies, where each worker bee represented a nectar source.

3 MLPSentFeat on text segmentation

This section explains the MLPSentFeat model, which has five stages shown in Figure 1, i.e., doctor's answer dataset preparation, pseudo-labeling, classification, model optimization, and text segmentation. The pseudo-labeling stage is for labeling new data with several machine learning methods, the classification stage is for testing model performance on new data, the text segmentation stage is for forming segments from the classification results, and the MLP optimization model is for optimizing the number of neurons and hidden layers of the MLP. The input for the MLPSentFeat model is the doctor's answer data, while the output is the doctor's answer data, segmented based on aspects of doctor communication. In general, the flow of this research is as follows: first, the doctor's answer data is processed into the pseudo-labeling stage, where there is no testing of the pseudo-labeling results. Second, the pseudo-labeling results are processed into the classification stage. Third, the classification results are processed into the text segmentation stage. Fourth, the classification results can also be processed into the model optimization stage to obtain the optimal number of neurons and hidden layers. Then, the optimal number of neurons and hidden layers is applied to all the doctor's answer data through the classification stage. Next, it can be processed into the text segmentation stage again.

Some terms to pay attention to include sentence, data, and segment. Sentences consist of one or more words. The data consists of several sentences. For example, the input data for the doctor's answers in Figure 1 is referred to as 1 data. Segments are small groups that are formed from the doctor's answer data. One dataset can contain one or more segments. One segment can contain one or more sentences. These segments are formed based on six aspects of doctor-patient communication.

3.1. Doctor's answer dataset preparation

Data from the doctor's answers was prepared using several steps, shown in Figure 2. This section explains the preparation of the dataset, starting from data collection until the data is ready to be processed using a method. This data preparation has five processes: doctor's answer data collection, text normalization for doctor's answer data, separation of doctor's answer data for each sentence, labeling of doctor's answer data, and multiclass classification with one-hot encoding.

A. Dataset collection

This research uses three datasets of doctors' answers from various OHC platforms, namely dataset 1 (DS1), dataset 2 (DS2), and dataset 3 (DS3). DS1 was taken from the Alodokter platform, adopted data published by Juanita [87] spanning December 8, 2014, to February 28, 2021, can be accessed at

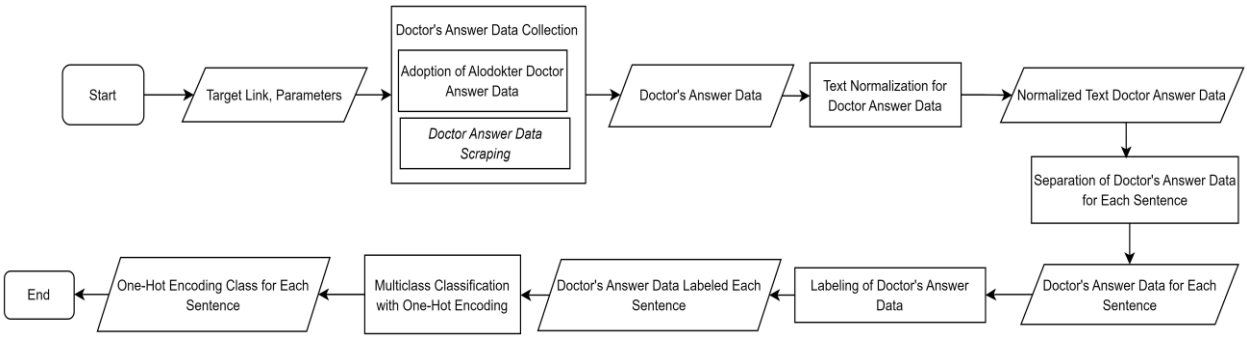


Figure 2: Doctors' answer dataset preparation.

https://data.mendeley.com/datasets/p8d5bynh3m/1, file name Indo-Online Health Consultation-Multilabel-Raw.csv. DS2 and DS3 were taken by scraping data from the OHC platform. DS2 is taken from the Alodokter platform, https://www.alodokter.com/komunitas/diskusi/penyakit, from February 29, 2021, to November 2024. DS3 is taken from several OHC platforms, namely HelloSehat (https://hellosehat.com/community/), detikHealth (https://health.detik.com/konsultasi), and Mayapada Hospital (https://mayapadahospital.com/). The three OHC platforms have a small number of doctors' answers, so the three doctors' answers are combined from January 2024 to December 2024. DS1 has 15441 data with 306955 sentences, DS2 has 36247 data with 495478 sentences, and DS3 has 3111 data with 53063 sentences. All doctors' answers are in Indonesian. A doctor's complete answer text consists of several sentences that provide medical information to the reader regarding the user's condition. This complete doctor's answer text is used as input in the MLPSentFeat model. The doctor's answer data is stored in CSV format, where each doctor's answer text is placed in a single cell containing multiple sentences.

The gathering aspect of doctors' answers on various platforms is very small; the Alodokter platform has a slightly higher number than other platforms. The response to the emotional aspect has three types of sentences: sympathy, empathy, and hope. This type of sympathy and empathy sentence has a small number of sentences, while the type of sentence of hope has a large number. Responding to emotions in the doctor's answer data from other platforms is more varied than Alodokter's. The aspect of responding to emotions, especially the sentences of sympathy and empathy, is of particular concern to the writer, because different annotators give this aspect different labels. First, the doctor's answer data is annotated by the midwife. This type of sentence of hope is labeled as fostering the relationship. Second, the doctor's answer data is annotated by the doctor. This type of sentence of hope is labeled as responding to the emotion. So, initially, the response to the emotion aspect has a small number, then it increases to a large number after getting additional types of hope sentences. Here, the MLPSentFeat model still pays attention to the types of empathy and sympathy sentences, so sentence features are also formed from the response to the emotion aspect.

Doctors rarely reveal the information-gathering aspect because this consultation is asynchronous. So, doctors do not want to dig up information from patients. Apart from that, some platforms do not provide a place to reply to doctors' questions. Doctors rarely express sympathy and empathy because patients do not explain their illnesses in detail. In addition, it is not easy to understand other people's emotions through text [5].

B. Text normalization for doctor answer data

Text normalization is an essential step in text standardization, reducing variation and ensuring consistency in format before being processed by the model. In this study, the doctor's answer data is normalized into two categories: general normalization and specific normalization. These text normalization techniques are performed using simple text normalization methods.

General normalization focuses on changes made to the basic structure of the text to improve consistency and uniformity. Examples of general text normalization applied to the doctor's answer data include the separation of hyphens (-), slashes (/), question marks (?), and exclamation marks (!) from the words they follow. This step separates the words before and after these marks during the tokenization process. If this step is not performed, the words before and after these marks would merge into one word, potentially altering the meaning of the word. For example, 'rata-rata' is changed to 'rata - rata', 'posisi tidur/horizontal' is changed to 'posisi tidur / horizontal'. Additionally, periods (.) following numbers, acronyms, or titles are removed. This step aims to place the period at the end of a sentence, as the initial input of this study is the intact doctor's answer data, which is separated by sentences by identifying the period as one of the sentence-ending markers. For example, *'1. Minum air putih* (1. Drink water)' is changed to *'1 Minum air putih'*, '10.000' is changed to '10000', *'dr. Syifa'* is changed to 'dr Syifa' (example of writing a doctor's title), and *'Spesialis Penyakit Dalam (Sp.PD)* (Internal Medicine Specialist)' is changed to *'Spesialis Penyakit Dalam (Sp PD)*' (example of writing an acronym). These steps ensure that the text is more consistent and can be processed more effectively by the model. In addition, there is also a step for correcting spelling errors. This step aims to ensure that a word is recognized as the same word, rather than being treated as a different one by the system. For example, a misspelled word such as '*inum'* is corrected to '*minum* (drink)'. If such errors are not corrected, the system may interpret

them as two distinct entities, which can lead to inaccurate analysis.

Specific normalization focuses on more specific changes related to the context or information present in the text, such as the processing of certain sentences or words to improve analysis accuracy. The goal of this specific normalization is to standardize words in the doctor's answers so that they can be classified correctly according to their communication aspects. Additionally, this specific text normalization aims to standardize the way steps in various instructions or tips are written. In some cases, these steps are often written in one long sentence, which can make it difficult for the model to recognize and classify sentences according to the relevant communication aspect. Therefore, these steps are organized into separate sentences to make it easier for the model to clearly recognize each step and identify the intended communication aspect. As a result, the model can classify the sentences more accurately and efficiently. In the initial experiment, when the model used very limited data, without specific normalization, the model struggled with classification. This normalization helps reduce data variation, allowing the model to classify more accurately despite the limited data available.

Steps in specific normalization include, first, shortening long sentences by separating clauses in those sentences into shorter, more manageable sentences. Additionally, this step is crucial because a sentence often contains several communication aspects, whereas each sentence should ideally contain one clear communication aspect. This shortening is performed carefully to ensure that the meaning of the sentence remains intact. For example, *'Beberapa Tips yang Anda dapat lakukan untuk mencegah terjadinya ISK atau infeksi pada saluran kemih yaitu Minum air putih minimal 2-liter perhari Jangan menahan keinginan BAK atau Buang Air Kecil* (Some tips that you can do to prevent UTI or urinary tract infections are: Drink at least 2 liters of water per day Don't hold back the urge to urinate)'. This sentence, which is about self-medication tips and consists of three clauses, is split into three sentences. Inaddition, this sentence also has two communication aspects: providing information and enabling disease and treatment-related behavior, thus it must be separated into several shorter sentences according to the clauses. The sentence is changed to, Sentence 1: *'Beberapa Tips yang Anda dapat lakukan untuk mencegah terjadinya ISK atau infeksi pada saluran kemih yaitu*.', sentence 2: '*Minum air putih minimal 2-liter perhari*.', sentence 3: '*Jangan menahan keinginan BAK atau Buang Air Kecil.*' Sentence 1 includes the information-giving aspect, while sentences 2 and 3 include the aspects of enabling the occurrence of disease and behavior related to treatment.

The second step is the process of synonym standardization to ensure consistent word usage in the text. For example, in the aforementioned sentence, acronyms ISK and BAK are added. This step aims to reduce the model's tendency to associate acronyms from one language with English, considering that some word embedding, classification, or segmentation methods often use English corpora in their training process. Furthermore, this step aims to reduce ambiguity in acronyms, improve classification and segmentation accuracy, and ensure consistency in the interpretation of terms used. Thus, the model can be trained more effectively and accurately in understanding the appropriate context.

The third step is the process of standardizing words for different synonyms. The goal of synonym standardization is to reduce variation in the data and ensure consistency in text processing, allowing the model or language processing system to more easily understand the same concept even when expressed with different words. For example, the word '*menstruasi* (menstruation)' has the synonym '*haid* (menstruation)'. Both words are standardized to 'menstruasi'. The choice of the word 'menstruasi' as the standardized form, compared to 'haid', is based on several reasons. The word '*menstruasi'* is more commonly used in formal and technical contexts, particularly in medical and academic literature, while '*haid'* is more frequently used in everyday conversation. Thus, using 'menstruasi' ensures consistency in the use of standardized terminology, especially in scientific and medical writing, which is crucial for maintaining the quality and clarity of information. By standardizing synonyms, such as equating 'menstruasi' and '*haid'*, the model will not consider them as two different entities.

The fourth step is the process of standardizing the writing of specific words. For example, the word '*terimakasih* (thank you)' is often written as a single word, but some doctors write it as two separate words, '*terima kasih'*. The standardized form of this word is the one written separately, so in this process, the writer standardizes the word by writing it separately. This step ensures that the model does not perceive it as two different entities, which could lead to confusion in understanding the context. Additionally, it helps to improve consistency in the data and facilitates the tokenization and text analysis process, allowing the model to recognize the word more accurately.

The fifth step is the process of using consistent acronyms for the same term. For example, the word '*Tuberkulosis* (Tuberculosis)' can be abbreviated as 'TBC' or 'TB'. The author chooses the acronym 'TBC' for this standardization because 'TBC' is more commonly used in formal writing in Indonesia, even in everyday conversation. The goal of this step is to ensure consistency in terminology usage, so the model does not consider it as two different entities.

The sixth step is the process of separating combined question sentences into individual question sentences. For example, the sentence 'apakah ada demam, terasa mual, muntah (Is there a fever, nausea, vomiting)' consists of one sentence containing three symptoms that need to be identified. The sentence is then converted into three separate sentences: Sentence 1: '*Apakah ada demam ?* (Is there a fever ?)', Sentence 2: '*Apakah terasa mual ?* (Do you feel nauseous ?)', and Sentence 3: '*Apakah ada muntah ?* (Is there vomiting?)'. This step is performed to increase the number of sentences with the gathering information aspect, considering that this aspect has relatively few sentences.

The seventh step is the process of adding question words and question marks (?) to question sentences. For example, a sentence that initially does not have a question mark, such as '*Apakah anda menjadi tidak bisa belajar atau bekerja* (Are you unable to study or work)', is changed into a sentence with a question mark, '*Apakah anda menjadi tidak bisa belajar atau bekerja ?'*. This step is intended to standardize question sentences, where a question sentence is generally initiated by a question word and ends with a question mark (?). Additionally, this step aims to facilitate the classification of the gathering information aspect, considering that this aspect has a limited number of sentences and a variety of question sentence forms. Without standardizing the question sentence forms, the model would struggle to classify the doctor's answer sentences into this aspect.

The eighth step is paraphrasing sentences. For example, the sentence '*Bau bagaimana yang Anda maksudkan ?'* is revised to '*Bagaimana bau yang Anda maksud ?* (What smell do you mean?)'. The original sentence is an interrogative sentence; however, it does not adhere to the standard rules of interrogative sentence construction, which require that a question begin with a question word. Such a sentence is more appropriate in informal or semi-formal spoken contexts. Therefore, the original sentence is reformulated to align with the conventions of formal written language. The purpose of this step is to improve sentence structure in accordance with standard writing conventions, enhance readability, and ensure that the meaning of the sentence is preserved despite changes in form. Paraphrasing is also performed to adjust the language style from informal or spoken forms to more formal and standardized written forms.

The ninth step involves adding punctuation marks to sentences that lack them or correcting the placement of punctuation that is inaccurately positioned. This step is intended to improve readability, clarify sentence structure, eliminate ambiguity, accurately mark sentence boundaries, and ensure compliance with standard Indonesian writing conventions. As an example, the sentence used in Step One can be referenced. The sentence consists of several phrases, but there are no punctuation marks separating them. Therefore, a comma can be inserted to distinguish the individual phrases. Additionally, the sentence does not end with a period, so a full stop should be added to mark the end of the sentence. The revised version is as follows: '*Beberapa tips yang Anda dapat lakukan untuk mencegah terjadinya ISK atau infeksi pada saluran kemih yaitu minum air putih minimal 2 liter per hari, jangan menahan keinginan BAK atau Buang Air Kecil.*' (Some tips that you can do to prevent UTI or urinary tract infections are: drink at least 2 liters of water per day, don't hold back the urge to urinate.).

These normalization steps—both general and domain-specific—are essential to ensure that the doctor's response data can be easily interpreted by both the model and human readers. The purpose of this normalization is not only to facilitate the model's understanding of the text and its ability to perform accurate classification, but also to ensure that the data can be comprehended by medical annotators who are responsible for labeling the data. Data that is not clearly readable can hinder the labeling process by medical annotators. By ensuring that the data is easily understandable by readers, normalization also supports the model in comprehending the meaning of the text and classifying the data more accurately.

General text normalization is applied to the data from DS1, DS2, and DS3. The basic structure of the text must be consistent across these datasets, as this consistency enables the model to recognize relevant patterns and features more effectively, increases accuracy in classification and analysis, and facilitates the tokenization and processing of the data. Moreover, this consistency reduces the potential for the model's confusion in identifying the true meaning, thereby improving the overall performance of the model. Specific text normalization is applied to the DS1 data because it contains many spelling errors and structural inconsistencies. Therefore, specific normalization is necessary to correct these spelling errors and improve the consistency of the text, allowing the model to process the data more effectively. This step also allows for assessing the impact of normalization on the model's performance in classification and analysis.

C. Separation of doctor's answer data for each sentence

The process of separating the doctor's answer data per sentence aims to divide the doctor's answer text into individual sentences. It means that each doctor's answer, initially located in a single cell, will be separated into one sentence per cell. This process is carried out because Python (the programming language used to build this model) reads the content in a single cell as one sentence, even though the cell contains multiple sentences. Since this study performs text segmentation at the sentence level, it is necessary to separate the sentences to process each individually.

The text segmentation process requires separating multiple sentences to be processed effectively. Working with just one sentence is insufficient, as these sentences will be divided into smaller segments. Therefore, to meet the segmentation processing requirements involving multiple sentences and Python's reading of cells, the author separates each sentence in the doctor's answer data into separate cells originally stored in one cell. Sentence identification is done by detecting period (.), question mark (?), and exclamation mark (!) as indicators of sentence boundaries [88]. This separation process is carried out automatically by identifying these three markers, and each sentence is stored in a separate cell. The doctor's answer data separation per sentence is applied to the entire DS1, DS2, and DS3 data.

Table 3: Distribution of doctor response data based on aspects of communication.

Data	Labelled Data								Unlabelled Data	
	Total Data	Total Sentences	F1	F2	F3	F4	F5	F6	Total Data	Total Sentences
DS1	5410	106231	16147	977	55824	9270	5446	18567	10031	200724
DS2	8000	110562	16555	274	48142	8246	8000	29345	28247	384916
DS3	1000	23280	1987	73	14460	1818	753	4189	2111	29783

Thus, the doctor's answer text, initially stored as a single unit in one cell, is now separated into multiple cells, each containing one sentence. This sentence segmentation is crucial, as the relatively short length of the doctor's answer text often leads to changes in communication aspects at the sentence level. By separating the doctor's answers based on sentences, the model can more easily recognize different communication aspects in each sentence, which would not be possible if the text were left as one long sentence. For example, changes in communication aspects at the sentence level can be observed in the previous explanation (point B), where in the example, the first sentence falls under the aspect of providing information. In contrast, the second and third sentences are categorized as enabling disease and treatment-related behavior. Therefore, by dividing the text into separate sentences in different cells, each sentence can be processed as a smaller and independent unit. This action allows the model to recognize and analyze each sentence in the doctor's answer text more accurately.

D. Labeling of doctor's answer data

The data labeling process aims to label each sentence in the doctor's response. Each sentence within the text is analyzed to identify the relevant aspect of communication. This labeling procedure determines that each sentence is assigned only one label for the communication aspect. The single-label-per-sentence approach is applied to avoid interpretative conflicts among annotators. This approach is essential to ensure data consistency and reliability, particularly when a sentence contains multiple meanings that each annotator may interpret differently. Moreover, manual labeling requires intensive cognitive processing. Annotators must comprehend the meaning of each sentence, recall the criteria for each communication aspect, and determine the most appropriate label. If annotators are required to assign more than one label, their cognitive load increases significantly, potentially affecting the accuracy and consistency of the labeling outcomes. Equally important, this approach also considers time efficiency, considering that the annotators are practitioners such as doctors with limited time availability. Restricting labeling to a single dominant aspect per sentence allows the annotation process to proceed more quickly and in a more focused manner while still maintaining the quality of the results.

The determination of these communication aspects is thoroughly explained in Subsection 2.1, which outlines the criteria for categorizing sentences under the appropriate communication aspect. Once the corresponding aspect is identified, each sentence is labeled F1 through F6, representing the various categories of communication aspects described in Subsection 2.1. This labeling is performed at the sentence level across the entire doctor's response. For example, Sentence 1 in Figure 1 is a greeting by the doctor directed at the patient or user. This greeting corresponds to the “fostering the relationship” aspect, wherein greeting the patient is one of the doctor’s responsibilities; therefore, Sentence 1 is labeled as F1. Sentence 4 is a question posed by the doctor to the user, aligning with the “gathering information” aspect, as this category involves the doctor asking open-ended questions; hence, it is labeled as F2. Sentence 6 is an informational statement from the doctor to the user, representing the “providing information” aspect, as this includes the doctor’s role in sharing relevant information with the patient; thus, it is labeled F3. Sentence 7 suggests that the user consult an obstetrician in person. This approach aligns with the “decision-making” aspect, where the doctor’s responsibility includes planning collaborative actions—such as referring patients to other doctors—thereby earning a label of F4. Sentence 8 is a self-care recommendation that fits the “enabling disease- and treatment-related behavior” aspect, where doctors advise actions supporting appropriate health-related behavior. Thus, it is labeled as F6. Sentence 11 expresses the doctor’s hope that the information provided will be helpful. Such expressions may help ease the patient’s emotions by making them feel acknowledged and are part of the “responding to emotions” aspect; accordingly, Sentence 11 is labeled as F5.

The doctor's responses were labeled by medical annotators, consisting of five general practitioners. Each physician annotated different subsets of the dataset during different periods, ensuring no annotator labeled the same data. This approach was necessary due to the large volume of unlabeled data requiring annotation. The process was conducted manually, with annotators carefully examining each sentence and assigning the most appropriate label based on the contextual communication content. The labeling results were manually verified by comparing them against predefined labeling guidelines for each category. In addition, the labeled data were re-reviewed by a more experienced annotator to ensure that all labels adhered to the defined standards.

Based on the labeling outcomes, the occurrence of communication aspects within a single doctor response does not follow a fixed order from F1 to F6 (i.e., from fostering the relationship to responding to emotions), and multiple aspects may appear more than once within a single response. Moreover, not all responses encompass all six communication aspects. In many cases, only a subset of the aspects is present. This variation is likely influenced by the individual doctor’s writing style,

$$LR\ of\ Feature\ X = \frac{Frequency\ of\ X\ in\ CorpusL}{(Frequency\ of\ X\ in\ CorpusL\ +\ Frequency\ of\ X\ in\ CorpusNon)} \quad (1)$$

$$Baseline\ LR = \frac{Number\ of\ Words\ in\ CorpusL}{(Number\ of\ Words\ in\ CorpusL\ +\ Number\ of\ Words\ in\ CorpusNon)} \quad (2)$$

$$G(D,t) = -\sum_{i=1}^{m} P(Ci)logP(Ci) + P(t)\sum_{i=1}^{m} P(Ci|t)logP(Ci|t) + P(\overline{t})\sum_{i=1}^{m} P(Ci|\overline{t})logP(Ci|\overline{t}) \quad (3)$$

intuition, and clinical judgment. For example, in dataset DS1, only 294 out of 5,410 responses contained all six aspects, meaning that approximately 94.56% of the data did not cover the full range of communication categories. This dataset indicates that only a few responses are comprehensive and aligned with all six communication aspects. These findings reflect the contextual and selective nature of doctor-patient communication in clinical practice, which often does not address every ideal aspect of communication. A similar pattern was observed in DS2 and DS3. In DS2, only 130 out of 8,000 responses (1.63%) included all six aspects. In DS3, only 2 out of 1,000 responses (0.2%) met this complete communication criteria. These results consistently demonstrate that doctor communication across all three datasets tends to be partial, varied, and seldom incorporates all predefined communication aspects.

The total number of labeled and unlabeled responses and the distribution of sentences across labeled communication aspects are presented in Table 3. Although the dataset contains a relatively large number of sentences, segmentation requires complete doctor responses. As shown in Table 3, the distribution of labeled data across DS1, DS2, and DS3 indicates a significant imbalance among communication aspects. The "providing information" aspect (F3) consistently dominates all datasets, with 55,824 sentences in DS1 and 48,142 in DS2—far exceeding other aspects. In contrast, the "gathering information" (F2) and "responding to emotions" (F5) aspects are underrepresented, particularly in DS2 (only 274 and 8,000 sentences, respectively), and even fewer in DS3. This imbalance suggests that doctors' responses are more focused on delivering information than on elicitation or emotional engagement.

In terms of volume, DS2 contains the largest number of labeled and unlabeled data, indicating its strategic role in training and evaluation. Although DS1 is smaller, it remains relevant due to its balanced ratio between labeled and unlabeled data. Even though DS3 has the most limited size, it is valuable for use as a separate evaluation set to test model generalizability on smaller-scale data.

Initially, the "responding to emotions" aspect of the dataset primarily included empathetic and sympathetic sentences [5]. However, upon further review, additional sentences in the form of "wish" expressions were added to this category. Since doctors frequently included wishes in their responses, these expressions were initially mislabeled as "fostering the relationship." However, they were later reclassified under "responding to emotions" to more accurately reflect their communicative function.

E. Multiclass representation using one-hot encoding

This process aims to convert class labels in a dataset containing more than two classes (multiclass classification) into a binary vector format using the one-hot encoding technique. As described, each sentence in the doctor's response is assigned a single communication aspect label. The classification involves six distinct communication aspects, symbolized as F1 through F6 (further explanation can be found in Subsection 2.1). Accordingly, each sentence is categorized into one of these six aspects. This type of classification is known as multiclass classification, where each sample (in this case, a sentence) is assigned to one class out of several available options [89]. In other words, once a sentence is assigned to one label, it cannot simultaneously belong to another. For example, if sentence-1 in Figure 1 is labeled as F1, it will not be assigned to any remaining labels (F2 to F6).

The one-hot encoding technique transforms the assigned labels for each sentence in the doctor's response into binary vectors. One-hot encoding represents each class in a multiclass classification task as a binary vector, where the value 1 indicates the selected class and the value 0 indicates the remaining classes [90][91]. One-hot encoding was chosen because it allows categorical labels to be numerically represented without assuming any ordinal relationship or hierarchical weight among the classes. An important consideration in multiclass classification tasks involving mutually exclusive categories, such as the six communication aspects (F1 through F6), which are conceptually equal and independent. Moreover, this representation enables direct comparison between predicted and actual labels, thereby supporting model performance evaluation based on class-specific metrics such as accuracy, precision, recall, and F1-score.

In this representation, each communication aspect class (F1 to F6) is mapped to a column, with a value of 1 assigned to the column corresponding to the true label and a value of 0 assigned to all others. For instance, if Sentence 1 in Figure 1 is labeled as F1, the F1 column will have a value of 1, while the columns for F2 to F6 will have a value of 0, resulting in a one-hot vector of [1, 0, 0, 0, 0, 0]. Similarly, if Sentence 6 is labeled as F3, its one-hot representation will be [0, 0, 1, 0, 0, 0], where the F3 column holds the value 1 and the remaining columns (F1, F2, F4, F5, and F6) each hold the value 0.

The one-hot encoding scheme represented all sentences in the DS1, DS2, and DS3 datasets. This format was consistently applied throughout the classification model's training and evaluation stages. The one-hot representation facilitated direct comparison between predicted and actual labels and supported the computation

of performance metrics such as accuracy, F1-score, precision, and recall for each individual class.

3.2. Pseudo-labeling

This stage was designed to expand the data by utilizing new data that has not been labeled. This data labeling was done by pseudo-labeling using a voting classifier from several machine learning models. This data addition was carried out to overcome overfitting, the first research problem. Using pseudo-labeling is necessary because manual labeling by annotators requires significant time, costs, and effort [92][93]. Using multiple methods for pseudo-labeling can reduce labeling errors by exploiting the advantages of each method to increase correct labels [93]. The pseudo-labeling stage in Figure 3 has four processes: separating the doctor's answer data labeled by the doctor with k-fold, preprocessing training data, training data and label prediction, and voting.

The input for the pseudo-labeling stage, namely labeled data and unlabeled data, can be seen in Table 3. The output for the pseudo-labeling stage is labeled unlabeled data.

3.2.1. Separating doctor's answer data labeled with K-Fold

The labeled doctor's answer data was split into training and test data, 80:20, using k-fold cross validation, which was rotated five times. Training data is for testing the model, while testing data is for testing the model. Cross-validation can overcome overfitting because it divides data fairly between training and testing [65][94]. K-fold cross-validation refers to [65], where the testing set is usually smaller than the training set to get better performance. K-fold cross validation divides the data into k equal subsets (folds), the model is iteratively trained in k-1 folds and validated against the remaining folds [95]. The k-fold cross-validation method can help reduce uncertainty in data sharing and prevent overfitting (the model fits the training data too well, so it is less accurate on new data) [94].

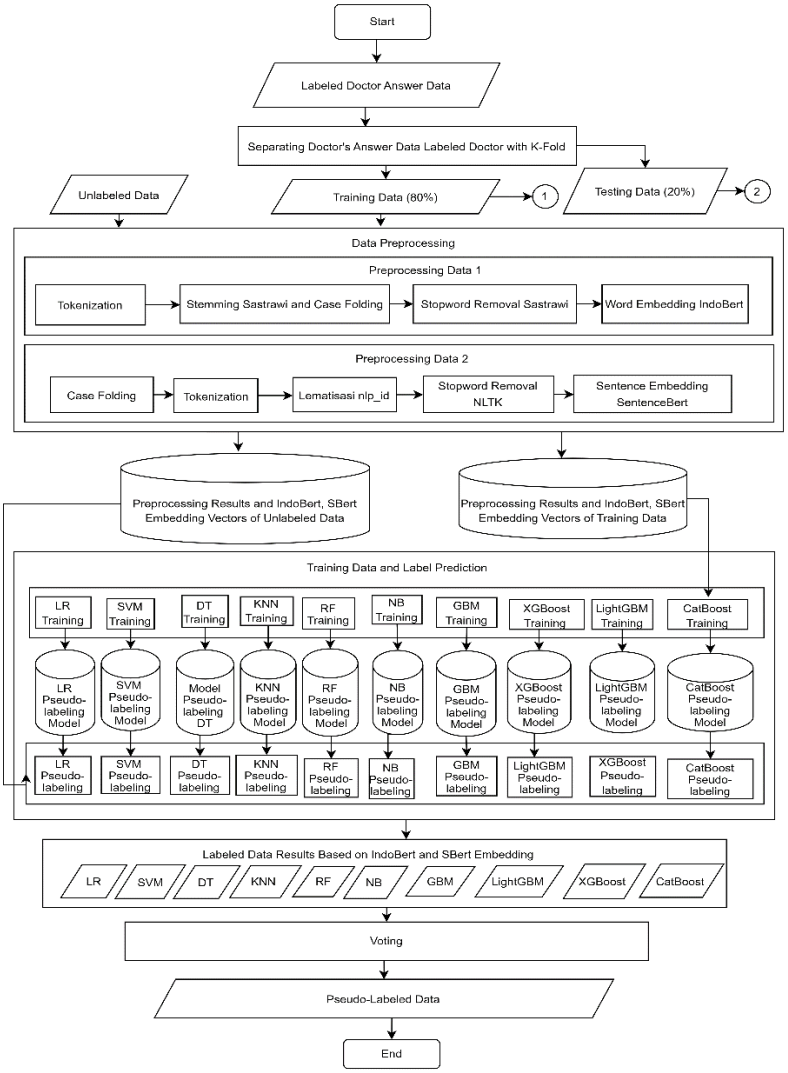


Figure 3: Pseudo-labeling architecture.

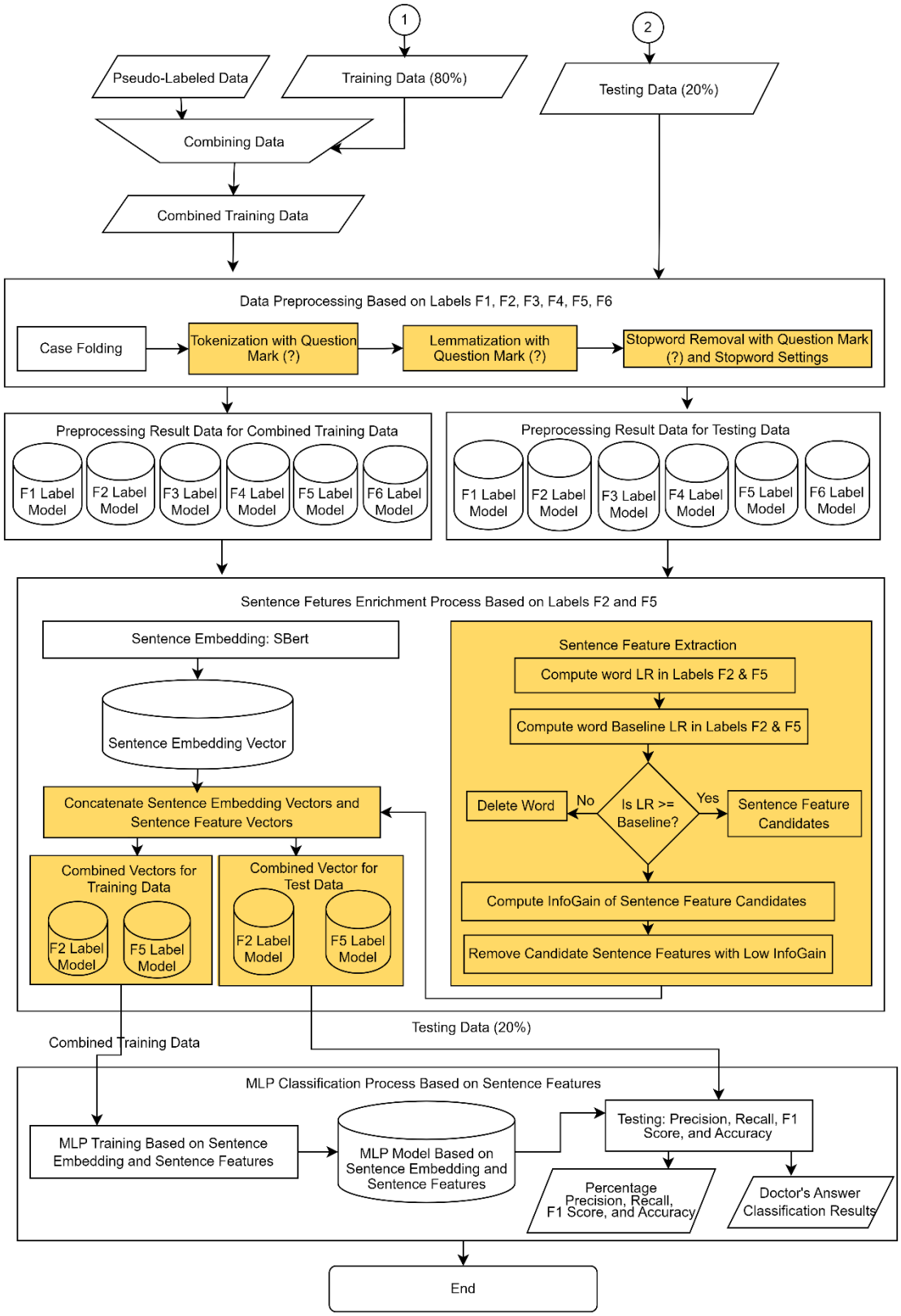


Figure 4: MLPSentFeat architecture.

3.2.2. Data preprocessing for pseudo-labelling process

The training data is used to train the pseudo-labeling model. The training and unlabeled data undergo a pseudo-labeling process: tokenization, root word retrieval, stopword removal, and embedding. The output of this stage is the results of preprocessing and embedding vectors from training and unlabeled data. Preprocessing results have two types: based on IndoBert and SBERT. So, there are four results of this process, namely, preprocessing results and embedding vectors from IndoBert training data, preprocessing results and embedding vectors from SBERT training data,

preprocessing results and embedding vectors from IndoBert unlabeled data, and preprocessing results and embedding vectors from SBERT unlabeled data. The training data's preprocessing results and embedding

vectors are used to train the machine learning model. In contrast, the preprocessing results and embedding vectors from unlabeled data are used for the pseudo-labeling process.

The embedding methods used are IndoBert [96] and SBERT, which are applied alternately. IndoBERT is a BERT variant optimized for Indonesian texts [97], providing accurate and efficient classification results [98]. These two embeddings are used because they can capture different sentence features [99].

3.2.3. Training data and label prediction

This training model classifies doctors' answers using multiclass single-label classification. This classification means that the target class is represented in one column with several possible labels, but each sample only has one label assigned [100][101]. So, the doctor's answer data has one target class, which contains six labels for aspects of doctor-patient communication, and each sentence of the doctor's answer is classified into one label.

This stage has two processes: first, training the training data with machine learning methods, LR, SVM, DT, KNN, RF, NB, GBM, Xgboost, LGBM, and CatBoost models with SBERT and IndoBert embedding. Second, the pseudo-labeling process uses the pseudo-labeling model. The result of this stage is labeled data with a total of 20 labels, obtained based on the training model, consisting of 10 labels with SBERT embedding and 10 labels with IndoBert embedding.

3.2.4. Voting

Determining the final label results for each doctor's answer sentence is done by voting based on the labeling results of each model. The pseudo-labeling results of 20 models were combined, then counted, and the majority vote was taken. The label with the most votes is the final label for the doctor's answer sentence. For example, a sentence labels F3 from 15 models and F4 from 5. Because label F3 has the most significant number, the final label chosen for the sentence is F3. This final label is the pseudo-labeling result used for the following model process.

3.3. Classification process based on six aspects of communication

This classification stage uses the MLPSentFeat model to see the model's performance in classifying doctors' answer data using communication aspects. The classification stage architecture is shown in Figure 4, which has three processes: data preprocessing, sentence features enrichment process, and text classification with the MLPSentFeat model. The colored part is the latest version of this research model, namely the preprocessing settings based on labels or communication aspects, adding sentence features based on labels or communication aspects, and combining sentence embedding vectors with sentence features. The input for the classification stage is labeled data from the pseudo-labeling process, training data is 80% of the labeled data, and test data is 20% of the labeled data. Labeled data from the pseudo-labeling process is combined with training data, as much as 80% of the labeled data, into training data, called combined training data.

Meanwhile, test data is used to test model performance. This testing model's output is the MLPSentFeat model's performance, and the doctor's answer data is classified. Research [102][103] also combines labeled data from pseudo-labeling results into training data.

3.3.1. Data Preprocessing based on labels

The doctor's answer data, combined with training and test data, underwent a preprocessing process, namely case folding, tokenization, lemmatization, and stopword removal. Several preprocessing processes are arranged based on specific label requirements (F1, F2, F3, F4, F5, and F6), namely the tokenization, lemmatization, and stopword removal processes. It is because the words or characters needed to recognize a label differ from one label to another. These preprocessing settings, especially stop words, are adjusted to the sentence structure and words used in a label by considering explanations from aspects of communication between doctors and patients. Therefore, it is important to understand the form of sentences and the use of words in the doctor's answer data according to the communication aspect. For example, the F2 label has an interrogative sentence form, so it requires a question mark (?), then tokenization, lemmatization, and stopword removal are processed, and a question mark (?) appears for class F2. By default, question marks (?) do not appear in preprocessing. In addition, stopword settings are based on specific label needs. For example, label F2 also requires a question word, so the question word is excluded from the stopword dictionary.

Another example is labels F4 and F6, prohibitive command sentences with negative words. Setting stopwords in labels F4 and F6 brings up negative words by removing them from the stopword dictionary. This also applies to other labels.

This preprocessing setting allows adding information in sentences after the preprocessing process, so the model can classify correctly because it can distinguish one sentence from another. In addition, additional information from the preprocessing setup process can act as additional data that can be utilized by the model in the process of adding sentence features. These two processes (preprocessing settings and sentence features) complement each other; the preprocessing settings process adds additional information that can be captured by adding sentence features, and vice versa. The output from this preprocessing stage is the preprocessing model for each label for the training and test data.

Algorithm 1: Sentence-fetures enrichment process.

Input: Sentence after preprocessing
Output: Sentence feature vector V
Procees:
Step 1: **Calculate Likelihood Ratios (LR):**
For each word in labels **F2** and **F5**, compute the LR using *Equation 1*.
Step 2: **Calculate Baseline LR:**
For labels **F2** and **F5**, compute the baseline LR using *Equation 2*.
Step 3: **Select Significant Words Based on LR:**
- For each label (**F2** and **F5**):
- If a word's **LR > Baseline LR**, retain the word.
- Otherwise, discard the word.
- The retained words will proceed to the next step.
Step 4: **Compute InfoGain:**
- For each retained word, calculate its InfoGain.
- Test the MLPSentFeat model with all selected words.
Step 5: **Filter Words with Low InfoGain:**
Remove words one by one starting with low InfoGain. For each removal:
- Test the MLPSentFeat model with all selected words.
-
- Compare the accuracy of the model with the current features with the accuracy of the model with the overall features.
- If a decrease in accuracy is found, stop removing features.
- Make the word a boundary for selecting sentence features.
Step 6: **Apply Sentence Features to Classification Model:**
For each sentence in the dataset:
- Create a feature attribute for each selected word.
- If a word from the sentence is in the feature set, assign **1** to the corresponding attribute; otherwise, assign **0**.
- Combine all attribute values into a **sentence feature vector**.
Step 7: **Combine Sentence Feature Vector with Embedding Vector:**
- Merge the sentence feature vector with the corresponding embedding vector to form a **combined vector**.
Step 8: **Feed Combined Vector into Classification Model:**
- Use the combined vector as input for the classification process.

3.3.2. Sentence vector enrichment process based on the labels

This sentence feature enrichment process adds sentence features to the MLPSentFeat model. Sentence features help the model capture additional information that cannot be captured with just the basic sentence embeddings. There are four processes in this stage: sentence embedding, sentence feature extraction, concatenating sentence embedding vectors and sentence feature vectors, and reducing sentence features. The sentence feature enrichment stage output is a model containing a combined vector of sentence embeddings and sentence features for each label, F1, F2, F3, F4, F5, and F6.

The sentence embedding process uses the SBERT method [104]. The result of this process is a sentence embedding vector. The length of the doctor's answer sentence vector is 384.

The sentence feature extraction process is carried out by adding sentence features to labels F2 and F5, considering that these two labels have a small number of sentences. The purpose of adding this sentence feature is to correctly classify labels with a small number by capturing certain information in the two labels.

The first step in determining sentence features is calculating the LR for each word in labels F2 and F5. The LR calculation formula can be seen in equation 1. Second, calculate the LR baseline at labels F2 and F5. The formula for calculating the baseline LR can be seen in equation 2. Third, choose words with an LR greater than the baseline in each class. A word with an LR greater than the baseline is retained, and vice versa. Words that have an LR greater than the baseline will be processed further. Fourth, calculate the InfoGain for each word. Fifth, remove words that have very small InfoGain.

The InfoGain calculation formula can be seen in equation 3. A small InfoGain indicates that the word does not significantly influence the data, so the word needs to be discarded. Then, the sentence features that have been formed are tested against the classification model. This word deletion stops when the accuracy resulting from the classification model is greater than the previous accuracy for the first time. Sixth, after the sentence features are formed, the sentence features are applied to the classification model. The application of sentence features is done by making sentence features into attributes, then filling in the attributes by counting the words that include sentence features for each sentence. If a word is a sentence feature, it will be given a value of 1 in the sentence feature attribute. Otherwise, it will be given a value of 0. Then, the values of each attribute are combined into a vector. Seventh, sentence feature vectors are combined with embedding vectors. The combination of these two vectors is called a combination vector. This combination vector is processed in the classification model. A complete explanation of adding sentence features is provided in Algorithm 1.

The LR calculation and LR baseline refer to [105][106], which calculates the frequency of word occurrences based on specific criteria, for example, sentence features are taken based on ambiguous or unambiguous sentences. The InfoGain method refers to [107], which performs feature selection using InfoGain and a genetic algorithm. InfoGain is an entropy-based feature selection method used in machine learning to measure how much information a feature provides about a text category. InfoGain helps determine the importance of a term in classification by calculating its contribution to differentiating information categories. The following are the equations for LR, baseline LR, and information gain.

Where, LR of Feature Frequency of Frequency of X in CorpusNon is the frequency of occurrence of words other than the labels being processed. For example, LR calculations are carried out on label F2, and CorpusNon is the frequency of word occurrences from all labels other than F2.

The number of Words in CorpusL is the number of words in a particular label, for example, label F2. The number of words in CorpusNon is the number of words other than processed labels. For example, LR calculations are carried out on label F2, and CorpusNon is the number of words from all labels other than F2.

Where C is a collection of documents with no feature t, the larger the value of G(D,t), the more useful t is for

classification in C. This feature t should be selected. If a larger value of G(D,t) is desired, then the values of P(t) and $P(\bar{t})$ must be smaller.

3.3.3. MLP Classification based on Sentence Embedding and Sentence Features Model

The classification experiment used the MLPSentFeat model to test its performance based on the communication aspects of doctors' answer data. The MLP model is a classification model because MLPSentFeat can classify doctors' answer data well and even outperform other models [5].

This classification stage has two processes: the model training process using training data and the data testing process using test data. This model produces classification performance and classifies doctors' answer data.

Here are some initial parameters used in the MLP model. The input layer consisted of one layer with several neurons in the form of a combination vector. The length of this combination vector depended on the length of the sentence feature vector, while the length of the SBERT embedding vector was 384. There was one hidden layer with a total of 100 neurons. The output layer consisted of one layer with six neurons, representing the number of communication aspect labels, namely F1, F2, F3, F4, F5, and F6. The MLP parameters used included max_iter = 1000, activation = 'relu', solver = 'adam', and random_state = 42. These parameters were set at this stage before the optimization process began.

Optimization was performed on the MLP model to improve classification results, focusing mainly on the number of hidden layers and neurons within them. The reason was that the hidden layer played an important role in the MLP model in capturing more complex and non-linear patterns in the data. Therefore, adjusting the number of neurons and hidden layers to optimal values could improve the model's capacity for better generalization. Meanwhile, the input and output layers were generally more structured and stable, so they were less frequently optimized than the hidden layer, which had a central role in information processing and decision-making.

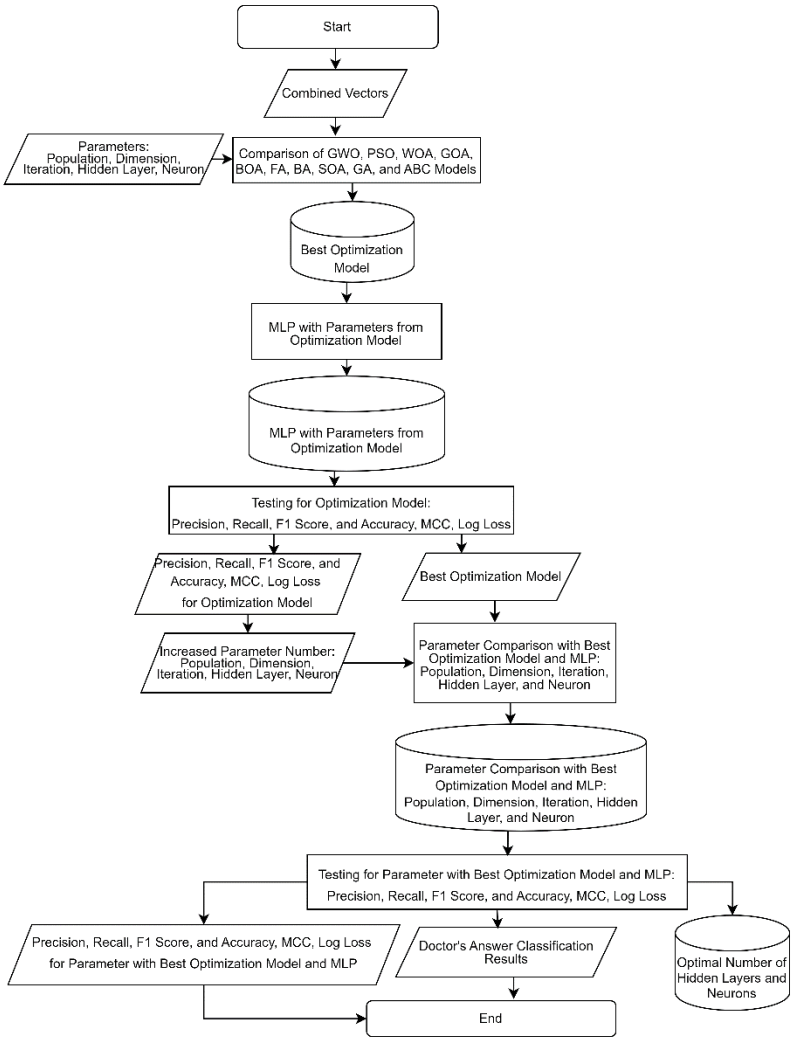


Figure 5: Optimization architecture.

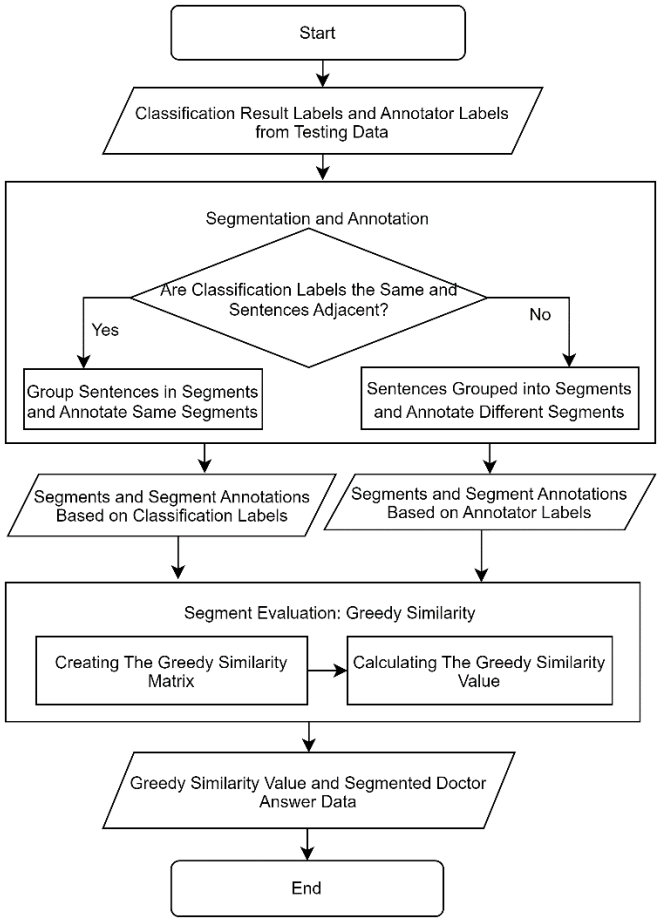


Figure 6: Text segmentation architecture.

Various optimization methods were applied to obtain the optimal number of layers and neurons in the MLP model to complete the classification task on the doctor's answer data based on communication aspects. Additionally, previous studies that served as references showed that by optimizing the number of hidden layers and neurons using optimization methods, the model achieved better performance in recognizing complex patterns [70]. Therefore, optimizing these parameters was considered more appropriate than modifying other parameters.

Testing uses precision, recall, accuracy, and F1 score measuring tools, which refer to [65]. Precision measures how accurate or precise a prediction is compared to the actual results. The precision formula is shown in equation (4). Recall or sensitivity measures how many positive samples are correctly identified by the model. The recall formula is shown in equation (5). Accuracy shows how many predictions are correct compared to the total predictions made. The accuracy formula is shown in equation (6). F1 score is a measure of the balance between precision and recall, especially if the costs of false positives and false negatives are different. The F1 score is shown in equation (7).

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$Accuracy = \frac{TP\,TN}{TP + \ +TN + FP + FN} \tag{6}$$

$$\text{F1-Score} = 2 \text{ x } \frac{(\text{Recall x Precision})}{(\text{Recall+ Precision})} \tag{7}$$

Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

3.4. Optimization process

This optimization model is for tuning parameters, namely optimizing the MLPSentFeat method to make it more optimal, by getting the optimal number of hidden layers and neurons. This optimization model optimizes the number of hidden layers and neurons using a metaheuristic optimization algorithm. Based on this optimization goal, the optimization stage was carried out after the initial classification experiment using MLP with the pre-established initial hidden layer parameters and neuron count to achieve more optimal results. Therefore, the number of hidden layers and neurons was parameter optimized to obtain optimal results with the optimal parameters. Several optimization algorithm parameters are set to obtain the optimal number of hidden layers and neurons, namely population number [108], dimensions [109], iterations [110], hidden layers, and neurons

[70][22]. The metaheuristic optimization models compared are GWO, PSO, CSA, WOA, GOA, BOA, BCO, HS, FA, HHO, CS, BA, SOA, GA, and ABC. Optimization model testing uses recall, accuracy, F1-Score, MCC (Matthew correlation coefficient), and log loss. MCC is suitable for measuring imbalanced and binary [111] and multiclass [112]. The MCC formula is shown in equation (8) [111].

$$MCC = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)}} \quad (8)$$

Algorithm 2: Segment formation and evaluation.

```
Inputs:
D ← Set of sentences
C_cls ← Class labels obtained from the
classification
C_annot ← Class labels from the annotator
Output:
S_cls ← Segments from classification
S_annot ← Segments from the annotator
G ← Greedy similarity matrix
GS ← Greedy similarity value
Step 1: Segment Formation
1. Initialize:
   For classification, set segment_id_cls = 0
   For annotation, set segment_id_annot = 0
   Create empty lists S_cls and S_annot
2. Create Segments (Same Process for Both
   Classification and Annotator):
   For each sentence i in D do:
     For Classification Segments:
       If i == 0, create a new segment
       S_cls(segment_id_cls)
       else
       If Ccls(i) == Ccls(i−1)
         Append sentence i to
         S_cls(segment_id_cls)
       else
         Increment segment_id_cls
         Create new segment
         S_cls(segment_id_cls) and add
         sentence-i
     For Annotator Segments:
       If i == 0, create a new segment
         S_annot(segment_id_annot)
       else
         If Ccls(i) == Ccls(i−1)
           Append sentence-i to
           S_annot(segment_id_annot)
       else
         Increment segment_id_annot
         Create new segment
       S_annot(segment_id_annot) and add
       sentence-i
Step 2: Assign Segment Annotations
1. For each segment Scls(i) in S_cls do:
   Assign segment_id_cls(i)
   Assign classification label Lb(i)
   Assign doctor-patient communication aspect
2. For each segment Sannot(i) in S_annot do:
   Assign segment_id_annot(i)
   Assign label annotator
   La(j)
   Assign doctor-patient communication aspect
Step 3: Segment Evaluation (Greedy Similarity
Calculation)
1. Initialize Greedy Similarity Matrix:
   Create an empty matrix G with dimensions
   |Scls|×|Sannot|
   Set all elements G(i, j) = 0
2. Greedy Similarity Matrix:
   For each segment i in S_cls do:
     For each segment j in S_annot do:
       If Lb(i) ≠ La(j), then:
         G(i, j) = 0
       else:
         The number of identical sentences is
         calculated using equation 9.
Step 4: Compute Final Greedy Similarity Score
1. The highest values were selected from matrix
   G, ensuring that no two selected values
   share the same row or column.
2. Compute the final greedy similarity score
   using equation 10.
```

The architecture of the optimization model, which has two processes, is shown in Figure 5. First, metaheuristic optimization models are compared to find the best model with constant parameters. Second, parameter comparison by increasing the number of parameters in the best metaheuristic model. Optimization model input, testing data, and optimization model parameters, namely, the number of populations, dimensions, iterations, hidden layers, and neurons. The output of the optimization model is the optimal number of hidden layers and neurons. The model will then be applied to other data to test its performance on different data.

3.5. Text segmentation process based on six aspects of communication

The text segmentation stage aims to form segments and annotate segments of doctors' answers from the classification results. The formation of this segment is based on aspects of doctor-patient communication. The input for the text segmentation model is testing data of 20% of labeled data, which contains labels from the classification results and labels from the annotator. The flow of the text segmentation model is shown in Figure 6.

The formation of this segment is based on labels generated from the classification process and labels provided by the annotator. This step evaluates segments by comparing segments from classification labels and annotator labels. The segment formation and segment annotation process combine sentences with the same class next to each other. A situation like this will form a segment. If these conditions are not met, a new segment will be formed. For example, if several sentences have the same class but are not located next to each other, then these sentences are not included in the same segment. Apart from that, if two sentences are located next to each other but do not have the same class, these sentences are not included in the same segment. Algorithm 2 explains the segment formation process, segment annotation, greedy similarity matrix creation, and greedy similarity value calculation. This step was taken by Manchanda [113], who showed that sentences with the same topic are not always in one segment in the data.

One data can consist of one or several segments, and a segment can consist of one or several sentences. This annotation is important because conveying an explanation in the doctor's answer includes certain aspects of communication. Segment formation and segment annotation are done on labels from text classification results, with labels from the annotator.

Segment evaluation aims to measure the similarity of the segments resulting from text classification with the annotator segments. This segment is evaluated using a

greedy similarity model by comparing sentences in the classification result segment with sentences in the annotator segment. The greedy similarity model refers to [114] [3], which measures the similarity of two categories.

Table 4: Results of training and testing data, pseudo-labeling results.

Data Types	Total Data	Total Sentences	F1	F2	F3	F4	F5	F6
Training1 DS1	4328	85374	12947	799	44754	7431	4359	15084
Training2 DS1	4328	84907	12903	782	44622	7442	4363	14795
Training3 DS1	4328	84379	12887	781	44198	7376	4357	14780
Training4 DS1	4328	84786	12915	753	44688	7351	4351	14728
Training5 DS1	4328	85478	12936	793	45034	7480	4354	14881
Testing1 DS1	1082	20857	3200	178	11070	1839	1087	3483
Testing2 DS1	1082	21324	3244	195	11202	1828	1083	3772
Testing3 DS1	1082	21852	3260	196	11626	1894	1089	3787
Testing4 DS1	1082	21445	3232	224	11136	1919	1095	3839
Testing5 DS1	1082	20753	3211	184	10790	1790	1092	3686
Pseudo-Labeled DS1	10031	200724	30363	11	139285	9575	6324	15166
Training1 DS2	6403	88314	13251	236	38533	6562	6401	23331
Training2 DS2	6403	88470	13253	210	38609	6544	6398	23456
Training3 DS2	6400	88564	13228	210	38468	6648	6397	23613
Training4 DS2	6403	88569	13226	226	38591	6629	6406	23491
Training5 DS2	6403	88331	13262	214	38367	6601	6398	23489
Testing1 DS2	1600	22248	3304	38	9609	1684	1599	6014
Testing2 DS2	1600	22092	3302	64	9533	1702	1602	5889
Testing3 DS2	1600	21998	3327	64	9674	1598	1603	5732
Testing4 DS2	1600	21993	3329	48	9551	1617	1594	5854
Testing5 DS2	1600	22231	3293	60	9775	1645	1602	5856
Pseudo-Labeled DS2	28247	384916	48834	74	208943	26824	27066	73175
Training1 DS3	799	18918	1592	52	11887	1443	592	3352
Training2 DS3	799	18412	1586	65	11327	1458	595	3381
Training3 DS3	800	18683	1581	58	11693	1474	591	3286
Training4 DS3	799	18637	1584	56	11511	1476	619	3391
Training5 DS3	799	18418	1601	61	11402	1413	611	3330
Testing1 DS3	200	4349	394	21	2568	373	160	833
Testing2 DS3	200	4855	400	8	3128	358	157	804
Testing3 DS3	199	4584	405	15	2762	342	161	899
Testing4 DS3	200	4630	402	17	2944	340	133	794
Testing5 DS3	200	4849	385	12	3053	403	141	855
Pseudo-Labeled DS3	2111	29783	2560	185	21338	2429	826	2445

The way to fill in the matrix elements is to compare sentences in segments between rows (containing sentences in the classification result segment) and columns (containing sentences in the annotator segment). While comparing these sentences, pay attention to the labels and information of the row sentences and the labels and information of the column sentences. Conditions for filling matrix elements are as follows: 1) If the row segment annotation and column segment annotation are not the same, then the matrix element is filled with a value of zero (0). 2) If the row and column segment annotations are the same, then pay attention to the sentences in the row and column segments. Count the total of sentences in the row and column simultaneously, which is called the correct sentences. A sentence in the classification result segment is said to be correct if it is located in the same segment as the annotator segment. Calculation of the greedy similarity matrix using the formula is shown in equation (9).

$$G(i,j) = \frac{2\ x\ C}{|i| + |j|} \tag{9}$$

Where G is the greedy similarity matrix, C is the number of correct sentences between rows and columns, namely the number of sentences of a segment in a row of the matrix, and j is the number of sentences of a segment in a column of the matrix.

After all elements are filled in, the greedy similarity value can be calculated by 1 selecting the most significant values from the greedy similarity matrix. The most considerable value cannot be located in one row with the same column as another row. 2) Calculate the greedy similarity value using the formula shown in equation (10).

$$GS = \frac{2\ x\ \sum Gmax}{\sum|i| + \sum|j|} \tag{10}$$

Where GS is the greedy similarity value, Gmax is the most considerable, i is the number of sentences of a segment in the matrix row, and j is the number of sentences of a segment in the matrix column.

4 Result and analysis

This section explains the results and analysis of research on text segmentation in online health consultation using MLPSentFeat. These results were obtained based on the method that has been designed. The research results are the pseudo-labeling stage, classification stage, text segmentation stage, and optimization stage.

4.1. Pseudo-labeling

The pseudo-labeling stage aims to label new data using several machine learning models. The pseudo-labeling stage uses two inputs: unlabeled data and labeled data (see Table 3). The training process uses training data to train pseudo-labeling models, which include LR, SVC, DT, KNN, RF, NB, GBM, Xboost, LightGBM, and CatBoost. Each model runs two preprocessing processes: word embedding (IndoBERT) and sentence embedding (SBERT). So, for each data type (DS1, DS2, and DS3), this training model produces 20 labels from each pseudo-labeling model, based on the embedding model. Then, a classifier voting process was carried out on the 20 labels. This classifier voting process takes the highest number of labels to be used as the final label.

The output of the training model is shown in Table 4, which displays data divided using K-fold and rotated five times. This data division is carried out on training data, as much as 80% of the labeled data, and test data, as much as 20% of the labeled data. This division applies to every DS1, DS2, and DS3 dataset. The training and testing dataset amounts are divided almost equally in each dataset.

However, it has a different number of sentences in each fold, which also causes the distribution of sentences in each label to be different. The amount of data from pseudo-labeling results is the same as that of unlabeled data (see Table 3). The resulting pseudo-labeling data is used as training data in the testing model. The results of this pseudo-labeling stage are then used in the classification stage as training data combined with the previously existing training data.

Table 5: Examples of stopwords list for each aspect.

Label	List of Stopword
F1	-
F2	*Mengapa* (why), *berapa* (how many), *apakah* (how), *kapan* (when)
F3	*Berapa* (how many), *beberapa* (several), *berikut* (following)
F4	*Jangan* (do not), *dalam* (in), *umum* (general), *tanpa* (without)
F5	*Tidak* (do not), *jangan* (do not), *tanpa* (without), *ada* (there is)
F6	*Perlu* (need), *tidak* (do not), *jangan* (do not), *perlu* (need), *cukup* (enough)

Table 6: Results of InfoGain for F2 and F5.

Words on F2	InfoGain	Words on F5	InfoGain
Konsultasi (consultation)	0.0406	*Moga* (hopefully)	0.1765
Spesialis (specialist)	0.0279	*Manfaat* (benefits)	0.0755
Air (water)	0.0254	*Bantu* (help)	0.0382
?	0.0242	*Tidak* (no)	0.0324
Darah (blood)	0.0238	*Jangan* (don't)	0.0219
Keluh (complaints)	0.0208	*Ya* (yes)	0.0156
….	….	….	….
Begah (bloated)	0.0001	*Bingung* (confused)	0.0002
Inhaler (inhaler)	0.0001	*Masif* (massive)	0.0002
Bisul (boils)	0.0001	*Signifikan* (significant)	0.0002
Kakak (brother)	0.0001	*Bermafaat* (benefit)	0.0001

Table 7: Results of LR and baseline LR for F2 and F5.

Words on F2	LR	Words on F5	LR
Apakah (what)	1.0000	*Moga* (hopefully)	1.0000
Bagaimana (how)	1.0000	*Informasi* (information)	0.2149
Fresh (fresh)	0.5000	*Bantu* (help)	0.6023
Essence (essence)	0.5000	*Lekas* (soon)	0.7347
Sigmoid (sigmoid)	0.5000	*Sembuh* (get well)	0.1356
?	0.9818	*Paham* (understand)	0.4211
….	….	….	….
Paru(lungs)	0.0105	*Selesai* (finished)	0.02222
Suplemen (supplements)	0.0105	*Tangis* (crying)	0.02174
Sakit (pain)	0.0105	*Ayah* (father)	0.02128
Demam (fever)	0.0104	*Iring* (accompany)	0.02128
Baseline LR	**0.0103**	**Baseline LR**	**0.0205**

Table 8: InfoGain reduction results for F2, F5 label, and testing on MLPSentFeat model.

Ranking	**Features F2**	**InfoGain**	**Accuracy**	**Ranking**	**Features F5**	**InfoGain**	**Accuracy**
-	Full features	-	0.9906	-	Full features	-	0.9911
12	*Alami* (natural)	0.0103	0.9910	6	*Ya* (yes)	0.0156	0.9912
11	*Anak* (children)	0.0104	0.9909	5	*Jangan* (don't)	0.0219	0.9914
10	***Nyeri* (painful)**	**0.0106**	**0.9895**	**4**	***Tidak* (no)**	**0.0324**	**0.9910**
9	*Apakah* (is)	0.0187	0.9903	3	*Bantu* (help)	0.0382	0.9909
8	*Menstruasi* (menstruation)	0.0194	0.9891	2	*Manfaat* (benefits)	0.0755	0.9910
7	*Sakit* (pain)	0.0195	0.9908	1	*Moga* (hopefully)	0.1765	0.9906
6	*Keluh* (complaints)	0.0208	0.9903	-	-	-	-
5	*Darah* (blood)	0.0238	0.9896	-	-	-	-
4	?	0.0242	0.9881	-	-	-	-
3	*Air* (water)	0.0254	0.9906	-	-	-	-
2	*Spesialis* (specialist)	0.0279	0.9902	-	-	-	-
1	*Konsultasi* (consultation)	0.0406	0.9904	-	-	-	-

Table 9: Results of InfoGain calculations on Features1.

Feature	**InfoGain**	**Feature**	**InfoGain**
Dokter (doctor)	0.1773	Obat (medicine)	0.0180
Hindar (avoid)	0.0493	Gejala (symptoms)	0.0136
Makan (eat)	0.0415	Ganggu (disturb)	0.0130
Tidak (not)	0.0324	Nyeri (pain)	0.0106
Minum (drink)	0.0255	Buang (throw)	0.0079
?	0.0243	Perlu (need)	0.0049
Konsumsi (consume)	0.0190	Khawatir (worry)	0.0038
Apakah (what)	0.0187	-	-

Table 10: InfoGain reduction results on Features1 and testing on MLPSentFeat model.

Rank	**Features**	**InfoGain**	**Accuracy**
-	Full features	-	0.9898
15	*Khawatir* (worry)	0.0038	0.9902
14	*Perlu* (need)	0.0049	0.9906
13	***Buang* (throw)**	**0.0079**	**0.9888**
12	*Nyeri* (pain)	0.0106	0.9904
11	*Ganggu* (disturb)	0.0130	0.9893
10	*Gejala* (symptoms)	0.0136	0.9900
9	*Obat* (medicine)	0.0180	0.9885
8	*Apakah* (what)	0.0187	0.9895
7	*Konsumsi* (consume)	0.0190	0.9901
6	?	0.0243	0.9899
5	*Minum* (drink)	0.0255	0.9910
4	*Tidak* (not)	0.0324	0.9908
3	*Makan* (eat)	0.0415	0.9891
2	*Hindar* (avoid)	0.0493	0.9885
1	*Dokter* (doctor)	0.1773	0.9899

4.2. Classification process based on six aspects of communication

This classification aims to classify doctors' answer data based on aspects of doctor-patient communication using the MLPSentFeat model. The classification stage uses input as training data, as much as 80% of labeled data, testing data, as much as 20% of labeled data, and pseudo-labeled data. Combined training data is used for model training. Test data is used for model testing. This classification stage has a preprocessing model for each label, F1, F2, F3, F4, F5, and F6. This preprocessing model for each label assists the MLPSentFeat model in recognizing sentences in each label. This preprocessing model is obtained by arranging the displayed punctuation marks or the words removed from the stopword dictionary into a specific label. The punctuation used is a question mark (?). Examples of the use of stopwords for each label are shown in Table 5. It can be seen that the F1 label does not have a specific stopword, because the MLPSentFeat model can recognize the F1 label. This F1 label uses common stopwords in the stopword removal process. There is a stop word to identify several labels because this stop word is often used on both labels. For example, the word '*tidak'* (do not) refers to labels F5 and F6, and the word '*tanpa'* (without) refers to F4 and F5. Stop words can be determined by looking at words that often appear in a label or based on the type of sentence used in the label. For example, the labels F4, F5, and F6 use the command sentence prohibition, so the word prohibition is excluded from the stopword dictionary. Another example is the F3 label, which uses many of the word '*berikut'* (following), so this word was removed from the stopword dictionary.

4.2.1. Establishment of sentence features

The formation of sentence features was carried out on labels F2 and F5 because these two labels have a small number of sentences. This data condition applies to all DS1, DS2, and DS3 data. Formation of sentence features is done by calculating the LR for each word on the two labels, calculating the baseline LR on each label, selecting words that have a greater LR than the label, calculating the InfoGain for each selected word, and deleting words that have a low InfoGain (see Figure 4 and Algorithm 1). The results of the LR calculation for each word in labels F2 and F5, as well as the baseline LR in both labels, are shown in Table 6. Apart from that, Table 6 also displays words that exceed the baseline LR in labels F2 and F5. The LR value ranges between 0 and 1. A word with an LR value of 1 on a particular label will not appear on another label, and vice versa. For example, the words *'Apakah'* (what) and '*Bagaimana'* (how) have an LR of 1, so these words do not appear in other labels. The baseline LR for label F2 is 0.0103, while the baseline LR for label F5 is 0.0205. Label F2 has 451 unique words, and words with an LR exceeding the baseline LR of 298 words. Label F5 has 223 unique words, and words with an LR exceeding the baseline LR of 88 words. Words that have LR more than the baseline LR will be processed further. Then, these selected words are made into temporary sentence features (called full features) in the MLPSentFeat model to see their performance. After that, record the test results.

Next, InfoGain calculations are done on the selected words in labels F2 and F5. The InfoGain value has a range between 0 and 1. Examples of InfoGain calculation results for each word in labels F2 and F5 are shown in Table 7. Table 7 displays words with InfoGain sorted from largest to smallest value. The word with the most considerable InfoGain value is the most informative word on the label, and conversely, the lower the InfoGain value, the less informative the word is. For example, the word '*konsultasi'* (consultation) has the highest InfoGain of 0.0406 on label F2, indicating that this word is informative on label F2. The word '*moga'* (hopefully) has the highest InfoGain of 0.1765 on the F5 label, indicating that this word is informative in the F5 label.

InfoGain provides an initial picture of the relationship between sentence features and classification targets. Features with high InfoGain have a statistically significant relationship with the target, while features with low InfoGain provide less information. This analysis is important to understand the relevance of each feature in the context of the classification target. Further classification trials are needed to ascertain the contribution of each feature to the overall model performance.

Next, words with a very low InfoGain are deleted, and those with relatively high InfoGain are retained. There are special provisions for determining the limit for deleting these words. A word with a low InfoGain means that the word does not have much influence on the label, so it needs to be deleted. So, one of the tips to stop deleting this word is to see how influential the word is on a label. The words that are not deleted are used as temporary sentence features. Then, this temporary sentence feature is applied in the MLPSentFeat model to see its performance. After that, record the test results. Testing the MLPSentFeat model with features by removing features one by one. Compare the temporary sentence features with the accuracy of full sentence features and observe and find a lower accuracy value than full sentence features for the first time. Temporary sentence features with lower accuracy than full features are the limit for selecting fixed sentence features. So, the selection of fixed sentence features starts from words with the highest InfoGain in a particular label to words with lower accuracy than full sentence features. Please remember that the process of deleting and testing words in the MLPSentFeat model is carried out in stages (one by one) and starts with the word with the lowest InfoGain. Gradual deletion of words may result in a different performance than removing several words immediately, even if the deletion stops with the same word.

For example, deleting words with the lowest InfoGain and the results of testing the MLPSentFeat model with temporary features to obtain fixed sentence features are shown in Table 8. Table 8 shows the temporary sentence features sorted from smallest to largest InfoGain. Then, the accuracy will be observed until the accuracy drops for the first time compared to the model using full features. This process is carried out on labels F2 and F5. Deleting sentence features in label F2, starting from the word with

the lowest InfoGain until after the word 'alami' (natural). This decision was taken because these words have a very low value, namely below 0.01 (keep in mind that there are no special provisions in this determination). So there are 12 words retained in the F2 label. This process is also applied to the F5 label. There are six words retained in the F5 label.

The accuracy of testing the MLPSentFeat model with full sentence features resulted in an accuracy of 0.9898 for label F2 and 0.9911 for label F5. After removing temporary sentence features is carried out in stages, testing with the MLPSentFeat model is carried out on labels F2 and F5, observing the accuracy with full sentence features until the first decrease in accuracy is found, marked in yellow. Label F2 found the first decrease in the word '*nyeri'* (painful), while label F5 found the first decrease in the word '*tidak'* (no). These two words are used as the limit for determining fixed sentence features. So, the fixed sentence features in the F2 label start from the fixed sentence features boundary to the feature with the highest InfoGain, starting with the word '*nyeri'* (painful), which is the feature selection boundary, to '*konsultasi'* (consultation), which has the highest InfoGain. The F2 label has 10 fixed sentence features.

Meanwhile, the fixed sentence features in the F5 label start with the word 'tidak' (no), which is the limit for feature selection, to 'moga' (hopefully), which has the highest InfoGain. The F5 label has four fixed sentence features. The sentence features formed consist of a collection of fixed sentence features labeled F2 and F5, totaling 14 features called Features2.

Features 2 is taken from the words in the labels F2 and F5. Features 2 contains '*nyeri'* (painful), '*apakah'* (is), '*menstruasi'* (menstruation), '*sakit'* (pain), '*keluh'* (complaints), '*darah'* (blood), ?, '*air'* (water), '*spesialis'* (specialist), '*konsultasi'* (consultation), '*tidak'* (no), '*bantu'* (help), '*manfaat'* (benefits), and '*moga'* (hopefully). Apart from Features2, there are also Features1 and Features3. Features1 contains 15 features: '*dokter'* (doctor), '*hindar'* (avoid), '*makan'* (eat), '*tidak'* (no), '*minum'* (drink), ?, '*konsumsi'* (consume), '*apakah'* (what), '*obat'* (medicine), '*gejala'* (symptoms), '*ganggu'* (disturb), '*nyeri'* (pain), '*buang'* (throw), '*perlu'* (need), and '*khawatir'* (worry). Features 1 takes words often appearing in the labels F2, F4, F5, and F6 [5]. The reason for taking sentence features in these four labels and not focusing on labels F2 and F5 is that the MLPSentFeat model is not optimal in classifying these four labels. Features3 is the result of reducing the features of Features1 by two features to 12 features. Features3 contains '*dokter'* (doctor), '*hindar'* (avoid), '*makan'* (eat), '*tidak'* (no), '*minum'* (drink), ?, '*konsumsi'* (consume), '*apakah'* (what), '*obat'* (medicine), '*gejala'* (symptoms), '*ganggu'* (disturb), and '*nyeri'* (pain).

Reducing Features1 to Features3 aims to obtain optimal sentence features.The process of reducing sentence features is the same as the previous process. The InfoGain calculation results for each word from Features1 are displayed in Table 9. Table 9 displays the InfoGain sorted from largest to smallest value. The '*dokter'* (dokter) feature has the highest information gain value of 0.1773, which indicates that this feature provides the most significant information contribution to the classification target.

Next, the MLPSentFeat model was tested by deleting sentence features individually, starting from the lowest InfoGain. Then the accuracy results for each feature are compared with the accuracy of the full features. If it is found that the accuracy of the feature has decreased for the first time, then stop. This feature is used as a limit for selecting full sentence features. The results of testing the MLPSentFeat model by deleting sentence features one by one are shown in Table 10. Full sentence features from Features3 are taken from this limit to the highest InfoGain, starting from the word through, which is the feature selection limit, to the word doctor, which has the highest InfoGain. So, subtracting sentence features from Features1 produces 13 features called Features3.

4.2.2. Experiment MLPSentFeat model

Testing of the MLPSentFeat model is carried out in several types, namely testing three data sets, DS1, DS2, and DS3, with three sentence features, Features1, Features2, and Features3. This test shows the performance of the three features against these three datasets. In addition, the MLPSentFeat model is compared with the MLP and HTCN models. Comparison of the MLPSentFeat model with the MLP because the basis of this research model uses the MLP model to see the performance of the two models in classifying doctors' answer data with certain aspects. A comparison of the MLPSentFeat model with HTCN [39] was carried out because this HTCN model also aims to classify doctors' answer data with certain aspects. So, both of the models have the same goal. The results of various MLPSentFeat tests are shown in Table 11. The MLP model parameters used are hidden_layer_sizes=100, max_iter=1000, activation='relu', solver='adam', random_state=42, warm_start=True, and learning_rate=0.001. The HTCN model parameters used are channels=[64, 64], kernel_size=2, dropout=0.2, learning_rate=0.001, epochs=10, and batch_size=1.

Researcher Juanita [87] also identified doctors' answer data from the Alodokter platform (DS1) using the Multi-Label Classification (MLC) model. This model was not tested in the study because it is for multi-label classification, while this study uses multiclass classification. The F1 score results of the MLC model with various combinations of methods are around 0.97. while the average F1 score results of the MLPSentFeat model for DS1 with various sentence features are between 0.98 and 0.99. The MLPSentFeat model can outperform the MLC model by 1 or 2 points.

Table 11 presents the performance of several models across different datasets. The comparative analysis focuses on the baseline model (MLP) and the proposed model (MLPSentFeat), particularly related to labels F2 and F5. These two labels are the focus of the study due to

Table 11: Experiment MLPSentFeat model.

Models	F1-Score							Accuracy
	F1	F2	F3	F4	F5	F6	AVG	
MLP+DS1	0.9919	0.8660	0.9826	0.9806	0.9931	0.9673	0.9636	0.9808
MLP+DS2	0.9946	0.8750	0.9785	0.9835	0.9991	0.9712	0.9670	0.9806
MLP+DS3	0.9819	0.9272	0.9834	0.9840	0.9656	0.9580	0.9667	0.9780
HTCN+DS1	0.9938	0.9657	0.9605	0.9020	0.9913	0.9189	0.9554	0.9552
HTCN+DS2	0.9936	0.8678	0.9438	0.9308	0.9978	0.9180	0.9420	0.9474
HTCN+DS3	0.9864	0.9605	0.9859	0.9813	0.9427	0.9631	0.9700	0.9800
MLPSentFeat+DS1+Features1	0.9954	0.9933	0.9875	0.9773	0.9967	0.9760	0.9877	0.9864
MLPSentFeat+DS1+Features2	0.9957	0.9964	0.9882	0.9848	0.9970	0.9741	0.9894	0.9872
MLPSentFeat+DS1+Features3	0.9952	0.9969	0.9928	0.9916	0.9920	0.9857	0.9924	0.9918
MLPSentFeat+DS2+Features1	0.9946	0.9833	0.9906	0.9918	0.9989	0.9890	0.9914	0.9914
MLPSentFeat+DS2+Features2	**0.9946**	**0.9871**	**0.9911**	**0.9928**	**0.9992**	**0.9899**	**0.9925**	**0.9920**
MLPSentFeat+DS2+Features3	0.9946	0.9757	0.9904	0.9885	0.9992	0.9898	0.9897	0.9913
MLPSentFeat+DS3+Features1	0.9903	0.9861	0.9905	0.9898	0.9717	0.9757	0.9840	0.9872
MLPSentFeat+DS3+Features2	0.9908	0.9861	0.9909	0.9890	0.9604	0.9782	0.9826	0.9875
MLPSentFeat+DS3+Features3	0.9903	0.9861	0.9911	0.9877	0.9672	0.9788	0.9835	0.9878

the relatively limited number of sentences representing these aspects. At the same time, the MLPSentFeat model was designed to classify such sentences despite the scarcity of data in those categories. The results show that the MLP model tends to produce lower F1-scores for label F2 compared to the MLPSentFeat model in several combinations of datasets and features. The MLPSentFeat model is more effective in classifying label F2, demonstrating better and more consistent outcomes. Conversely, for label F5, the MLP model performs better in some datasets, with very stable F1-scores. MLPSentFeat still shows competitive performance for this label, although it does not always outperform the MLP model.

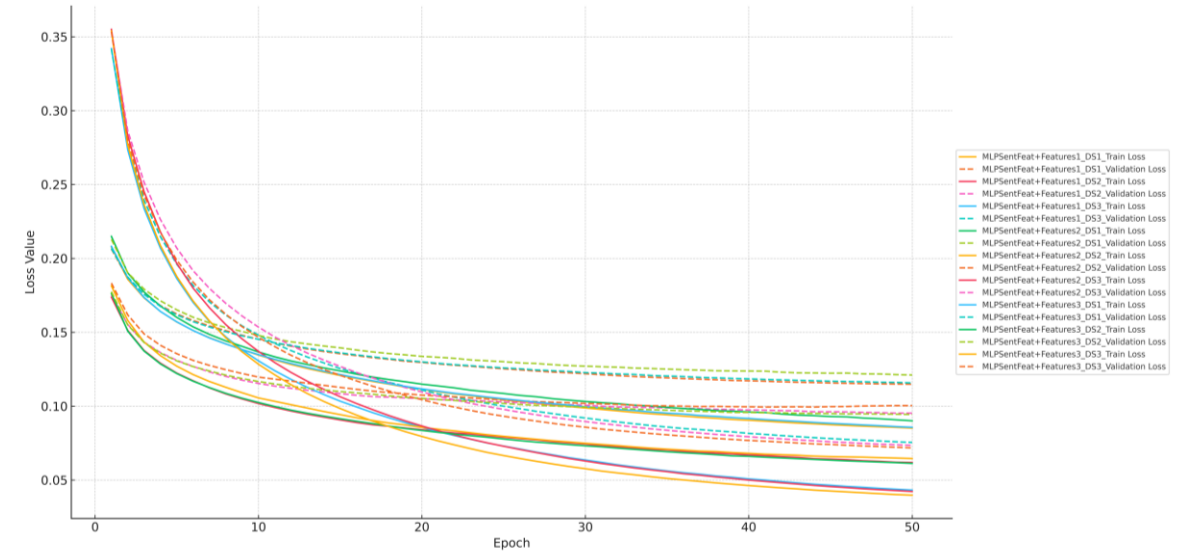
As previously described, the sentence features used in the model were extracted based on the lexical characteristics of labels F2 and F5. However, these features may also influence classification performance on other labels, as certain words appear in F2 and F5 and other communication-related categories. This overlap may result in side effects on cross-label classification performance. Overall, the MLPSentFeat model performs better in handling labels F2 and F5. More specifically, the combination of dataset DS2 with various sentence features in the MLPSentFeat model shows fluctuating F1-scores for label F2, compared to other data-feature combinations that tend to be more stable. For instance, the MLPSentFeat+DS2+Features3 combination yields the lowest F1-score for label F2 at 0.9757—lower than DS3, despite DS3 having significantly less data than DS2.

This finding suggests that the observed fluctuation is due to the quality of interrogative sentence construction in label F2 of dataset DS2. Many questions in this label are written in informal formats, often starting with interjections or introductory expressions commonly used in spoken communication. Such constructions can confuse the model, as the sentence structure resembles a statement rather than a question. In contrast, the F2 sentences in DS1 and DS3 are more structurally regular and conform to standard interrogative conventions, resulting in more stable model performance on these datasets.

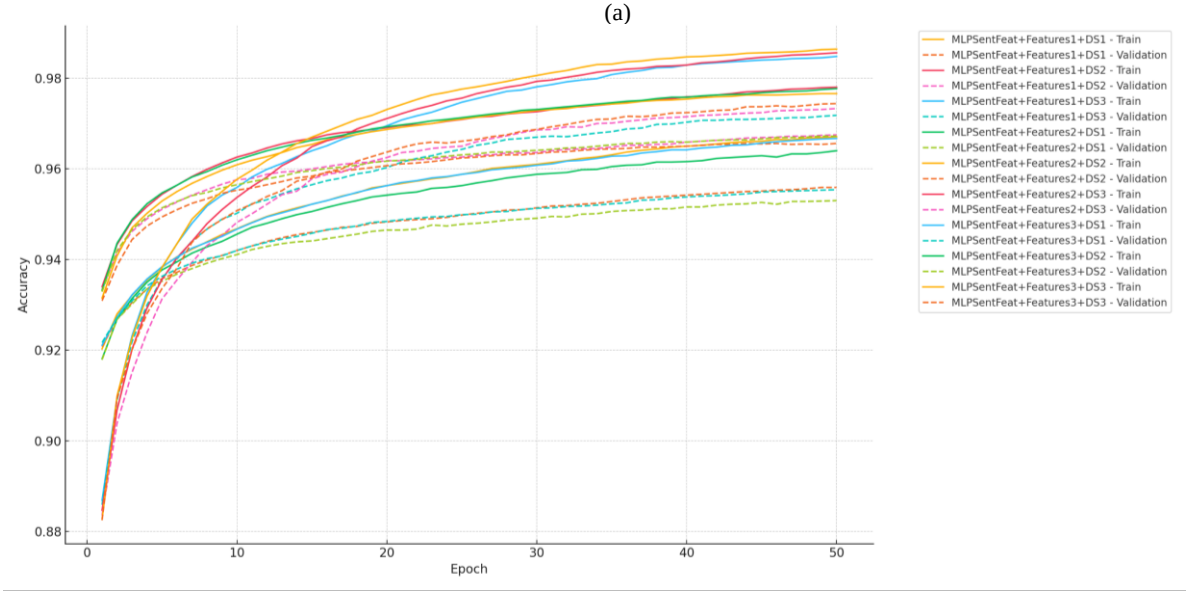
The MLPSentFeat model also shows stable performance on label F5, comprising sentences expressing hope or wishes. These sentences follow consistent lexical patterns that the model more easily recognizes. Furthermore, expressions of hope often carry positive or neutral sentiment, aligning well with the sentiment-based features employed in the model. On the other hand, interrogative sentences under label F2 are more difficult to identify due to their flexible and varied structure, which does not always begin with an explicit question word and may, in some cases, resemble declarative statements. Structural variation, the use of interjections at the beginning of sentences, and their emotionally neutral nature all present challenges for the model in consistently distinguishing questions.

The strength of the MLPSentFeat model in handling labels F2 and F5 demonstrates that the integration of explicitly designed sentence features within the model architecture significantly contributes to its sensitivity to linguistic structure variation. It indicates that a sentence feature–based approach can enhance the model's semantic representation capabilities, particularly for sentences that are uncommon or written in non-standard formats. Practically, the model's success in identifying labels F2 and F5 is highly significant, as these aspects play a crucial role in establishing effective two-way communication between doctors and patients—in the form of exploratory questions and empathetic expressions. Therefore, enhancing the model's ability to recognize these two aspects may serve as a foundation for developing quality evaluation systems for online medical consultations.

Although the MLPSentFeat model has demonstrated promising performance, it remains a challenge, especially in identifying label F2, encompassing interrogative sentences with highly variable structures that do not always adhere to formal writing conventions. This challenge opens avenues for further research, such as developing syntactic augmentation techniques, implementing sentence structure–based representations, and improving specialized text normalization procedures that can significantly affect model performance. At present, specialized text normalization has proven to make a meaningful contribution, particularly for label F2, as indicated by the stable F1-score in DS1 compared to DS2. Meanwhile, DS3, which did not undergo such



(a)



(b)

Figure 7: a) Training and validation loss and b) training and validation accuracy of the MLPSentFeat model on various datasets and sentence features.

normalization, still shows stable performance due to the inherently well-structured interrogative sentences in its F2 category, which adhere to standard linguistic conventions.

An analysis of Table 11 from the perspective of overall model comparison reveals that the MLPSentFeat model consistently demonstrates the best performance across all datasets and feature variations. The highest achievement is observed in the combination of MLPSentFeat+DS2+Features2, which yields an average F1-score of 0.9925—the highest score recorded in the entire set of experiments. This finding indicates that incorporating relevant sentence features can significantly enhance the model's ability to recognize key patterns involving doctors' responses based on communication aspects in the classification task. Furthermore, including specific feature variations, particularly Features2 and Features3, contributes substantially to improving model performance, as evidenced by consistently high F1-scores across various datasets. This inclusion suggests that the integration of appropriate sentence-level features enables the model to better adapt to differences in data characteristics. In contrast, the HTCN method demonstrates the lowest and least stable performance, with the lowest average F1-score of 0.9420 recorded in the HTCN+DS2 combination. It underscores that HTCN is less capable of adapting to the characteristics of the datasets used in this study, resulting in overall lower performance compared to the other models—especially when compared to MLPSentFeat, which excels in leveraging additional sentence-level features.

The analysis of Table 11 in terms of dataset size and text normalization indicates that the performance of text classification models is not solely determined by the volume of data, but is also significantly influenced by the text normalization processes applied to the data. The three datasets used—DS1, DS2, and DS3—exhibit different characteristics in terms of both data volume and the cleanliness and consistency of textual content. DS1, which

contains a moderate amount of data and has undergone a specific text normalization procedure, demonstrates highly stable performance across all labels. The combination of the MLPSentFeat model with Features3 on DS1 yields an average F1-score of 0.9924, with F1-scores for all labels (F1–F6) consistently above 0.99. Notably, label F2—known to be sensitive to semantic variation—achieves the highest F1-score across all configurations, reaching 0.9969. This finding suggests that proper text normalization can enhance the consistency and quality of sentence feature representations, regardless of dataset size. In contrast, DS2 represents the largest dataset in terms of volume but does not undergo specific normalization. Its best-performing combination (MLPSentFeat+Features2) achieves the highest average F1-score of 0.9925. However, label-level stability is less consistent; the F1-score for label F2 in this combination is only 0.9871, which is lower than the corresponding score in DS1. It implies that while a large dataset can improve overall performance, without adequate handling of noise and linguistic variation, the model may remain vulnerable to decreased performance on certain labels. Meanwhile, DS3, the smallest dataset in data size, presents relatively clean text and does not require special normalization. Despite having a lower average F1-score than DS1 and DS2 (maximum 0.9835), the model maintains stable F1-scores across labels. The F1-score for label F2 in the MLPSentFeat+Features3 combination reaches 0.9861, indicating that well-structured text can contribute to stable model performance even with limited data. Overall, these experimental findings highlight that text normalization and structural data quality play a crucial role in improving the accuracy and stability of text classification models, and can compensate for limitations in data quantity. A larger dataset does not necessarily lead to superior performance if not accompanied by adequate text normalization, particularly in classification tasks that require sensitivity to semantic structure and variation in expressive sentence forms.

The analysis of Table 11 from the perspective of sentence feature utilization shows that three types of features—Features1, Features2, and Features3—were tested to evaluate their contributions to the performance of the MLPSentFeat-based classification model. The experimental results indicate that no single sentence feature consistently outperformed the others in all conditions, as the effectiveness of each feature is highly influenced by the characteristics of the data and the accompanying text processing applied. Features1, which was constructed based on the frequency of words predominantly appearing in labels F2 and F5, demonstrated relatively stable performance but did not dominate. This feature generally yielded good F1-scores across all three datasets but was never part of the combination that achieved the highest average F1-score. These findings suggest that although Features1 may contribute to sentence representation in general, its capacity to capture semantic variation and expressive structures remains limited. In contrast, Features2—generated through a LR approach and feature selection using Information Gain—achieved the highest average F1-score of 0.9925 when applied to the DS2 dataset. However, in this combination, the F1-score for label F2 decreased to 0.9871, indicating that while this feature effectively captures global semantic context, maintaining performance stability for expressive or minor labels remains challenging. This limitation is compounded by the nature of DS2, which contains a large volume of data but lacks text normalization, making the feature more susceptible to the effects of noise and imbalanced label distribution. On the other hand, Features3, which was derived by reducing Features1 using Information Gain, demonstrated the most stable performance across labels, particularly when combined with datasets DS1 and DS3. The combination of MLPSentFeat with DS1 and Features3 resulted in F1-scores exceeding 0.99 across all labels, including label F2, which achieved the highest score among all configurations at 0.9969. This feature was designed to capture sentence structures related to pragmatic functions, such as interrogative expressions, expressions of hope, and other communicative forms. The performance stability achieved through Features3 was especially evident in datasets that had undergone specific normalization (DS1) or those that naturally contained well-structured sentences (DS3), suggesting that the effectiveness of this feature strongly depends on the cleanliness and consistency of the data to reach optimal performance.

Figure 7 presents the training and validation loss curves (Figure 7a) and the training and validation accuracy curves (Figure 7b) for the MLPSentFeat model across various combinations of datasets and sentence features. These visualizations provide insights into the model's learning behavior throughout 50 training epochs and are used to identify convergence patterns, generalization capability, and potential overfitting. Each color in the graphs represents a unique combination of sentence feature type (Features1, Features2, or Features3) and dataset (DS1, DS2, or DS3), while line styles distinguish between training data (solid lines) and validation data (dashed lines). In general, most combinations exhibit a decreasing trend in loss and an increasing trend in accuracy as the number of epochs increases, indicating an effective learning process. However, the degree of alignment between the training and validation curves is a key indicator in assessing whether the model is overfitting.

The combinations that exhibited symptoms of overfitting were MLPSentFeat+DS1+Features2, MLPSentFeat+DS2+Features2, and MLPSentFeat+DS2 +Features3. These three configurations shared a common pattern: the training loss decreased sharply, while the validation loss remained stagnant or increased after several epochs. Additionally, a significant gap between training and validation accuracy was observed, indicating that the model had overly adapted to the training data but failed to generalize effectively to the validation set. In the case of MLPSentFeat+DS2+Features2, overfitting occurred despite the large dataset size and the highest average F1 score among all combinations. It was primarily due to the absence of specialized text normalization in DS2, which caused the model to learn from inconsistent

Table 12: Results of various classification experiments DataID #3422.

Order	Sentences	1*	2**	3***	4****	5*****	6******
0	Halo (Hello)	F1	F1	F1	F1	F1	F1
1	Terima kasih telah menggunakan layanan konsultasi Alodokter (Thank you for using Alodokter consultation services)	F1	F1	F1	F1	F1	F1
….	….	….	….	….	….	….	….
13	Apakah terdapat keluhan lain yang menyertai? (Are there any other accompanying complaints?)	F2	F3	F2	F2	F2	F2
14	Apakah mual? (Is it nausea?)	F2	F3	F2	F2	F2	F2
15	Apakah muntah? (Is it vomiting?)	F2	F3	F2	F2	F2	F2
16	Apakah perut kembung? (Is your stomach bloated?)	F2	F3	F2	F2	F2	F2
17	Apakah diare? (What is diarrhea?)	F2	F3	F2	F2	F2	F2
18	Apakah konstipasi? (What is constipation?)	F2	F3	F2	F2	F2	F2
19	Apakah BAB (Buang Air Besar) berdarah? (Is your bowel movement bloody?)	F2	F3	F2	F2	F2	F2
….	….	….	….	….	….	….	….
24	Beri nutrisi yang bergizi dan sehat (Give it nutritious and healthy nutrition)	F6	F6	F3	F6	F6	F6
25	Konsumsi air putih (Drink water)	F6	F6	F6	F6	F6	F6
….	….	….	….	….	….	….	….
29	Semoga membantu (I hope this helps)	F5	F5	F5	F5	F5	F5

1: Annotator labels. 2: MLP labels. 3: MLPSentFeat+Features1 labels. 4: MLPSentFeat+Features2 labels. 5: MLPSentFeat+Features3 labels. 6: MLP Optimization Labels.

Table 13: Hyperparameter optimization with various optimization algorithms and MLP method.

Optimization	F1 score							Neuron	Log Loss	MCC	Accuracy
	F1	F2	F3	F4	F5	F6	AVG				
MLP+PSO	1.00	1.00	0.99	0.97	0.99	0.98	0.99	91	0.06	0.98	0.99
MLP+BOA	1.00	1.00	0.99	0.97	0.99	0.98	0.99	100	0.07	0.98	0.99
MLP+GWO	**1.00**	**1.00**	**0.99**	**0.97**	**0.99**	**0.98**	**0.99**	**91**	**0.06**	**0.99**	**0.99**
MLP+SOA	1.00	1.00	0.99	0.97	0.99	0.98	0.99	91	0.06	0.98	0.99
MLP+BA	1.00	1.00	0.99	0.97	0.99	0.98	0.99	91	0.06	0.98	0.99
MLP+FA	1.00	1.00	0.99	0.97	0.99	0.98	0.99	77	0.07	0.98	0.99
MLP+GA	1.00	1.00	0.99	0.97	0.99	0.98	0.99	68	0.07	0.98	0.99
MLP+GOA	1.00	1.00	0.99	0.97	0.99	0.98	0.99	100	0.07	0.98	0.99
MLP+WOA	0.99	1.00	0.99	0.97	0.99	0.98	0.99	100	0.08	0.98	0.99
MLP+ABC	1.00	1.00	0.99	0.97	0.99	0.95	0.98	98	0.09	0.97	0.99

words and sentence structures. Features2, constructed using LR and feature selection based on InfoGain, tended to capture dominant word frequency patterns. In noisy data, this characteristic could lead to learning bias. This combination demonstrated the most pronounced gap between the training and validation curves, making it the most severe case of overfitting observed in the study. Meanwhile, the combination of MLPSentFeat+DS1+Features2 experiences overfitting because Features2 is more suitable for data with high variation and a large amount of data, as seen in the DS2 dataset. Given the features' highly selective nature, the model could not handle expressive variations present in the validation set. A similar situation was observed in MLPSentFeat+DS2+Features3, where the pragmatic sentence structures intended to be captured by the features did not align well with the varied and unnormalized nature of DS2. As a result, the model struggled to develop a generalized representation.

Conversely, the remaining six combinations did not exhibit signs of overfitting. The MLPSentFeat+DS1+Features3 configuration was the most stable, showing highly aligned training and validation loss and accuracy curves. This configuration indicates strong generalization ability, supported by the quality of the DS1 dataset, which underwent text normalization, and the high relevance of Features3 to the pragmatic structure of sentences. Other combinations, such as MLPSentFeat+DS3+Features3 and MLPSentFeat+DS3+Features2, also demonstrated stability despite being derived from a smaller dataset. It suggests that the cleanliness and consistency of sentence structures in DS3 contributed significantly to the model's stable performance. Combinations involving Features1 across all datasets also showed reasonably stable performance without overfitting, although they did not achieve top scores. It can be attributed to the general and less selective nature of Features1, which prevented the model from overfitting to specific patterns in the training

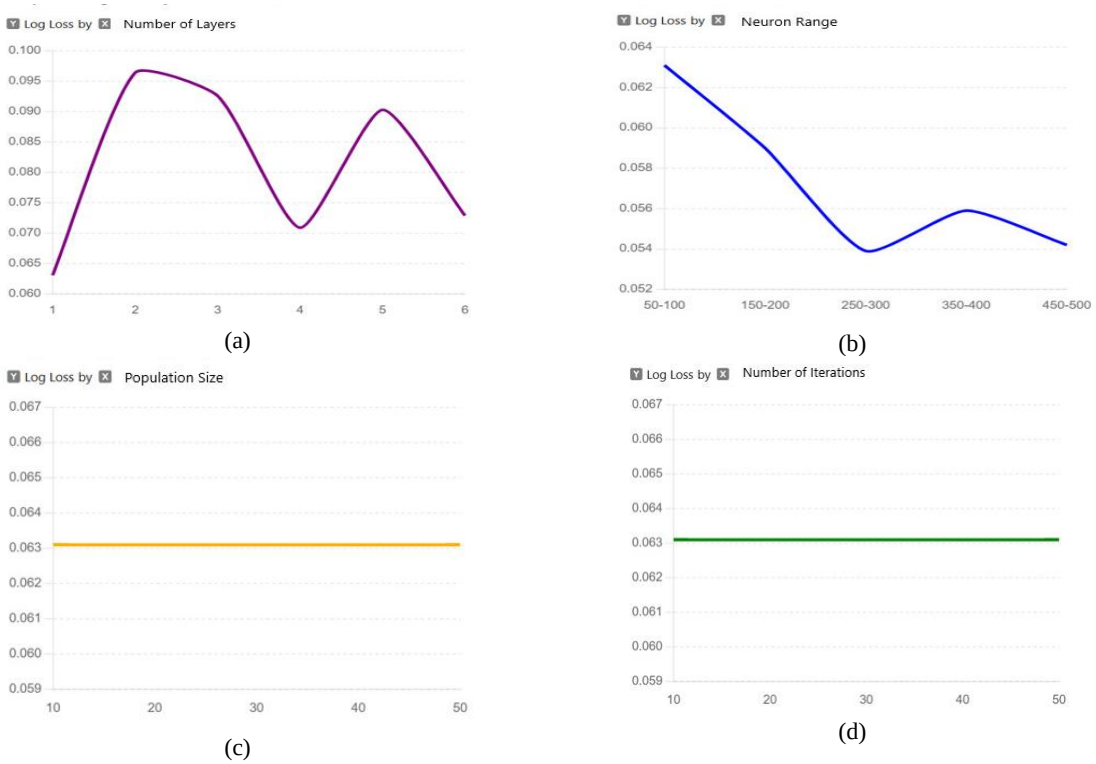


Figure 8: Log loss graph of GWO and MLP with various parameters. (a) Log loss parameter number of hidden layers. (b) Log loss parameter neuron range. (c) Log loss parameter number of populations. (d) Log loss parameter number of iterations.

Table 14: GWO and MLP parameter optimization.

Hyperparameter Setting	Parameters	F1 score							Neurons	Log Loss	MCC
		F1	F2	F3	F4	F5	F6	AVG			
Population=10, HL*=4, Iter**=10, Neurons= 50-100, dim***=4	Hidden Layer	1.00	1.00	0.99	0.98	1.00	0.98	0.99	76, 93, 69, 60	0.07	0.98
Population=10, HL=1, Iter=10, Neurons= 250-300, dim=1	**Neurons**	**0.99**	**1.00**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**283**	**0.05**	**0.99**
Population=50, HL=1, Iter=10, Neurons= 50-100, dim= 1	Population	1.00	1.00	0.99	0.97	0.99	0.98	0.99	91	0.06	0.98
Population=10, HL=1, Iter=50, Neurons= 50-100, dim= 1	Iteration	1.00	1.00	0.99	0.97	0.99	0.98	0.99	91	0.06	0.98

*HL=Hidden Layer. **Iter=Iteration. ***dim=Dimension.

data. These findings reinforce the notion that the presence or absence of overfitting is highly dependent on the interaction between dataset characteristics, the method of sentence feature extraction, and model capacity. Text normalization processes, dataset size and cleanliness, and the alignment between features and sentence semantics are critical factors in maintaining model stability and avoiding overfitting.

An example of the results of classifying doctors' answers based on communication aspects is shown in Table 12. The example shown uses data ID_Data 3422 from DS1. The models shown in this example are only the MLP and MLPSentFeat models because they are used to display the classification results of the MLP and MLPSentFeat models, considering that the performance of the MLPSentFeat model is increased from the MLP model. In addition, the MLPSentFeat model can classify F2 labels better than the MLP model (based on the accuracy in Table 11). These two classification results must be displayed because the MLPSentFeat model is a development of the MLP model.

The reference label used is the annotator label, which is given directly by the doctor. The colored sections in the table highlight errors in the classification results. The green color indicates an error in the classification results by the MLP model, so the MLP model has difficulty identifying sentences with label F2 in sentences 13-19. However, the MLPSentFeat model (with various sentence features and model optimization) can classify these sentences into label F2. The similarity of the annotator labels with the MLPSentFeat model labels and model optimization proves this. Meanwhile, the blue color

(sentence 24) shows an error in the classification results by the MLPSentFeat model with Features1 (column 3), but the classification results are correct in other types of features. The results of this classification cannot ensure that the sentence features that cause errors in classification are bad. The results of this classification will be processed into the text segmentation and model optimization stages.

4.3. Optimization process

This optimization stage has two processes, namely hyperparameter optimization and application of the parameters that have been obtained (explained in Figure 6). Hyperparameter optimization to find the best configuration through trials using several optimization algorithms. The hyperparameters that have been formed are applied to various data (DS1, DS2, and DS3) and sentence features (Features1, Features2, and Features3). The classification results using optimal parameters were also analyzed in terms of the performance of the MLPSentFeat model, data overfitting, and the segments formed.

4.3.1. Hyperparameter optimization

The best configuration desired from this parameter optimization process is to obtain the optimal number of hidden layers and neurons for the MLP method. The process carried out in the hyperparameter optimization process first compares several metaheuristic optimization algorithms, including PSO, BOA, GWO, SOA, BA, FA, GA, GOA, WOA, and ABC. This optimization algorithm is run simultaneously with the MLP method, because this optimization algorithm functions to optimize the MLP method. This comparison of optimization algorithms is to obtain the best algorithm for optimizing the MLP method applied to doctors' answer data. The best optimization algorithm is determined by looking at the F1 score or the highest accuracy that exceeds the performance of the MLPSentFeat model. If several optimization algorithms have the same F1 score or accuracy value, look at the log loss value and choose the smallest value. If several optimization algorithms have the same log loss value, consider the MCC value and choose the highest value. Alternatively, directly choose the lowest log loss value. This provision was taken because this research used accuracy and F1 score metrics.

However, during optimization experiments, it turned out that these two metrics were not enough to determine the best algorithm. Second, the value of the initial parameters used in the best optimization algorithm should be increased. Then, the performance between parameter values is compared. The parameters set in the optimization algorithm to obtain the optimal number of hidden layers and neurons are the number of populations, dimensions, iterations, hidden layers, and neurons. The initial parameter settings are population=10, hidden layer=1, iteration=10, neurons=50-100, dimension=1. Each parameter value is increased 5 times. Each parameter value and type is compared, and the smallest log loss value is selected. A model with the smallest log loss value also has a high F1 score or accuracy. The parameter with the smallest log loss value is the optimal parameter, which is the parameter value used by that parameter.

The results of comparative experiments between optimization algorithms are shown in Table 13. The optimization model parameters include population=10, hidden layer=1, iteration=10, neurons=50-100, and dimension=1. MLP parameter settings include max_iter=1000, activation='relu', solver='adam', and random_state=42. Other MLP parameters have been determined: one input layer with 399 neurons, one hidden layer with 100 neurons, and one output layer with six neurons. Due to limited available resources, this hyperparameter optimization experiment was implemented on DS1 data, where training and test data had not been separated using K-fold (explained in Table 4).

The results of the optimization and MLP experiments in Figure 8 show that almost all optimization models produce an average F1 score and accuracy that is the same as the F1 score percentage of the MLPSentFeat model of 0.99, or even reduced. If only these two metrics were used, the best optimization model could not be chosen. Therefore, model selection requires metrics like log loss and MCC values. The first analysis looks at the smallest log loss value, namely 0.06, which is owned by three models, namely GWO, SOA, and BA. Because there are three of the best models, the analysis continues by paying attention to the MCC metrics of these three models. The best optimization model was chosen based on the highest MCC value, namely the GWO optimization model with 91 neurons 91.

After finding the best optimization algorithm, namely GWO, the algorithm was tested again by increasing the parameter values of each type of parameter. This parameter value was increased 5 times for each parameter. The experimental results of the performance of each parameter of GWO are shown in Figure 8. This figure shows the experimental results of the GWO algorithm, which displays log loss based on changes in the parameters of the number of hidden layers, neuron range, population size, and iteration size. An explanation of the initial GWO parameters has been provided previously. After that, each parameter value is increased gradually. The parameter for the number of hidden layers has the values 1, 2, 3, 4, 5, 6 (a). Setting the number of dimensions parameters is done together with the hidden layer parameters with values 1, 2, 3, 4, 5, and 6. The dimension parameters function to optimize the number of neurons in each layer [115]. This research uses dimension parameters to optimize the number of neurons in the hidden layer. The neuron range parameters are 50-100, 150-200, 250-300, 350-400, 450-500 (b). The population number parameter has the values 10, 20, 30, 40, 50 (c). The number of iterations parameter values 10, 20, 30, 40, 50 (d).

Figure 8(a) shows the relationship between the number of hidden layers and the log loss value. Log loss increases until it reaches a maximum value of 0.097 when two hidden layers exist. A significant decrease occurred when the number of hidden layers increased to four, with a minimum log loss value of 0.067. An increase occurred again when the hidden layers increased to five before experiencing a slight decrease when the hidden layers

Table 15: Experiment on applying MLP parameter optimization to the MLPSentFeat model.

Model	F1	F2	F3	F4	F5	F6	AVG	Accuracy
MLPSentFeat+DS1+Features1	**0.9954**	**0.9979**	**0.9923**	**0.9869**	**0.9966**	**0.9862**	**0.9926**	**0.9915**
MLPSentFeat+DS1+Features2	0.9953	0.9979	0.9895	0.9868	0.9901	0.9795	0.9899	0.9885
MLPSentFeat+DS1+Features3	0.9948	0.9985	0.9898	0.9899	0.9946	0.9779	0.9909	0.9888
MLPSentFeat+DS2+Features1	0.9946	0.9852	0.9918	0.9936	0.9992	0.9908	0.9925	0.9926
MLPSentFeat+DS2+Features2	0.9946	0.9776	0.9898	0.9901	0.9992	0.9884	0.9900	0.9908
MLPSentFeat+DS2+Features3	0.9946	0.9680	0.9921	0.9933	0.9992	0.9915	0.9898	0.9929
MLPSentFeat+DS3+Features1	0.9908	0.9861	0.9909	0.9863	0.9715	0.9771	0.9838	0.9874
MLPSentFeat+DS3+Features2	0.9903	0.9861	0.9919	0.9890	0.9748	0.9802	0.9854	0.9889
MLPSentFeat+DS3+Features3	0.9908	0.9861	0.9906	0.9884	0.9687	0.9762	0.9835	0.9871

numbered six. This pattern shows that a larger number of hidden layers does not continually improve model performance, but can cause overfitting or increased noise in learning. The best experimental results with parameter settings for the number of hidden layers are shown in Table 14.

Figure 8(b) illustrates the effect of the number of neurons on the log loss value. Log loss experienced a decreasing trend as the number of neurons increased until it reached the range of 250–300 neurons, with the lowest log loss value being 0.054. An increase in the number of neurons after passing this number causes slight fluctuations in the log loss. Too few neurons increase log loss, while too many neurons do not provide significant improvement. The 250–300 neurons range produces the lowest log loss and balances model complexity and good generalization. The best experimental results with parameter settings for the number of neurons are shown in Table 14.

Figure 8(c) visualizes the relationship between the optimization algorithm's population size and the log loss value. Varying population size, ranging from 10 to 50, did not significantly change log loss, which remained stable at around 0.063. Population size is not the main factor influencing model performance. Increasing the population size does not necessarily improve the optimization quality, as the model has reached stable performance with smaller population sizes. The best experimental results with population number parameter settings are shown in Table 14.

Figure 8(d) shows the relationship between the number of iterations and log loss. Increasing the number of iterations does not significantly impact log loss, remaining constant at around 0.063. The model has reached a convergence point before increasing the number of iterations. Additional iterations only increase computing time without providing significant performance improvements. The best experimental results with parameter settings for the number of iterations are shown in Table 14.

The results in Table 14 show that the best parameter to optimize is the number of neurons, with a configuration of 283 neurons producing the lowest log loss of 0.05. This value is smaller than the hidden layer, population, and other iteration parameters, which have a higher log loss. This configuration also produces an average F1 score of 0.99, showing that the model has excellent prediction performance and a minimal error rate.

4.3.2. Parameter application to the MLPSentFeat model

The hyperparameters used are the number of hidden layers of 1 and neurons of 283. These parameters are applied to the MLPSentFeat model with various datasets (DS1, DS2, and DS3) and sentence features (Features1, Features2, and Features3). The results of applying parameter optimization to the MLPSentFeat model with various datasets and sentence features are shown in Table 15.

The results of the experiment presented in Table 15 demonstrate that the application of parameter optimization to the MLP model positively impacts the MLPSentFeat model's performance in classifying doctor responses based on communication aspects. It is evident that the model optimization successfully increased the number of models with an average F1-score greater than 0.99, which doubled to four models achieving an average F1-score above 0.99, namely MLPSentFeat+DS1+Features1 (0.9926), MLPSentFeat+DS1+Features2 (0.9899), MLPSentFeat+ DS1+Features3 (0.9909), and MLPSentFeat+DS2+ Features1 (0.9925). This improvement indicates that the optimization enhances the model's accuracy and improves its performance's consistency and stability, as reflected in the stable F1-scores across most model combinations. However, despite performance improvements in some models, not all models showed an increase in average F1-score or accuracy after optimization. Sometimes, although optimization aims to improve overall performance, the process can lead to a decrease in performance for certain labels that are more varied or complex in structure. For example, in label F2 with the DS2 dataset, the F1-score decreased after optimization when combining Features2 and Features3. This decline can be explained by the model's inability to handle the diversity of sentence structures in label F2, which contains many interrogative sentences—this sentence type is more difficult for the model to recognize. While optimization improves overall performance, performance decreases for labels with higher expressive variation, such as label F2. It impacts the average F1-score of MLPSentFeat+DS2+Features2, the best model before optimization, with an average F1-score of 0.9925. However, after optimization, this model experienced a decrease in the F1-score for label F2 by 0.0095, decreasing the overall average F1-score of this model to 0.9900.

The performance of sentence features in the dataset after optimization remains consistent with the state before optimization, where Features2 is still suitable for data with

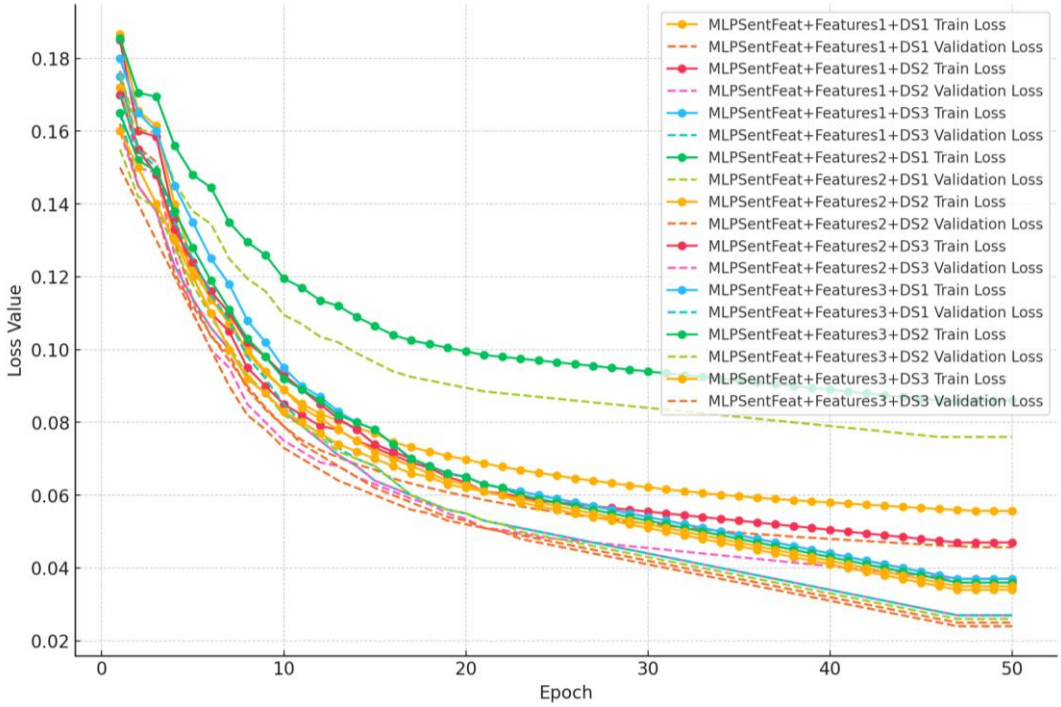


Figure 9: Training and validation loss from applying MLP parameter optimization to the MLPSentFeat model.

high variability (such as DS2), Features3 is more suitable for datasets that are well-structured and clean (such as DS1 and DS3), and Features1 shows better stability across the entire dataset. An important finding in this experiment is the improvement in Features1's performance after optimization. Features1, which initially did not provide the best performance, managed to become the best model in this experiment, with an average F1-score of 0.9926 in the combination of MLPSentFeat+DS1+Features1, which demonstrates excellent stability, as the F1-score across all labels tends to be more consistent compared to other features. Meanwhile, Features2 and Features3, which performed better before optimization, remained stable despite a slight decrease in F1-score after optimization.

Figure 9 illustrates the training and validation loss curves of the MLPSentFeat model after optimization, applied to various combinations of datasets and sentence features. Four models exhibited overfitting: MLPSentFeat+DS1+Features2, MLPSentFeat+DS2+Features2, MLPSentFeat+DS2+Features3, and MLPSentFeat+DS3+Features2. These models showed a sharp decline in training loss, while the validation loss stagnated or slightly increased post-optimization. The Features2 feature set experienced overfitting across multiple datasets due to its high selectivity in choosing the most informative features. As a result, the model became overly reliant on specific patterns (e.g., dominant words) and struggled to handle more diverse sentence structures. Although Features2 is well-suited for datasets with high frequency, such as DS2, it lacks the flexibility to manage more structured datasets, such as DS1 and DS3, leading to overfitting. In contrast, MLPSentFeat+DS2+Features3 continued to experience overfitting after optimization because the optimization did not fully address the structural issues in the data. Features3 could not adequately handle the highly variable data in DS2, which contained numerous sentences with diverse structures. Despite optimization improving certain aspects of the model (e.g., parameter tuning), if the data is not sufficiently normalized or exhibits high variability that the model cannot manage, optimization alone is insufficient to mitigate overfitting. The model tends to memorize patterns only relevant to the training data, failing to generalize to the more diverse validation data. The Features1 feature set, on the other hand, remained stable across all datasets and labels and did not experience overfitting, both before and after optimization. Furthermore, after optimization, the combination of Features1 achieved the highest average F1-score, particularly in the MLPSentFeat+DS1+Features1 model.

Models that did not experience overfitting after optimization—namely, MLPSentFeat+DS1+Features1, MLPSentFeat+DS2+Features1, MLPSentFeat+DS3+Features1, MLPSentFeat+DS1+Features3, and MLPSentFeat+DS3+Features3—demonstrated strong generalization capabilities. This can be attributed to the use of Features1 and Features3, both of which are more flexible in handling a variety of data types and sentence variations. Features1 tends to select more general features and is not overly focused on dominant words, enabling the model to adapt to variations in sentence structure, whether in structured data such as DS1 or more variable data like DS2. As such, Features1 helps the model remain stable in both training and validation datasets without over-relying on patterns that are specific to the training data. Features3, which focuses on sentence pragmatics, allows the model to capture more complex sentence contexts, such as expressions of sympathy or interrogative sentences. Although Features3 is more flexible in handling sentences with pragmatic variation, it still experiences overfitting on

(0) [F1:0] Halo. (1) Terima kasih atas pertanyaannya untuk Alodokter. (2) Pertama tama, saya hendak mengkonfirmasi terlebih dahulu. (3) [F2:1] apakah yang anda maksudkan dengan telah datang bulan hampir selama 3 bulan adalah anda mengalami perdarahan terus menerus selama 3 bulan ? (4) Apakah anda salah mengetik, jadi maksudnya telat menstruasi ? (5) Apakah anda tidak mengalami menstruasi selama 3 bulan ? (6) [F4:2] Bila anda mengalami menstruasi atau perdarahan terus menerus selama 3 bulan, maka anda harus segera memeriksakan diri anda ke dokter kandungan. (7) [F3:3] Terutama bila perdarahan yang anda alami cukup banyak, dan anda saat ini mengalami gejala gejala kekurangan darah seperti pusing, pucat, keringat dingin, jantung berdebar, dan lain lain. (8) [F6:4] Bila anda mengalami keterlambatan datang bulan atau anda tidak menstruasi sudah tiga bulan, maka sebaiknya anda melakukan pemeriksaan kehamilan terlebih dahulu dengan menggunakan testpack. (9) Lakukan pemeriksaan dengan menggunakan kencing pertama di pagi hari agar hasilnya lebih akurat dan pastikan anda membaca dan mengikuti dengan baik petunjuk pemakaian pada kemasan alat. (10) [F3:5] Bila hasilnya positif, maka keterlambatan menstruasi anda disebabkan oleh kehamilan. (11) Bila hasilnya negatif, maka perlu dipikirkan kemungkinan penyebab lainnya. (12) Misalnya saja. (13) Gangguan hormonal karena tubuh yang terlalu gemuk atau terlalu kurus. (14) Peningkatan ataupun penurunan berat badan berlebihan. (15) Stress berlebihan. (16) Olahraga berlebihan. (17) Penggunaan obat obatan tertentu ataupun kontrasepsi hormonal. (18) Gangguan hormonal karena penyakit tertentu Seperti PCOS atau Polycystic Ovarian Syndrome, diabetes, gangguan hormon tiroid tumor atau kanker tertentu yang mengganggu kondisi hormonal. (19) [F4:6] Anda harus melakukan pemeriksaan ke dokter kandungan dan sebaiknya pemeriksaan tidak ditunda tunda bila memang sudah tiga bulan anda tidak mengalami menstruasi. (20) Dokter kandungan akan melakukan pemeriksaan lebih lanjut untuk mencari penyebab pasti gangguan menstruasi anda. (21) [F1:7] Sekian informasi dari saya. (22) [F5:8] Semoga cukup menjawab.

Figure 10: Text segmentation results of DataID #222.

Table 16: Greedy data similarity matrix DataID #222.

Classification Segment	Annotator Segment								
	[F1:0]	[F2:1]	[F4:2]	[F3:3]	[F6:4]	[F3:5]	[F4:6]	[F1:7]	[F5:8]
[F1:0] (0,1,2)		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
[F2:1] (3,4,5)	0.00		0.00	0.00	0.00	0.00	0.00	0.00	0.00
[F4:2] (6)	0.00	0.00		0.00	0.00	0.00	0.00	0.00	0.00
[F3:3] (7)	0.00	0.00	0.00		0.00	0.00	0.00	0.00	0.00
[F6:4] (8,9)	0.00	0.00	0.00	0.00		0.00	0.00	0.00	0.00
[F3:5] (10,11, 12, 13,14,15,16, 17,18)	0.00	0.00	0.00	0.00	0.00		0.00	0.00	0.00
[F4:6] (19,20)	0.00	0.00	0.00	0.00	0.00	0.00		0.00	0.00
[F1:7] (21)	0.00	0.00	0.00	0.00	0.00	0.00	0.00		0.00
[F5:8] (22)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	

DS2, which contains a high level of variation in interrogative sentences. This is because Features3 struggles to handle the high variability in sentence structures in DS2, which are inconsistently structured. While optimization enhances certain aspects of the model, Features3 remains challenged in dealing with complex, unstructured interrogative sentences in DS2, leading to overfitting. On the other hand, in DS1 and DS3, which have more consistent sentence structures, Features3 is able to effectively handle pragmatic sentence structures, preventing overfitting and enabling better generalization on validation data.

Models that did not experience overfitting after optimization—namely, MLPSentFeat+DS1+Features1, MLPSentFeat+DS2+Features1, MLPSentFeat+DS3+Features1, MLPSentFeat+DS1+Features3, and MLPSentFeat+DS3+Features3—demonstrated strong generalization capabilities. It can be attributed to Features1 and Features3, which are more flexible in handling various data types and sentence variations. Features1 tends to select more general features and is not overly focused on dominant words, enabling the model to adapt to variations in sentence structure, whether in structured data such as DS1 or more variable data like DS2. As such, Features1 helps the model remain stable in training and validation datasets without over-relying on patterns specific to the training data. Features3, which focuses on sentence pragmatics, allows the model to capture more complex sentence contexts, such as expressions of sympathy or interrogative sentences. Although Features3 is more flexible in handling sentences with pragmatic variation, it still experiences overfitting on DS2, which contains a high level of variation in interrogative sentences. It is because Features3 struggles to handle the high variability in sentence structures in DS2, which are inconsistently structured. While optimization enhances certain aspects of the model, Features3 remains challenged in dealing with complex, unstructured interrogative sentences in DS2, leading to overfitting. On the other hand, in DS1 and DS3, which have more consistent sentence structures, Features3 can effectively handle pragmatic sentence structures, preventing overfitting and enabling better generalization on validation data.

Based on the experiments conducted, both with and without optimization, it can be concluded that optimization provides a mathematical performance improvement, such as more stable training loss reduction and increased accuracy in some models. However, optimization also negatively affects certain model combinations, such as the sharp decline in the F1 score for label F2 in DS2, indicating that optimization caused overfitting for that label. Although the optimized models achieved higher average F1 scores in some combinations, optimization did not adequately address the variation in sentences present in DS2, especially the varying interrogative sentences, which led the model to become too focused on patterns in the training data and fail to generalize well to the validation or test data. Overall, although optimization improves certain aspects of the model, it is not always better than the model without optimization, especially when the validation data contains

substantial variation. Optimization often exacerbates the overfitting issue, as seen with label F2 in DS2.

Table 17: Doctor's answer data segment evaluation results.

Model	AVG	STD	MAX	MIN	Identical Data	Segment Error
HTCN+DS1	0.9085	0.1074	1.0000	0.3309	2478	46.62%
HTCN+DS2	0.9081	0.1237	1.0000	0.2996	4508	43.65%
HTCN+DS3	0.9568	0.0850	1.0000	0.4444	706	29.33%
LDA+DS1	0.5963	0.0921	0.9041	-0.0249	0	100.00%
LDA+DS2	0.6209	0.0943	1.0000	0.0389	3	99.96%
LDA+DS3	0.4948	0.2022	1.0000	-1.0000	1	99.90%
CVS+DS1	0.7023	0.1061	0.9370	0.1060	0	100.00%
CVS+DS2	0.7022	0.0975	0.9723	0.0000	0	100.00%
CVS+DS3	0.6563	0.1377	1.0000	-0.1678	1	99.90%
K-means+DS1	0.7077	0.0997	1.0000	0.2000	8	99.85%
K-means+DS2	0.7343	0.0900	1.0000	0.1977	11	99.86%
K-means+DS3	0.6750	0.2055	1.0000	-0.3072	2	99.80%
MLPSentFeat+DS1+Features1	0.9729	0.0591	1.0000	0.5077	4203	22.31%
MLPSentFeat+DS1+Features2	0.9738	0.0593	1.0000	0.5077	4288	20.74%
MLPSentFeat+DS1+Features3	0.9821	0.5077	1.0000	0.0518	4669	13.69%
MLPSentFeat+DS2+Features1	0.9830	0.0582	1.0000	0.4440	7231	9.61%
MLPSentFeat+DS2+Features2	**0.9847**	**0.0547**	**1.0000**	**0.4945**	**7288**	**8.90%**
MLPSentFeat+DS2+Features3	0.9829	0.4306	1.0000	0.0583	7216	9.80%
MLPSentFeat+DS3+Features1	0.9754	0.5233	1.0000	0.0672	825	17.42%
MLPSentFeat+DS3+Features2	0.9763	0.5500	1.0000	0.0657	830	17.00%
MLPSentFeat+DS3+Features3	0.9782	0.5653	1.0000	0.0606	828	17.20%

To achieve a model with a high average value and no overfitting, the optimal combination is to use Features1, which is more flexible and general. Features1 can better adapt to the variations present in the data in a more generalizable way, without focusing too much on dominant features that could lead to overfitting. In datasets with high variation, such as DS2, models with Features1 are more flexible in handling varied sentences. Although optimization can improve some aspects, the Features1 combination remains more stable and does not lead to overfitting. Therefore, while optimization provides a mathematical improvement, combinations using Features1 provide more stable results and can achieve high values without sacrificing generalization ability.

4.4. Text segmentation process based on six aspects of communication

This segmentation aims to form small segments based on the classification results and labels. This text segmentation model has two processes: segment formation, annotation, and segment testing. The segment formation and segment annotation process aim to form segments and annotate segments based on aspects of doctor-patient communication. Segment testing aims to determine the performance of segments from the classification stage. The greedy similarity method is used as a text segmentation test. The data used in the text segmentation stage is test data of 20% of the labeled data; the distribution is shown in Table 4. The output of this model is the results of text segmentation of doctors' answers and segment annotations, as well as statistics on greedy similarity.

4.4.1. Segment formation and segment annotation

Several sentences have the same label but are not located next to each other, these sentences become different segments. Alternatively, if sentences are located next to each other but the labels are not the same, these sentences become different segments (Figure 5). This method applies to labels from text classification results and labels from annotators. The resulting segments and their annotations will be similar if the labels from the text classification results match the labels given by the annotator. One segment can contain one or more sentences. One dataset can contain several segments. This is formed based on aspects of doctor-patient communication, as in Figure 1. This segmented doctor's answer data is the final result of this research.

An example of segment formation and segment annotation from ID-222 data is shown in Figure 10. ID-222 data has 23 sentences, while the segmentation results of ID-222 data form 9 segments. Segment information is placed in square brackets [..] and at the sentence's beginning. The segment description contains the segment annotation and serial number, separated by a colon (:). For example, [F1:0], F1 means that the segment has an annotation in the form of F1 or fostering the relationship aspect, the number 0 means the segment serial number in the data, data ID-222. The number enclosed in regular brackets (...) and located in front of the segment or sentence description is the sentence serial number. The colors in Figure 10 are only to make distinguishing one segment from another easier. Segments that have the same segment annotation have the same color.

The method for forming segments based on ID-222 data is that sentences 0 to 2 have the same label in the form of F1, and the three sentences are located next to each other or in sequence so that they can form one segment,

namely segment 0. Meanwhile, sentence 21 has the same label as sentences 0, 1, and 2, but the location of sentence 21 does not follow these three sentences, so sentence 21 cannot be a segment with segment 0. So, sentence 21 forms a new segment, namely segment 7. Another example, sentences 2 and 3 are located separately. These two sentences follow each other but have different labels, so the two sentences are not located in the same segment or form different segments. This model forms segments based on the similarity of labeling and sentence order.

The description of the nine ID-222 data segments, namely segment 0, has the annotation F1 (fostering the relationship), consisting of 3 sentences (sentences 0 to 2). Segment 1 has annotation F2 (gathering information), consisting of 3 sentences (sentences 3 and 5). Segment 2 has the annotation F4 (decision making), consisting of 1 sentence (sentence 6). Segment 3 has the annotation F3 (providing information), consisting of 1 sentence (7). Segment 4 has the annotation F6 (enabling disease and treatment-related behavior), consisting of 2 sentences (8 and 9). Segment 5 has the annotation F3, consisting of 9 sentences, 10 to 18. Segment 6 has annotation F4, consisting of 2 sentences (19 and 20). Segment 7 has annotation F1, consisting of 1 sentence (sentence 21). Segment 8 has the annotation F5 (responding to the emotion), consisting of 1 sentence (sentence 22).

4.4.2. Segment evaluation

Evaluation of this segment uses the greedy similarity algorithm, where the classification result segment is compared with the segment from the annotator. Greedy similarity calculates the similarity of sentences in the same segment between the classification segment and the annotator. Comparisons are made in detail based on the sentences in a segment, by comparing classification labels as rows and annotator labels as columns. Algorithm 2 shows the calculation steps for evaluating this segment. The greedy similarity calculation process forms a matrix for each dataset and calculates the greedy similarity value for each dataset. The range of greedy similarity values is between 0 and 1. A value of 1 indicates that the segment of the classification label is the same as the label of the annotator; this situation is called correct data. On the other hand, a greedy similarity value of less than 1 indicates a segment in the classification label that does not match the annotator's label, which is called wrong data.

Example for filling in the matrix elements of data using the greedy similarity algorithm calculation. This matrix filling process applies to all doctors' answer data. An example of filling in the ID-222 data matrix is shown in Table 16. ID-222 data has the same classification and annotator labels, so the segment results are the same. First, the greedy similarity matrix is formed (can be seen in Table 16). The matrix's rows and columns consist of the label's name from the classification process or annotator label, the sequence number of the segment, and the sequence number of the sentences in that segment. For example, row and column 1 contain [F1:0] (0,1,2), meaning that the row and column include segment 0, the annotation label is F1, either the label from the classification process or the annotator label, as well as sentences 0, 1, and 2.

Matrix elements are filled in by comparing rows and columns, such as labels and sentences in the segments. For example, element (0,0) is obtained by comparing segment 0 and segment 0, element (0,1) is obtained by comparing segment 0 and segment 1, and so on.

When comparing rows and columns, pay attention to the labels and sentences in the segments. Conditions for filling in matrix elements are as follows: 1) If the label annotations in the row and column are different, then the matrix element is filled with the number 0. For example, element (0,1), row 1 ([F1:0] (0,1,2)), and column 1 ([F2:1] (3,4,5)) contain 0 because the label for row 1 (F1) is different from the label for column 2 (F2). 2) If the segment annotations in the row and column are the same, then pay attention to the sentences in the row and column segments. For example, elements in row 1 (F1:0] (0,1,2)) and column 1 ([F1:0] (0,1,2)) have the same label, namely F1, so pay attention to the sentences in the row and column segments. Row segment 1 has sentences 0, 1, and 2. Column segment 1 also has sentences 0, 1, and 2. Count the total number of correct sentences, namely, sentences that are the same and are in the classification results segment and the annotator segment simultaneously. For example, in element (0,7), row 1, and column 8, these two segments have the same label annotation, namely F1, but the sentences in these segments are different. The segment in row 1 has three sentences, while the segment in column 8 has one. None of the sentences in the two segments are the same, so the element (0.7) is filled with the value 0.00. Another example is element (0,0), row 1, and column 1. These two segments have three sentences located simultaneously in the row and column, so the correct sentence from this element is 3. Then it is calculated using the formula, namely 2 * the number of correct sentences / (the number of row sentences in a segment+the number of column sentences in a segment). So the elements of row 1 and column 1 are calculated by 2 * 3 / (the number of sentences in row 1 is 3+the number of sentences in column 1 is 3) = 6 / 6 = 1, then the elements of row 1 and column 1 contain 1.00.

Next, the final greedy similarity value is calculated after the ID-222 data matrix elements are filled in. Calculate the final greedy similarity value for all doctors' answer data. The following is an example of calculating the final greedy similarity value for ID-222 data by 1 selecting several of the largest values from the greedy similarity matrix. The largest value cannot be located in one row with the same column as another row. The yellow greedy similarity matrix elements (Table 16) indicate that the greedy similarity matrix has the largest values. The values appear not to be located in the same row and column. For example, the largest value 1 is 1, is in the elements of row 1 and column 1. The largest value 2 is 1, is in the elements of row 2 and column 2. 2) Calculate the greedy similarity value with the formula, namely 2 * the total largest value / (number of matrix row sentences+number of matrix column sentences). The calculation of the ID-222 greedy similairty value is 2 * (1+1+1+1+1+1+1+1+1) / (9+9) = 18 / 18 = 1. So the ID-

222 greedy similarity value is 1. It means that the ID-222 data has identical segments between the segments from the classification process and the segments from the annotator.

Thus, the greater the final greedy similarity value of the data, the higher the level of similarity between the segments resulting from the classification process and the segments from the annotator, and it will even be identical if the final greedy similarity value reaches 1. On the other hand, if the final greedy similarity value is close to 0, the segments formed by the classification process are very different from those formed by the annotator.

4.4.3. Text segmentation experiment

After calculating the final greedy similarity value for all the doctors' response data, statistical analysis was performed on these values by calculating the average (AVG), standard deviation (STD), minimum value (MIN), maximum value (MAX), the number of identical data points, and the segment error percentage. The AVG metric measures the average similarity between the automatic segmentation (classification) results and the annotator across the entire dataset. This metric provides an overall view of the method's performance, where a higher AVG value indicates that the segmentation produced is more relevant and closer to the desired result. The STD metric helps assess the method's reliability under different conditions by measuring the variation or spread of the segmentation results produced by the model. A lower STD value indicates that the model's segmentation results are more consistent, with fewer fluctuations between the segments produced, and vice versa. The MIN metric is used to identify the worst-case performance of the method, which may reveal specific data or conditions that cause the method to fail or significantly differ from the annotator. The MAX metric is used to identify the best-case scenario where the method performs most optimally. If many values approach the MAX, the method often performs very well. If the MAX is far from the AVG, it suggests that only a few cases are optimal. The number of identical data points refers to data that produces the same segments between the classification process results and the annotator's segments. A higher identical data value indicates that the model is more effective in producing segments that match those of the system (classification) and the annotator, and vice versa. The segment error percentage, calculated based on the number of incorrect data points compared to the total data points, is performed because the data types vary. A lower segment error percentage indicates that the model is more accurate in dividing the data into relevant segments with fewer errors, and vice versa. The experimental results of text segmentation formation on the doctor's response data based on six communication aspects are presented in Table 17.

Statistical analysis of the final greedy similarity value for all doctors' answer data using the MLPSentFeat model with various DS1, DS2, and DS3 data, and various feature sentences Features1, Features2, and Features3, is shown in Table 17. Apart from that, segment formation using the HTCN, CVS [38], LDA [116], and K-means [117]. Previous researchers have used these methods to form text segments, so the author used this method to segment the doctor's answer data based on six aspects. The parameters used by the LDA method to form segments are the topics displayed based on the number of unique labels for each data (n_components=Label value), max_iter=20, random state=0, doc_topic_prior=0.1, and topic_word prior=0.1. The main parameter in the K-Means method for segment formation is the number of clusters (k), which is determined by the number of communication aspects presented in each doctor's response. Although there are six communication aspects, not all responses contain all six aspects. The number of clusters can be optimized using the elbow method to achieve more optimal segmentation results [118]. However, in this study, the number of clusters could not be optimized, as the value of k was predetermined based on the number of communication aspects identified in each response. Thus, the number of clusters directly represents the number of segments formed for each response. The other parameters used in the K-Means method were: init='k-means++', max_iter=300, n_init=100, tol=0.0001, and random_state=seed. The parameters used by the CVS method in forming segments are the minimum sentence limit for data, namely 4, penalty = get_penalty([sentence_vectors], segment_len), and sentence_vectors = vecr.transform(sentence_text).dot(wrdvecs).

Table 17 presents the results of the segmentation evaluation of doctors' answer data using various methods based on five metrics. The MLPSentFeat method performs better with the highest AVG and lowest STD values, while the LDA method records the lowest performance. The LDA+DS3 method has the lowest AVG of 0.4948 with the highest STD among all methods of 0.2022, as well as an extreme MIN value of -1.0000, which indicates instability and low segmentation accuracy. In contrast, the HTCN and K-means methods show moderate performance with AVG values ranging from 0.6750 to 0.9568, depending on the dataset used. The CVS model also shows moderate performance with the highest AVG of 0.7023 on the DS1 combination. The overall data shows that the combination of model and dataset greatly influences the stability and accuracy of segmentation results.

More specifically, the MLPSentFeat model, combined with various datasets and sentence features, demonstrates the most consistent and superior performance. Based on the evaluation results conducted on two segmentation models, MLPSentFeat+DS2+ Features2 and MLPSentFeat+DS1+ Features3, several significant differences are observed in terms of segmentation quality, consistency, and accuracy. First, regarding the AVG metric, the MLPSentFeat+DS2+ Features2 model shows a slightly higher value (0.9847) than the MLPSentFeat+DS1+Features3 model (0.9821). This AVG metric indicates that MLPSentFeat+DS2+ Features2 provides slightly better overall segmentation quality, although the difference is minimal. Second, in the STD metric, MLPSentFeat+DS2+Features2 has a very low STD value (0.0547), indicating that this model is more

Table 18: Data segment evaluation results from optimized classification.

Model	AVG	STD	MAX	MIN	Identical Data	Segment Error
MLPSentFeat+DS1+Features1	0.9822	0.0495	1.0000	0.5077	4616	14.68%
MLPSentFeat+DS1+Features2	0.9760	0.0578	1.0000	0.5000	4384	18.96%
MLPSentFeat+DS1+Features3	0.9768	0.5077	1.0000	0.0562	4422	18.26%
MLPSentFeat+DS2+Features1	0.9863	0.0508	1.0000	0.4840	7329	8.39%
MLPSentFeat+DS2+Features2	0.9825	0.0574	1.0000	0.4494	7171	10.36%
MLPSentFeat+DS2+Features3	**0.9867**	**0.5091**	**1.0000**	**0.0503**	**7346**	**8.18%**
MLPSentFeat+DS3+Features1	0.9752	0.0678	1.0000	0.5505	823	17.70%
MLPSentFeat+DS3+Features2	0.9783	0.0627	1.0000	0.5505	843	15.70%
MLPSentFeat+DS3+Features3	0.9768	0.5527	1.0000	0.0640	826	17.4%

consistent in producing stable segments without large fluctuations between the generated segments. In contrast, the MLPSentFeat+DS1+Features3 model has a higher STD value (0.5077), which indicates greater variation in the segmentation results. However, this does not mean the model is unstable. The low STD in MLPSentFeat+DS2+Features2 shows better stability, which is crucial for applications requiring consistency in segmentation across varying data. Next, in the Segment Error Percentage metric, MLPSentFeat+ DS1+Features3 records a lower value (6.62%) than MLPSentFeat+DS2+Features2 (8.90%). This finding indicates that MLPSentFeat+DS1+Features3 is more effective in separating the data into relevant segments, with fewer segmentation errors. This metric suggests that MLPSentFeat+DS1+Features3 has higher segmentation accuracy, although MLPSentFeat+DS2+Features2 demonstrates better consistency. Although MLPSentFeat+DS2+Features2 is slightly superior in terms of segmentation quality and consistency, MLPSentFeat+DS1+Features3 excels in segmentation accuracy, with a lower Segment Error Percentage. Therefore, if segmentation accuracy is the top priority, MLPSentFeat+DS1+Features3 can be considered the best model. However, if consistency and stability in segmentation results are prioritized, MLPSentFeat+DS2+ Features2 would be the more suitable choice.

Based on the segmentation evaluation results from the optimized classification presented in Table 18, the MLPSentFeat+DS2+Features3 model demonstrates the best overall segmentation performance. Two primary metrics indicate this conclusion: the lowest segment error percentage of 8.18% and the highest number of identical data points, totaling 7346. These results imply that this model achieves the most accurate segmentation alignment with the manual annotations while producing the fewest segmentation errors among all evaluated models. However, when considering performance stability across data samples (as reflected by the STD), the MLPSentFeat+DS1+Features1 model yields the lowest STD value of 0.0495, indicating the most consistent performance. Despite this stability, the model shows a relatively high segment error percentage of 14.68%, suggesting that it is less accurate in segmenting the data, even though it maintains a stable output. In contrast, the MLPSentFeat+DS2+Features1 model offers a well-balanced compromise, with the second-lowest segment error percentage (8.39%) and the second-lowest STD (0.0508). These findings demonstrate that the model is relatively accurate and reasonably stable. Furthermore, its average similarity score (AVG) is 0.9863, close to the highest observed value (0.9867).

In summary, if the primary objective is to maximize segmentation accuracy, the MLPSentFeat+DS2+ Features3 model is the most suitable choice, despite its relatively lower stability (STD = 0.5091). On the other hand, if a more stable yet sufficiently accurate model is preferred, then MLPSentFeat+DS2+Features1 represents a more balanced alternative. Conversely, although MLPSentFeat+DS1+Features1 demonstrates the highest stability, it is not recommended when segmentation accuracy is the main priority, due to its relatively high error rate.

In comparing the results from Tables 17 and 18, which presented the segment evaluation of the MLPSentFeat model across different feature sets and datasets, notable improvements and variations in performance were observed. In Table 17, the MLPSentFeat+DS2+Features2 model achieved the best performance, with an average of 0.9847 and a segment error percentage of 8.90%. The worst-performing model in this table was MLPSentFeat+DS1+Features3, which showed a segment error percentage of 13.69%. Upon optimization, as shown in Table 18, the same model, MLPSentFeat+DS2+ Features2, demonstrated an improvement in performance, achieving a segment error percentage of 8.39%, thus confirming the positive impact of the optimization process. However, the worst-performing model in the optimized classification, MLPSentFeat+DS3+Features1, showed a significant increase in segment error to 17.70%, highlighting the negative effect of the optimization on this particular variation. Overall, while the optimization process resulted in slight improvements in the performance of the best-performing models, it also led to a deterioration in the performance of certain configurations, particularly for those utilizing DS3. These findings underscored the necessity of careful model selection and an understanding of dataset characteristics to achieve optimal segmentation performance.

5 Conclusions

This study proposes a novel approach to text segmentation focused on communication aspects, specifically doctor-patient communication. Unlike classical segmentation methods that rely on topic similarity, this approach segments text based on the communicative function of each sentence, thus more accurately reflecting the structure of interactions in a medical context. The developed model, MLPSentFeat, explicitly integrates sentence features into its architecture, proving effective in handling data with highly imbalanced label distributions—such as cases where one label has only a few dozen data points, while others have thousands—without requiring synthetic data or complex balancing techniques.

The MLPSentFeat model consists of four main stages: pseudo-labeling, classification, text segmentation, and optimization. In the classification stage, sentence features are formed using a combined approach, with the LR algorithm filtering relevant words for each label and InfoGain selecting the most informative features. The results of the classification experiments show that integrating these sentence features significantly improves the model's sensitivity to variations in linguistic structure, particularly in recognizing non-standard sentences, such as questions commonly found in the information-gathering aspect. It is evident from the experiment, which shows a sharper decline in the F1-score for F2 (gathering information) compared to other labels, both before and after optimization. This decline is attributed to the high variability in question sentence structures in the DS2 dataset. The best classification model before optimization was MLPSentFeat+DS2+Features2, with an F1-score of 0.9925.

In the text segmentation stage, the MLPSentFeat+DS2+Features2 model also yielded the best results, with an average F1-score of 0.9847 and the lowest segment error percentage of 8.90%, indicating that the model performs well in segmenting text. After parameter optimization, the model remained stable, although it did not achieve the highest average F1-score. However, this challenge led to the model experiencing overfitting.

The use of text normalization, data size, cleanliness, and the alignment of sentence features with the semantic structure of sentences are key factors in preventing overfitting. Features2 has been proven well-suited for high-variability data, such as DS2, despite fluctuations in label F2, as seen in the MLPSentFeat+DS2+Features2 model. On the other hand, Features3 is better suited for well-structured datasets like DS1 and DS3, where the model is more effective in capturing pragmatic patterns, especially in sentences with a more formal structure. It is evident from the combination of Features3 with DS1 and DS3 in label F2, which shows F1-score stability, with the Features3 and DS3 combination achieving the highest F1-score for label F2, both before and after optimization. Features1 exhibited better stability across the entire dataset, meaning it functions well with both structured and varied data, though its results were more generic and less in-depth compared to Features2 or Features3. It is reflected in the stable average F1-score seen in combinations of Features1 with various datasets. The Features1+DS1 combination was the best after optimization, with an average F1-score of 0.9926. Moreover, this model was stable across all labels and did not experience overfitting. However, this model showed a relatively high segment error percentage after text segmentation, indicating that MLPSentFeat+DS1+ Features1 was not yet optimal in completing the text segmentation task.

No model was dominant in the experiment results, though several models exhibited distinct advantages and stability. Based on the goal of segmenting text based on communication aspects, the best model produced was MLPSentFeat+DS2+Features3 after optimization. This model completed the text segmentation task with the lowest segment error percentage of 8.18%. Furthermore, optimizing the model's parameters had a positive impact, as evidenced by the number of models achieving an average F1-score above 0.99 and the improved classification consistency in most model combinations. Nevertheless, not all models experienced an increase in average F1-score or accuracy after optimization. This finding is due to the optimization's inability to handle data diversity, even though mathematically, the optimization improved the model's performance. The MLPSentFeat model has been trained and tested with various health domains, showing that the model can adapt well to cross-domain applications.

Future work includes applying the MLPSentFeat model to doctor responses in different languages or other domains. Additionally, efforts will further improve the model's ability to identify questions with varying sentence structures, not limited to formal question structures.

References

[1] X. Kou, “Personalized Doctor Recommendation Method Based on Patient Online Consultation Text,” *2021 IEEE Int. Conf. Comput. Sci. Electron. Inf. Eng. Intell. Control Technol. CEI 2021*, pp. 31–34, Sep. 2021, doi: 10.1109/CEI52496.2021.9574585.

[2] M. Tanis, T. Hartmann, and F. te Poel, “Online Health Anxiety and Consultation Satisfaction: A Quantitative Exploratory Study on Their Relations,” *Patient Educ. Couns.*, vol. 99, no. 7, pp. 1227–1232, Jul. 2016, doi: 10.1016/J.PEC.2016.01.021.

[3] Y. Rahmawati, D. Siahaan, and D. Purwitasari, “Text Segmentation Methods for Annotation on eHealth Consultation with Interview Function Labels: A Comparative Study,” *8th Int. Conf. Softw. Eng. Comput. Syst. ICSECS 2023*, pp. 72–77, 2023, doi: 10.1109/ICSECS58457.2023.10256340.

[4] S. Hobma, P. Ram, A. Muijtjens, C. van der Vleuten, and R. Grol, “Effective Improvement of Doctor-Patient Communication: A Randomised Controlled Trial,” *Br. J. Gen. Pract.*, vol. 56, no.

529, pp. 580–586, 2006.

[5] Y. Rahmawati, D. Siahaan, D. Purwitasari, R. A. Wijayanti, and Violita Elsha Salshabillah, "Text Classification based on Sentence Features and Preprocessing Settings for Labeling eHealth Consultation Answer," in *7th International Seminar on Research of Information Technology and Intelligent Systems*, Yogyakarta: IEEE, 2024, pp. 1065–1070. doi: 10.1109/ISRITI64779.2024.10963536.

[6] M. Hadryan and J. Rutecka-Góra, "Readability and Clarity of Individual Pension Product Contracts," *Int. J. Semiot. Law*, vol. 36, no. 4, pp. 1749–1777, Aug. 2023, doi: 10.1007/S11196-023-09997-8/TABLES/12.

[7] A. van den Brink-Muinen *et al.*, "Doctor–patient communication in different European health care systems:: Relevance and performance from the patients' perspective," *Patient Educ. Couns.*, vol. 39, no. 1, pp. 115–127, Jan. 2000, doi: 10.1016/S0738-3991(99)00098-1.

[8] J. W. Y. Kee, H. S. Khoo, I. Lim, and M. Y. H. Koh, "Communication Skills in Patient-Doctor Interactions: Learning from Patient Complaints," *Heal. Prof. Educ.*, vol. 4, no. 2, pp. 97–106, Jun. 2018, doi: 10.1016/J.HPE.2017.03.006.

[9] R. Ramdani, A. F. Huda, and M. A. Bijaksana, "Text Segmentation Based on Word Embedding on Indonesian Quran Translation by Greedy with Window Approaching," *Int. Conf. Inf. Commun. Technol.*, vol. 7, Jul. 2019, doi: 10.1109/ICOICT.2019.8835338.

[10] S. Martikainen *et al.*, "Text-based Patient – Doctor Discourse Online And Patients' Experiences of Empathy," *Comput. Support. Coop. Work CSCW An Int. J.*, vol. 33, no. 4, pp. 1151–1175, Dec. 2024, doi: 10.1007/s10606-023-09481-8.

[11] A. King and R. B. Hoppe, "'Best Practice' for Patient-Centered Communication: A Narrative Review," *J. Grad. Med. Educ.*, vol. 5, no. 3, pp. 385–393, Sep. 2013, doi: 10.4300/JGME-D-13-00072.1.

[12] M. Mostafapour, J. D. Smith, J. H. Fortier, and G. E. Garber, "Beyond Medical Errors: Exploring the Interpersonal Dynamics in Physician-Patient Relationships Linked to Medico-Legal Complaints," *BMC Health Serv. Res.*, vol. 24, no. 1, pp. 1–11, Dec. 2024, doi: 10.1186/S12913-024-11457-3/TABLES/2.

[13] N. A. Mohd Salim, N. S. Roslan, R. Hod, S. F. Zakaria, and S. K. Adam, "Exploring Critical Components of Physician-Patient Communication: A Qualitative Study of Lay and Professional Perspectives," *Healthcare*, vol. 11, no. 2, p. 162, Jan. 2023, doi: 10.3390/HEALTHCARE11020162.

[14] J. F. Ha and N. Longnecker, "Doctor-Patient Communication: A Review," *Ochsner J.*, vol. 10, no. 1, p. 38, 2010, doi: 10.3329/jbcps.v32i2.26036.

[15] E. Bindels, C. Verberg, A. Scherpbier, S. Heeneman, and K. Lombarts, "Reflection revisited: How physicians conceptualize and experience reflection in professional practice - A qualitative study," *BMC Med. Educ.*, vol. 18, no. 1, pp. 1–10, May 2018, doi: 10.1186/S12909-018-1218-Y/TABLES/1.

[16] F. B. Romeiro, V. G. Pais, G. Humphris, and M. Figueiredo-Braga, "Patients' Emotional Expressions and Clinicians' Responses in Oncology – From Recognition to Exploration of Concerns," *PEC Innov.*, vol. 6, p. 100374, Jun. 2025, doi: 10.1016/J.PECINN.2025.100374.

[17] Y. Luo *et al.*, "How about Trust in Physician-Patient Relationship? A Concept Analysis of Physicians' Perspectives," *Patient Educ. Couns.*, vol. 112, p. 107709, Jul. 2023, doi: 10.1016/J.PEC.2023.107709.

[18] W. Wallace *et al.*, "The diagnostic and triage accuracy of digital and online symptom checker tools: a systematic review," 2022. doi: 10.1038/s41746-022-00667-w.

[19] W. Wallace *et al.*, "The Diagnostic and Triage Accuracy of Digital and Online Symptom Checker Tools: a Systematic Review", doi: 10.1038/s41746-022-00667-w.

[20] V. R. Hananto, U. Serdült, and V. Kryssanov, "A Text Segmentation Approach for Automated Annotation of Online Customer Reviews, Based on Topic Modeling," *Appl. Sci. 2022, Vol. 12, Page 3412*, vol. 12, no. 7, p. 3412, Mar. 2022, doi: 10.3390/APP12073412.

[21] A. Tagarelli and G. Karypis, "A Segment-based Approach to Clustering Multi-Topic Documents," *Knowl. Inf. Syst.*, vol. 34, no. 3, pp. 563–595, 2013, doi: 10.1007/s10115-012-0556-z.

[22] B. Qolomany, M. Maabreh, A. Al-Fuqaha, A. Gupta, and D. Benhaddou, "Parameters Optimization of Deep Learning Models using Particle Swarm Optimization," *2017 13th Int. Wirel. Commun. Mob. Comput. Conf. IWCMC 2017*, pp. 1285–1290, Jul. 2017, doi: 10.1109/IWCMC.2017.7986470.

[23] C. Bi *et al.*, "Optimizing a Multi-Layer Perceptron Based on an Improved Gray Wolf Algorithm to Identify Plant Diseases," *Math. 2023, Vol. 11, Page 3312*, vol. 11, no. 15, p. 3312, Jul. 2023, doi: 10.3390/MATH11153312.

[24] F. Z. El-Hassani, M. Amri, N. E. Joudar, and K. Haddouch, "A New Optimization Model for MLP Hyperparameter Tuning: Modeling and Resolution by Real-Coded Genetic Algorithm," *Neural Process. Lett.*, vol. 56, no. 2, pp. 1–31, Apr. 2024, doi: 10.1007/S11063-024-11578-0/FIGURES/13.

[25] R. Anggoro, W. Suadi, A. M. Shiddiqi, R. Ijtihadie, S. Lili, and D. W. L. Pamungkas, "The Development of Blackhole Attack in AODV Routing Protocol," *CENIM 2020 - Proceeding Int. Conf. Comput. Eng. Network, Intell. Multimed. 2020*, no. Cenim, pp. 309–314, 2020, doi:

10.1109/CENIM51130.2020.9297962.

[26] J. Li, B. Chiu, S. Shang, and L. Shao, "Neural Text Segmentation and its Application to Sentiment Analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 2, pp. 828–842, Feb. 2022, doi: 10.1109/TKDE.2020.2983360.

[27] H. Misra, F. Yvon, O. Cappé, and J. Jose, "Text Segmentation: A Topic Modeling Perspective," *Inf. Process. Manag.*, vol. 47, no. 4, pp. 528–544, Jul. 2011, doi: 10.1016/J.IPM.2010.11.008.

[28] T. Brants, F. Chen, and I. Tsochantaridis, "Topic-Based Document Segmentation with Probabilistic Latent Semantic Analysis," in *International Conference on Information and Knowledge Management, Proceedings*, Association for Computing Machinery (ACM), 2002, pp. 211–218. doi: 10.1145/584792.584829.

[29] F. Y. Y. Y. Choi, P. Wiemer-Hastings, J. Moore, F. Y. Y. Choi., P. Wiemer-Hastings, and J. Moore, "Latent Semantic Analysis for Text Segmentation," *Proc. 2001 Conf. Empir. Methods Nat. Lang. Process.*, vol. 102, no. 7, pp. 109–117, 2001.

[30] S. R. Joty, G. Carenini, and R. T. Ng, "Topic Segmentation and Labeling in Asynchronous Conversations," *J. Artif. Intell. Res.*, vol. 47, pp. 521–573, Feb. 2014, doi: 10.1613/jair.3940.

[31] M. Bayomi and S. Lawless, "C-HTS: A concept-based hierarchical text segmentation approach," *Lr. 2018 - 11th Int. Conf. Lang. Resour. Eval.*, pp. 1519–1528, 2019.

[32] J. W. Wu, J. C. R. Tseng, and W. N. Tsai, "An efficient linear text segmentation algorithm using hierarchical agglomerative clustering," *Proc. - 2011 7th Int. Conf. Comput. Intell. Secur. CIS 2011*, pp. 1081–1085, 2011, doi: 10.1109/CIS.2011.240.

[33] M. A. Hearst, "TextTiling : Segmenting Text into Multi-Paragraph Subtopic Passages," *1997 Assoc. Comput. Linguist.*, vol. 23, no. 1, pp. 33–64, 1997.

[34] F. Y. Y. Y. Choi, "Advances in Domain Independent Linear Text Segmentation," *1st Meet. North Am. Chapter Assoc. Comput. Linguist.*, vol. 1, pp. 26–33, 2000.

[35] G. Glavaš, F. Nanni, and S. P. Ponzetto, "Unsupervised Text Segmentation Using Semantic Relatedness Graphs," *Assoc. Comput. Linguist.*, pp. 125–130, Aug. 2016, doi: 10.18653/v1/S16-2016.

[36] S. Zhafiirah, Y. Rahmawati, D. Purwitasari, and D. O. Siahaan, "Semantic Relatedness Graph for Text Segmentation of Patient-Centered Communications in Question-Answer Data," *2024 9th Int. Conf. Informatics Comput. ICIC 2024*, pp. 1–6, 2024, doi: 10.1109/ICIC64337.2024.10957542.

[37] A. A. Alemi and P. Ginsparg, "Text Segmentation based on Semantic Word Embeddings," Mar. 2015. doi: https://doi.org/10.48550/arXiv.1503.05543.

[38] V. Gupta, G. Zhu, A. Yu, and D. E. Brown, "A Comparative Study of the Performance of Unsupervised Text Segmentation Techniques on Dialogue Transcripts," *2020 Syst. Inf. Eng. Des. Symp. SIEDS 2020*, pp. 0–5, 2020, doi: 10.1109/SIEDS49339.2020.9106639.

[39] B. S. Wiguna, D. Purwitasari, and D. O. Siahaan, "Deep Learning Approach for Health Question and Answer Text Segmentation based on Physician-Patient Communication Aspect," *Procedia Comput. Sci.*, vol. 234, pp. 213–221, 2024, doi: 10.1016/j.procs.2024.02.168.

[40] I. Pak and P. L. Teh, "Text segmentation techniques: A critical review," *Stud. Comput. Intell.*, vol. 741, no. July, pp. 167–181, 2018, doi: 10.1007/978-3-319-66984-7_10.

[41] I. Ghinassi, "Unsupervised Text Segmentation via Deep Sentence Encoders: a First Step Towards a Common Framework for Text-Based Segmentation, Summarization and Indexing of Media Content," *Proc. 2nd Int. Work. Data-driven Pers. Telev. ACM Int. Conf. Interact. Media Exp. (IMX 2021), June 2021*, vol. 2, no. September, pp. 1–15, 2021, doi: 10.5281/zenodo.4744399.

[42] J. Zhu, C. Zhang, and M. Y. Ma, "Multi-aspect rating inference with aspect-based segmentation," *IEEE Trans. Affect. Comput.*, vol. 3, no. 4, pp. 469–481, 2012, doi: 10.1109/T-AFFC.2012.18.

[43] S. Chowdhury, H. Yerebakan, Y. Shinagawa, and P. S. Yu, "MedTextSeg: A Deep Dual Sequential Model for Section Segmentation in Medical Reports," *Proc. - 2021 IEEE Int. Conf. Big Data, Big Data 2021*, pp. 1582–1588, 2021, doi: 10.1109/BigData52589.2021.9671518.

[44] J. Eisenstein, "Hierarchical Text Segmentation from Multi-Scale Lexical Cohesion," *NAACL HLT 2009 - Hum. Lang. Technol. 2009 Annu. Conf. North Am. Chapter Assoc. Comput. Linguist. Proc. Conf.*, vol. 1, no. June, pp. 353–361, 2009, doi: 10.3115/1620754.1620806.

[45] K. Ganesan and M. Subotin, "A General Supervised Approach to Segmentation of Clinical Texts," *Proc. - 2014 IEEE Int. Conf. Big Data, IEEE Big Data 2014*, pp. 33–40, 2014, doi: 10.1109/BIGDATA.2014.7004390.

[46] E. Apostolova, D. S. Channin, D. Demner-Fushman, J. Furst, S. Lytinen, and D. Raicu, "Automatic Segmentation of Clinical Texts," 2009, doi: 10.1109/IEMBS.2009.5334831.

[47] T. Edinger, D. Demner-Fushman, A. M. Cohen, S. Bedrick, and W. Hersh, "Evaluation of Clinical Text Segmentation to Facilitate Cohort Retrieval," *AMIA Annu. Symp. Proc.*, vol. 2017, p. 660, 2018.

[48] N. Sadoughi *et al.*, "Detecting Section Boundaries in Medical Dictations: Toward Real-Time Conversion of Medical Dictations to Clinical Reports," in *International Conference on Speech and Computer*, Springer Verlag, 2018, pp. 563–573. doi: 10.1007/978-3-319-99579-3_58/TABLES/5.

[49] F. Ginter, H. Suominen, S. Pyysalo, and T.

Salakoski, "Combining Hidden Markov Models and Latent Semantic Analysis for Topic Segmentation and Labeling: Method and Clinical Application," *Int. J. Med. Inform.*, vol. 78, no. 12, pp. 1–6, 2009, doi: 10.1016/j.ijmedinf.2009.02.003.

[50] A. J. C. Trappey, C. V. Trappey, M. H. Chao, and C. T. Wu, "VR-Enabled Engineering Consultation Chatbot For Integrated And Intelligent Manufacturing Services," *J. Ind. Inf. Integr.*, vol. 26, p. 100331, Mar. 2022, doi: 10.1016/J.JII.2022.100331.

[51] X. Luo, X. Li, Y. M. Goh, X. Song, and Q. Liu, "Application of Machine Learning Technology for Occupational Accident Severity Prediction in the Case of Construction Collapse Accidents," *Saf. Sci.*, vol. 163, p. 106138, Jul. 2023, doi: 10.1016/J.SSCI.2023.106138.

[52] R. Sun, X. Li, J. Shen, and W. Jin, "An Effective Hybrid Automated Chinese Scoring System for Medical Education," *Expert Syst. Appl.*, vol. 234, p. 121114, Dec. 2023, doi: 10.1016/J.ESWA.2023.121114.

[53] D. Yokokawa, K. Noda, T. Uehara, Y. Yanagita, Y. Ohira, and M. Ikusaka, "Do Japanese Word-Embedded Representations Obtained in the Academic Corpus Retain the Medical Concepts of 'Infarction'?," *Artif. Intell. Med.*, vol. 143, p. 102604, Sep. 2023, doi: 10.1016/J.ARTMED.2023.102604.

[54] Z. Zhu, J. Li, Q. Zhao, and F. Akhtar, "A Dictionary-Guided Attention Network for Biomedical Named Entity Recognition in Chinese Electronic Medical Records," *Expert Syst. Appl.*, vol. 231, p. 120709, Nov. 2023, doi: 10.1016/J.ESWA.2023.120709.

[55] P. Wen, Y. Ma, and R. Wang, "Systematic Knowledge Modeling and Extraction Methods for Manufacturing Process Planning Based on Knowledge Graph," *Adv. Eng. Informatics*, vol. 58, p. 102172, Oct. 2023, doi: 10.1016/J.AEI.2023.102172.

[56] C. Y. Wang and J. J. H. Lin, "Utilizing Artificial Intelligence to Support Analyzing Self-Regulated Learning: A Preliminary Mixed-Methods Evaluation from a Human-Centered Perspective," *Comput. Human Behav.*, vol. 144, p. 107721, Jul. 2023, doi: 10.1016/J.CHB.2023.107721.

[57] M. Y. Landolsi, L. Ben Romdhane, and L. Hlaoua, "Medical Named Entity Recognition using Surrounding Sequences Matching," *Procedia Comput. Sci.*, vol. 207, pp. 674–683, Jan. 2022, doi: 10.1016/J.PROCS.2022.09.122.

[58] M. Mirzapour, A. Abdaoui, A. Tchechmedjiev, W. Digan, S. Bringay, and C. Jonquet, "French FastContext: A publicly accessible system for detecting negation, temporality and experiencer in French clinical notes," *J. Biomed. Inform.*, vol. 117, no. December 2020, p. 103733, 2021, doi: 10.1016/j.jbi.2021.103733.

[59] A. A. Shaikina and A. A. Funkner, "Medical Corpora Comparison Using Topic Modeling," *Procedia Comput. Sci.*, vol. 178, no. 2019, pp. 244–253, 2020, doi: 10.1016/j.procs.2020.11.026.

[60] A. Zehe *et al.*, "Detecting Scenes in Fiction: A New Segmentation Task," pp. 3167–3177, Apr. 2021, doi: 10.18653/v1/2021.eacl-main.276.

[61] J. Abbas, C. Zhang, and B. Luo, "EnsCL-CatBoost: A Strategic Framework for Software Requirements Classification," *IEEE Access*, vol. 12, no. October, pp. 127614–127628, 2024, doi: 10.1109/ACCESS.2024.3452011.

[62] F. Alzamzami, M. Hoda, and A. El Saddik, "Light Gradient Boosting Machine for General Sentiment Classification on Short Texts: A Comparative Evaluation," *IEEE Access*, vol. 8, pp. 101840–101858, 2020, doi: 10.1109/ACCESS.2020.2997330.

[63] K. He, R. Mao, T. Gong, C. Li, and E. Cambria, "Meta-Based Self-Training and Re-Weighting for Aspect-Based Sentiment Analysis," *IEEE Trans. Affect. Comput.*, vol. 14, no. 3, pp. 1731–1742, Jul. 2023, doi: 10.1109/TAFFC.2022.3202831.

[64] J. Li *et al.*, "KRA: K-Nearest Neighbor Retrieval Augmented Model for Text Classification," *Electron. 2024*, vol. 13, no. 16, p. 3237, Aug. 2024, doi: 10.3390/ELECTRONICS13163237.

[65] E. Elsaeed, O. Ouda, M. M. Elmogy, A. Atwan, and E. El-Daydamony, "Detecting Fake News in Social Media Using Voting Classifier," *IEEE Access*, vol. 9, pp. 161909–161925, 2021, doi: 10.1109/ACCESS.2021.3132022.

[66] R. Mahadeva, M. Kumar, S. P. Patole, and G. Manik, "Desalination Plant Performance Prediction Model Using Grey Wolf Optimizer Based ANN Approach," *IEEE Access*, vol. 10, pp. 34550–34561, 2022, doi: 10.1109/ACCESS.2022.3162932.

[67] A. Al Bataineh, D. Kaur, and S. M. J. Jalali, "Multi-Layer Perceptron Training Optimization Using Nature Inspired Computing," *IEEE Access*, vol. 10, pp. 36963–36977, 2022, doi: 10.1109/ACCESS.2022.3164669.

[68] M. Alazab, R. A. Khurma, A. Awajan, and M. Wedyan, "Digital Forensics Classification Based on a Hybrid Neural Network and the Salp Swarm Algorithm," *Electron.*, vol. 11, pp. 1–17, 2022, doi: https://doi.org/10.3390/electronics11121903.

[69] M. S. Khan, F. Jabeen, S. Ghouzali, Z. Rehman, S. Naz, and W. Abdul, "Metaheuristic Algorithms in Optimizing Deep Neural Network Model for Software Effort Estimation," *IEEE Access*, vol. 9, pp. 60309–60327, 2021, doi: 10.1109/ACCESS.2021.3072380.

[70] S. Udaiyakumar, C. L. Chinnadurrai, C. Anandhakumar, and S. Ravindran, "Electricity Price Forecasting using Multilayer Perceptron Optimized by Particle Swarm Optimization," *Proc. - 2nd Int. Conf. Smart Technol. Commun. Robot. 2022, STCR 2022*, vol. 2, no. 12, pp. 1–6, 2022, doi: 10.1109/STCR55312.2022.10009414.

[71] N. Zlobinsky, D. L. Johnson, A. K. Mishra, and A.

A. Lysko, "Comparison of Metaheuristic Algorithms for Interface-Constrained Channel Assignment in a Hybrid Dynamic Spectrum Access - Wi-Fi Infrastructure WMN," *IEEE Access*, vol. 10, pp. 26654–26680, 2022, doi: 10.1109/ACCESS.2022.3155642.

[72] A. Melman and O. Evsutin, "Comparative Study of Metaheuristic Optimization Algorithms for Image Steganography Based on Discrete Fourier Transform Domain," *Appl. Soft Comput.*, vol. 132, p. 109847, Jan. 2023, doi: 10.1016/J.ASOC.2022.109847.

[73] T. Nagadurga, R. Devarapalli, and Ł. Knypiński, "Comparison of Meta-Heuristic Optimization Algorithms for Global Maximum Power Point Tracking of Partially Shaded Solar Photovoltaic Systems," *Algorithms 2023, Vol. 16, Page 376*, vol. 16, no. 8, p. 376, Aug. 2023, doi: 10.3390/A16080376.

[74] E. ; Bayona *et al.*, "Comparative Analysis of Metaheuristic Optimization Methods for Trajectory Generation of Automated Guided Vehicles," *Electron. 2024, Vol. 13, Page 728*, vol. 13, no. 4, p. 728, Feb. 2024, doi: 10.3390/ELECTRONICS13040728.

[75] G. Innocenti, F. Corinto, and E. Kaya, "A Comprehensive Comparison of the Performance of Metaheuristic Algorithms in Neural Network Training for Nonlinear System Identification," *Math. 2022, Vol. 10, Page 1611*, vol. 10, no. 9, p. 1611, May 2022, doi: 10.3390/MATH10091611.

[76] S. Ghaemifard and A. Ghannadiasl, "A Comparison of Metaheuristic Algorithms for Structural Optimization: Performance and Efficiency Analysis," *Adv. Civ. Eng.*, vol. 2024, no. 1, p. 2054173, Jan. 2024, doi: 10.1155/2024/2054173.

[77] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[78] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, IEEE, Aug. 1995, pp. 1942–1948. doi: 10.1109/ICNN.1995.488968.

[79] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016, doi: 10.1016/J.ADVENGSOFT.2016.01.008.

[80] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper Optimisation Algorithm: Theory and application," *Adv. Eng. Softw.*, vol. 105, pp. 30–47, Mar. 2017, doi: 10.1016/J.ADVENGSOFT.2017.01.004.

[81] S. Arora and S. Singh, "Butterfly Optimization Algorithm: a Novel Approach for Global Optimization," *Soft Comput.*, vol. 23, no. 3, pp. 715–734, Feb. 2019, doi: 10.1007/S00500-018-3102-4/FIGURES/10.

[82] X.-S. Yang, "Firefly Algorithm, Lévy Flights and Global Optimization," 2010, doi: https://doi.org/10.48550/arXiv.1003.1464.

[83] X.-S. Yang, "A New Metaheuristic Bat-Inspired Algorithm," vol. 284, pp. 65–74, 2010, doi: https://doi.org/10.48550/arXiv.1004.4170.

[84] G. Dhiman and V. Kumar, "Seagull Optimization Algorithm: Theory and its Applications for Large-Scale Industrial Engineering Problems," *Knowledge-Based Syst.*, vol. 165, pp. 169–196, Feb. 2019, doi: 10.1016/J.KNOSYS.2018.11.024.

[85] P. Bajpai, "Genetic Algorithm – an Approach to Solve Global Optimization Problems," *Indian J. Comput. Sci. Eng.*, vol. 1, no. 3, pp. 199–206, 2016.

[86] D. Karaboga and B. Akay, "Artificial Bee Colony (ABC) Algorithm on Training Artificial Neural Networks," *2007 IEEE 15th Signal Process. Commun. Appl. SIU*, 2007, doi: 10.1109/SIU.2007.4298679.

[87] S. Juanita, D. Purwitasari, I. K. E. Purnama, A. F. Abdillah, and M. H. Purnomo, "Multi-Label Classification for Doctor's Behavioral Pattern Matching During Online Medical Interview using Machine Learning," *Int. J. Electr. Eng. Informatics*, vol. 15, no. 3, pp. 435–452, 2023, doi: 10.15676/ijeei.2023.15.3.5.

[88] S. I. G. Situmeang, R. K. Lubis, F. J. N. Siregar, and B. J. D. C. Panjaitan, "Movie Summarization based on Indonesian Subtitles with Restricted Boltzmann Machine," *Proc. 2019 4th Int. Conf. Sustain. Inf. Eng. Technol. SIET 2019*, no. 2014, pp. 338–342, 2019, doi: 10.1109/SIET48054.2019.8986127.

[89] R. C. Morales-Hernández, J. G. Juagüey, and D. Becerra-Alonso, "A Comparison of Multi-Label Text Classification Models in Research Articles Labeled with Sustainable Development Goals," *IEEE Access*, vol. 10, pp. 123534–123548, 2022, doi: 10.1109/ACCESS.2022.3223094.

[90] M. K. Dahouda and I. Joe, "A Deep-Learned Embedding Technique for Categorical Features Encoding," *IEEE Access*, vol. 9, pp. 114381–114391, 2021, doi: 10.1109/ACCESS.2021.3104357.

[91] S. Qiu, Q. Liu, S. Zhou, and W. Huang, "Adversarial Attack and Defense Technologies in Natural Language Processing: A Survey," *Neurocomputing*, vol. 492, pp. 278–307, Jul. 2022, doi: 10.1016/j.neucom.2022.04.020.

[92] W. Yang, R. Zhang, J. Chen, L. Wang, and J. Kim, "Prototype-Guided Pseudo Labeling for Semi-Supervised Text Classification," vol. 1, pp. 16369–16382, doi: 10.18653/v1/2023.acl-long.904.

[93] D. Jiménez, O. J. Gambino, H. Calvo, D. Jiménez, O. J. Gambino, and H. Calvo, "Pseudo-Labeling Improves News Identification and Categorization with Few Annotated Data," *Comput. y Sist.*, vol. 26, no. 1, pp. 183–193, 2022, doi: 10.13053/CYS-26-1-4163.

[94] H. L. Vu, K. T. W. Ng, A. Richter, and C. An,

"Analysis of Input Set Characteristics and Variances on K-Fold Cross Validation for a Recurrent Neural Network Model on Waste Disposal Rate Estimation," *J. Environ. Manage.*, vol. 311, p. 114869, Jun. 2022, doi: 10.1016/J.JENVMAN.2022.114869.

[95] L. Chen *et al.*, "SSD Drive Failure Prediction on Alibaba Data Center Using Machine Learning," *2022 IEEE Int. Mem. Work. IMW 2022 - Proc.*, 2022, doi: 10.1109/IMW52921.2022.9779284.

[96] S. Muhamad Isa, G. Nico, and M. Permana, "Indobert for Indonesian Fake News Detection," *ICIC Express Lett.*, vol. 16, no. 3, pp. 289–297, 2022, doi: 10.24507/icicel.16.03.289.

[97] A. R. Baskara, M. A. S. Jati, M. Maulida, Y. Sari, N. F. Mustamin, and E. S. Wijaya, "Classification of User Reviews for Software Maintenance in Indonesian Language Using IndoBERT-BiLSTM (Case Study: MyPertamina)," *2023 8th Int. Conf. Informatics Comput. ICIC 2023*, 2023, doi: 10.1109/ICIC60109.2023.10381946.

[98] A. B. Y. A. Putra, Y. Sibaroni, and A. F. Ihsan, "Disinformation Detection on 2024 Indonesia Presidential Election using IndoBERT," *2023 Int. Conf. Data Sci. Its Appl. ICoDSA 2023*, pp. 350–355, 2023, doi: 10.1109/ICODSA58501.2023.10277572.

[99] Z. Liu, Q. Zhang, and L. Gao, "Optimizing Sentence Embedding with Pseudo-Labeling and Model Ensembles: A Hierarchical Framework for Enhanced NLP Tasks," Jan. 2025, doi: https://doi.org/10.48550/arXiv.2501.15876.

[100] A. Kag, L. M. Jenila Livingston, L. M. Livingston Merlin, and L. G. X. Agnel Livingston, "Multiclass Single Label Model for Web Page Classification," *2019 Int. Conf. Recent Adv. Energy-Efficient Comput. Commun. ICRAECC 2019*, Mar. 2019, doi: 10.1109/ICRAECC43874.2019.8995087.

[101] B. R. Awangditama *et al.*, "Web Service Classification From Network Flow Using Undersampling Technique," *2023 Int. Conf. Adv. Mechatronics, Intell. Manuf. Ind. Autom. ICAMIMIA 2023 - Proc.*, pp. 1–6, 2023, doi: 10.1109/ICAMIMIA60881.2023.10427874.

[102] D. Kim, J. Koo, and U. M. Kim, "OSP-Class: Open Set Pseudo-labeling with Noise Robust Training for Text Classification," *Proc. - 2022 IEEE Int. Conf. Big Data, Big Data 2022*, pp. 5520–5529, 2022, doi: 10.1109/BIGDATA55660.2022.10020273.

[103] G. Patel, J. Allebach, and Q. Qiu, "Seq-UPS: Sequential Uncertainty-aware Pseudo-label Selection for Semi-Supervised Text Recognition," *Proc. - 2023 IEEE Winter Conf. Appl. Comput. Vision, WACV 2023*, pp. 6169–6179, 2023, doi: 10.1109/WACV56688.2023.00612.

[104] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," Association for Computational Linguistics, Nov. 2019, pp. 3982–3992. doi: 10.18653/v1/D19-1410.

[105] I. Hussain, O. Ormandjieva, and L. Kosseim, "Automatic QQuality Assessment of SRS Text by Means of a Decision-Tree-Based Text Classifier," *Proc. - Int. Conf. Qual. Softw.*, pp. 209–218, 2007, doi: 10.1109/QSIC.2007.4385497.

[106] F. Halim and D. Siahaan, "Detecting Non-Atomic Requirements in Software Requirements Specifications Using Classification Methods," *2019 1st Int. Conf. Cybern. Intell. Syst. ICORIS 2019*, pp. 269–273, Aug. 2019, doi: 10.1109/ICORIS.2019.8874888.

[107] S. Lei, "A feature selection method based on information gain and genetic algorithm," *Proc. - 2012 Int. Conf. Comput. Sci. Electron. Eng. ICCSEE 2012*, vol. 2, pp. 355–358, 2012, doi: 10.1109/ICCSEE.2012.97.

[108] A. Saad, A. P. Engelbrecht, and S. A. Khan, "An Analysis of Differential Evolution Population Size," *Appl. Sci. 2024, Vol. 14, Page 9976*, vol. 14, no. 21, p. 9976, Oct. 2024, doi: 10.3390/APP14219976.

[109] J. Schlauwitz and P. Musilek, "Dimension-Wise Particle Swarm Optimization: Evaluation and Comparative Analysis," *Appl. Sci. 2021*, vol. 11, no. 13, p. 6201, Jul. 2021, doi: 10.3390/APP11136201.

[110] B. Li and R. Y. Xiao, "The Particle Swarm Optimization Algorithm: How to Select the Number of Iteration," in *International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP)*, 2007, pp. 191–193. doi: 10.1109/IIH-MSP.2007.298.

[111] D. Chicco and G. Jurman, "The Advantages of the Matthews Correlation Coefficient (MCC) Over F1 Score and Accuracy in Binary Classification Evaluation," *BMC Genomics*, vol. 21, no. 1, pp. 1–13, Jan. 2020, doi: 10.1186/S12864-019-6413-7/TABLES/5.

[112] P. Stoica and P. Babu, "Pearson–Matthews correlation Coefficients for Binary and Multinary Classification," *Signal Processing*, vol. 222, pp. 1–15, 2024, doi: https://doi.org/10.1016/j.sigpro.2024.109511.

[113] S. Manchanda and G. Karypis, "Text Segmentation on Multilabel Documents: A Distant-Supervised Approach," *Proc. - IEEE Int. Conf. Data Mining, ICDM*, vol. 2018-November, pp. 1170–1175, Dec. 2018, doi: 10.1109/ICDM.2018.00154.

[114] R. Fauzan, D. Siahaan, S. Rochimah, and E. Triandini, "Class Diagram Similarity Measurement: A Different Approach," pp. 215–219, May 2019, doi: 10.1109/ICITISEE.2018.8721021.

[115] M. K. Chegeni, A. Rashno, and S. Fadaei, "Convolution-Layer Parameters Optimization in Convolutional Neural Networks," *Knowledge-Based Syst.*, vol. 261, p. 110210, Feb. 2023, doi: 10.1016/J.KNOSYS.2022.110210.

[116] J. C. Campbell, A. Hindle, and E. Stroulia, "Latent

Dirichlet Allocation: Extracting Topics from Software Engineering Data," *Art Sci. Anal. Softw. Data*, vol. 3, pp. 139–159, 2015, doi: 10.1016/B978-0-12-411519-4.00006-9.

[117] M. Jeong and I. Titov, "Multi-document topic segmentation," in *International Conference on Information and Knowledge Management, Proceedings*, 2010, pp. 1119–1128. doi: 10.1145/1871437.1871579.

[118] D. Marutho, S. Hendra Handaka, E. Wijaya, and Muljono, "The Determination of Cluster Number at k-Mean Using Elbow Method and Purity Evaluation on Headline News," *Proc. - 2018 Int. Semin. Appl. Technol. Inf. Commun. Creat. Technol. Hum. Life, iSemantic 2018*, pp. 533–538, Nov. 2018, doi: 10.1109/ISEMANTIC.2018.8549751.

[119] E. Scheihing, M. Vernier, J. Guerra, J. Born, and L. Cárcamo, "Understanding the Role of Micro-Blogging in B-Learning Activities: Kelluwen Experiences in Chilean Public Schools," *IEEE Trans. Learn. Technol.*, vol. 11, no. 3, pp. 280–293, Jul. 2018, doi: 10.1109/TLT.2017.2714163.

[120] H. G. Daniel Thamrin and T. E. Widagdo, "Natural Language Interface to Database (NLIDB) for Interrogative Sentences with Temporal Elements in Indonesian Language," in *Proceedings of 2022 International Conference on Data and Software Engineering, ICoDSE 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 120–125. doi: 10.1109/ICODSE56892.2022.9971944.

[121] N. Fadhil Abbas, T. Ali Qasim, and H. Abdul-Salam Jasim, "Request Constructions in Classical Arabic versus Modern Arabic," *J. Ethn. Cult. Stud.*, vol. 10, no. 5, pp. 1–15, 2023, doi: 10.2307/48756354.

[122] E. Cahyaningtyas and D. Arifianto, "HMM-Based Indonesian Speech Synthesis System with Declarative and Question Sentences Intonation," in *2015 International Symposium on Intelligent Signal Processing and Communication Systems, ISPACS 2015*, Institute of Electrical and Electronics Engineers Inc., Mar. 2016, pp. 153–158. doi: 10.1109/ISPACS.2015.7432756.

[123] N. F. Nabila *et al.*, "Using Probability Theory to Identify the Unsure Value of an Incomplete Sentence," in *Proceedings - UKSim-AMSS 17th International Conference on Computer Modelling and Simulation, UKSim 2015*, Institute of Electrical and Electronics Engineers Inc., Sep. 2016, pp. 497–501. doi: 10.1109/UKSIM.2015.90.

[124] Y. Lv, "Negative Rhetoric Based on the Error Sentences of Foreign Chinese Learners from the Perspective of Computer Aided Syntax Recognition Algorithms," in *Proceedings of the 2nd International Conference on Artificial Intelligence and Smart Energy, ICAIS 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 704–707. doi: 10.1109/ICAIS53314.2022.9743002.

[125] E. C. Montiel-Vazquez, C. A. Cruz, J. A. R. Uresti, and R. Gomez, "EmpatheticExchanges: Towards Understanding the Cues for Empathy in Dyadic Conversations," *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3520000.

[126] J. M. Müller, "Sympathy, Interpersonal Awareness and Acknowledgment," *Topoi*, vol. 41, no. 5, pp. 849–858, Nov. 2022, doi: 10.1007/S11245-022-09803-3/METRICS.

YOLOv8 and ResNet-50 Based Real-Time Fabric Defect Detection and Quality Grading System

Abdul Muchlis, Eri Prasetyo Wibowo, Rudi Irawan, Afzeri
Departement Information Technology, Gunadarma University, Depok, Indonesia,
Departement Mechanical Engineering, Gunadarma University, Depok, Indonesia
Department Mechanical Engineering, Indorama Engineering Polytechnic, Purwakarta, Indonesia
E-mail: muchlis07@staff.gunadarma.ac.id, eri@staff.gunadarma.ac.id, Rudi@staff.gunadarma.ac.id, afzeri.pei.ac.id

Keywords: YOLOv8, ResNet-50, fabric quality, automatic inspection, 4-Point System

Received: July 5, 2025

Automated fabric defect inspection is essential to address the low speed and inefficiency of manual inspection, which often requires machine stoppages to record defects. In contrast, this study presents a continuous, real-time defect detection and grading system based on the 4-point standard, integrating computer vision and deep learning techniques. A dataset of 1,456 annotated fabric images was developed from captured production videos. The proposed method employs ResNet-50 as the backbone for feature extraction and YOLOv8 for object detection, followed by defect quantification to determine fabric grades. Experimental evaluation shows that the model achieves a precision of 0.789, recall of 0.554, mAP50 of 0.675, mAP50-95 of 0.461, and an F1-score of 0.61. The class-wise mAP50 scores are 0.744 for line defects, 0.839 for stain defects, and 0.442 for hole defects. While the integration of convolutional neural networks with automated grading logic enhances detection accuracy and reliability in textile quality control, the relatively low performance on hole defects indicates the impact of dataset imbalance, as this category contained significantly fewer samples. Addressing this limitation through targeted data augmentation or additional data collection is recommended to improve generalization and achieve balanced detection performance across defect categories.

Povzetek: Članek obravnava problem počasnega in nezanesljivega ročnega pregleda tekstila. Predlaga metodo, ki združuje YOLOv8 z ogrodjem ResNet-50 za sprotno detekcijo napak ter samodejno razvrščanje kakovosti po 4-točkovnem sistemu. Model učinkovito zazna madeže in črte, a slabše luknje zaradi neuravnoteženega nabora podatkov.

1 Introduction

The integration of advanced information technologies particularly artificial intelligence (AI), machine learning (ML), and deep learning (DL) has fundamentally transformed various stages of the global textile manufacturing process. These technologies have enabled real-time defect detection, predictive maintenance of machinery, intelligent quality control, and end-to-end automation, thereby significantly enhancing operational efficiency, product consistency, and decision-making accuracy across the textile supply chain [1], [2], [3]. The application of this technology enables the machines to automatically detect defects in fabric, identify complex patterns, and improve product quality without human intervention [4]. Deep learning models, such as Convolutional Neural Networks (CNN), are widely used in the visual inspection to detect imperfections in fabrics with higher accuracy compared to the conventional methods, thereby helping to speed up the quality control processes and to increase the effectiveness of fabric defect detection. According to the Allied Market Research report, the market value of AI in the textile industry is expected to reach $4.8 billion by 2031, with a compound annual growth rate (CAGR) of 26.4% from 2022 to 2031, indicating a significant growth in the adoption of this technology.

The application of AI in the industry 4.0 and the smart manufacturing in the textile sector have transformed the traditional production processes by increasing efficiency, reducing errors and optimizing production workflows. AI-based technologies, such as machine learning, computer vision, and robotics, are being integrated into various stages of textile production, enabling the manufacturers to automate complex tasks, reduce waste, and increase productivity [5]

The textile industry has experienced a major transformation with the application of artificial intelligence (AI), especially in fabric defect detection. A fabric defect is a flaw in a material that degrades its quality, reducing its value and utility.

Once these defects are on the markets, they cannot be repaired or returned. Therefore, ensuring defect-free products is essential to produce high-quality goods [6]. Quality control plays an important role in ensuring customer satisfaction and reducing production costs [7], [8]. Traditional approaches to fabric inspection often rely on manual processes that are time-consuming and susceptible to human errors. This process involves visual

observation by the operator which not only requires high concentration but is also difficult to maintain for long periods of time. As a result, errors in detecting fabric defects, such as holes, lines and stains, become common, which ultimately affects product quality.

The development of technological advances in the field of artificial intelligence, especially deep learning, has opened up new opportunities to improve the accuracy and efficiency of fabric defect detection. One appropriate technology is the YOLO (You Only Look Once) model. YOLO is designed to detect objects in one processing stage, so that it is able to provide fast detection results. With its detection speed, YOLO is very suitable for use in the textile industry which requires a fast and efficient inspection process [9], [10]. In addition, YOLO implementation can be integrated with other automation systems to create a solution that can reduce dependence on human labor and increase consistency in fabric defect inspection.

Detecting fabric defects with a YOLO-based approach method, the textile industry can optimize the quality control process and meet higher quality standards. This technology not only addresses the challenges of manual inspection but also represents a step forward towards digitalization and automation in the modern textile industry. Fabric defect detection using deep learning has been carried out using 2 methods in terms of the process, namely Double Shot Detector and Single Shot Detector [11]. The previous research using double shot detectors such as that conducted by research using a deep learning-based framework with a two-stage strategy, used the CNN Inception-V1 architecture to detect local defects and the LeNet-5 model to recognize the type of defect. This framework achieved 96% accuracy, 97% detection rate, and 6% false rate, demonstrating the great potential of deep learning in supporting the automation and precision of fabric defect detection [13]. The use of the YOLO and Single Shot Detector (SSD) architectures differs in that YOLO produces a set of object predictions across the entire images, while SSD produces object predictions at the various resolution levels [11]. In terms of speed, the YOLO architecture is faster than SSD but can provide better detection accuracy for objects of various scales and shapes [13]–[17].

Based on the previous research, it can be concluded that the development of deep learning technology has become the right step in detecting defects in fabrics. This technology is able to replace the role of human inspectors in the process of automatically detecting fabric defects, which ultimately increases the efficiency in terms of both time and production costs. By leveraging the advantages of deep learning [17], [18], [19], the textile industry can accelerate the fabric inspection process while ensuring higher accuracy, resulting in products with more consistent quality.

The 4-point system provides a way to determine defects according to their severity by assigning demerit or penalty holes. This system falls under ASTM D5430- 93, a standard test method for determining fabric grading [20]. This system is used by INDORAMA SYNTHETICS LTD. in conducting fabric defect inspections, although the weaving process is carried out in a modern way with automated weaving machines, there are still defects that occur, then an inspection is carried out by an inspector with a fabric length of 100 yards or 1 roll of fabric with a width of 1.8 meters.

Previous studies have generally focused on the application of deep learning models using image processing technology but have not yet reached the stage of fabric quality classification based on the 4-point system [19], In contrast, this study integrates both hardware and software components to develop an automated deep learning-based system capable of detecting and directly classifying fabric quality according to the 4-point grading standard. This integration can be seen in the research scheme in Figure 1, the fabric video was taken using an Intel RealSense D435i camera located above the fabric during the data acquisition process using a DC motor, then the data were processed into a dataset for the model training process. Once the model is good, the quality control process can be directly processed by integrating the system as in Figure 1.

Based on these issues, this study is focused on addressing two main research questions: (1) Can the YOLOv8 model combined with a ResNet-50 backbone detect fabric defects in real-time with sufficient accuracy? and (2) Can the proposed system automatically classify fabric quality using the 4-point system based on CNN predictions? These research questions provide a structured direction for system development and serve as the foundation for the evaluation framework in this study.

2 Literature review

Fabric defect detection is an important aspect of quality control in the textile industry. Automation in this process plays a vital role in improving the efficiency and consistency of an inspection. Convolutional Neural Network (CNN), especially the YOLO architecture, has been widely used for real-time object detection tasks due to its high accuracy and speed. Various previous studies have used the YOLO model to detect fabric defects, focusing on the model augmentation and modification techniques to improve the performances [21], [22].

2.1 Acquisition from video

The data collection process was carried out by recording the moving fabric using an Intel RealSense D435i camera. The video footage is then processed into individual image frames which are used to train the YOLOv8 model. In contrast to the static image datasets, this method allows a dynamic data acquisition, so that the fabric defects can be captured from the various fabric orientations and lighting conditions similar to the real-world conditions. The rotational and translational speeds of the fabric are precisely controlled to ensure a uniform coverage of the fabric area [23]. The fabric translational speed is 4.07 meters/minute.

2.2 Augmentation techniques

In 2022, Yue, Wang utilized the improved YOLO-based small target detection technology. This research

uses data augmentation such as mosaic, impulse noise, Gaussian Blur, mirror flip, and affine transform, as well as anchor regrouping in the added dataset. Meanwhile, in this research, the imbalanced datasets were overcome by increasing and balancing the number of datasets in each class, and the augmentation techniques such as flipping, rotation, and contrast adjustment were applied. This augmentation is in line with the previous research [15], [21], where a similar method was proven to be effective in increasing the detection accuracy. The results show an increase in the small target detection accuracy of up to 8% with a mAP of 56.68% compared to the base model of 52.17%, especially in the hole and knot defects.

2.3 Deep learning model

Research on textile defect detection has advanced rapidly with the evolution of the *You Only Look Once* (YOLO) architecture. Wang improved YOLOv4 by incorporating SoftPool into the Spatial Pyramid Pooling (SPP) module and applying *contrast-limited adaptive histogram equalization* (CLAHE), achieving a 6% increase in mAP over the standard YOLOv4 [24]. In the same year, Wang combined YOLOv5 for defect detection with ResNet for classification in automotive *heat staking*, attaining a detection mAP of 95.1% and a classification F1-score of 98%, though at the cost of increased computational complexity due to its two-stage nature [25]

Later works shifted to optimizing YOLOv8. Li proposed FF-YOLO, a lightweight variant with Slim Attention Module (SimAM) and Lightweight Multi-Level Asymmetry Detector Head (LADH), which maintained high accuracy (mAP@0.5 of 75.5%) while reducing computational cost [26]. In 2025, DA-YOLOv8s integrated Deformable Convolution v2 (DCNv2) and Polarized Self-Attention (PSA), improving feature representation and achieving high-speed detection (257.38 FPS)[27].

This study extends these developments by embedding ResNet directly into YOLOv8 as its backbone, enabling *end-to-end* detection with deeper feature extraction from the earliest stages. This integration aims to improve accuracy for small or subtle textile defects while preserving real-time performance, bridging the gap between hybrid YOLO–ResNet pipelines and fully integrated, high-performance YOLO models.

Year	Study Title	Authors	Methodology	Key Results	Main Contribution
2022	A Fabric Defect Detection Method Based on Deep Learning	Wang et al.	YOLOv4 + SoftPool + CLAHE	mAP ↑ 6% vs. YOLOv4	Improved textile defect accuracy via enhanced feature pooling and preprocessing.
2022	Application of YOLO and ResNet in Heat Staking Process Inspection	Wang et al.	YOLOv5 + ResNet	mAP: 95.1%; F1: 98%	Combined fast YOLO detection with accurate ResNet classification for industrial inspection.
2024	FF-YOLO: Fashion Fabric Detection Algorithm Based on YOLOv8	Li et al.	YOLOv8 + SimAM + LADH	mAP@0.5: 75.5%; FLOPS ↓ 5%	Maintained high accuracy while reducing computational complexity.
2025	Textile Defect Detection Algorithm Based on the Improved YOLOv8	Song W, Lang D	YOLOv8 + DCNv2 + PSA	mAP@0.5: 44.6%; FPS: 257.38	Enhanced spatial channel feature representation for high-speed detection.
2025	**Proposed Model**		YOLOv8 + ResNet (backbone)	mAP@0.5: 67.5%	Integrated ResNet backbone for *end-to-end* deep feature extraction and real-time textile defect detection.

2.4 Four point system

The 4-Point system or four-point system is an approach often used in the textile industry to evaluate fabric quality based on detected defects. This system provides comprehensive metrics for assessing fabric condition by considering four key dimensions: defect density, defect distribution, defect size, and defect severity. This approach not only assists in a decision making in the production process but also becomes a widely accepted quality standard [28]. This assessment uses a penalty system in the form of a score from 1 to 4 based on the severity and size of the defects that occur. The application of the 4-point system by converting the accumulation of the number, size and severity of defects with total holes. These holes are used to classify the quality grade of the fabric, which is calculated based on the maximum length of the bounding box.

Formulation for calculating bounding box[26]–[28].

$$w, h = x_2 - x_1,\ y_2 - y_1 \quad (1)$$

Then the pixel length is obtained which is used to convert the actual size to pixels in the images using the DPI (Dots per Inch) formula which is shown in equation.

$$Pixel\ Length\ (cm) = \frac{Pixel\ length}{\text{DPI}}\ x\ 2.54 \quad (2)$$

$$DPI = \frac{Pixel\ Widht}{Fabric\ Widht\ (inch)} \quad (3)$$

3 Method

Research on the development of a system of a fabric quality ranking device based on Deep Learning can be seen in Figure 1. The automation tool system is designed for taking images of moving fabric using an Intel RealSense D435i camera, producing a video which is then processed by a machine learning model to detect defects and determine fabric quality using a 4-point system. The research methodology is divided into 7 parts of the process, as follows:

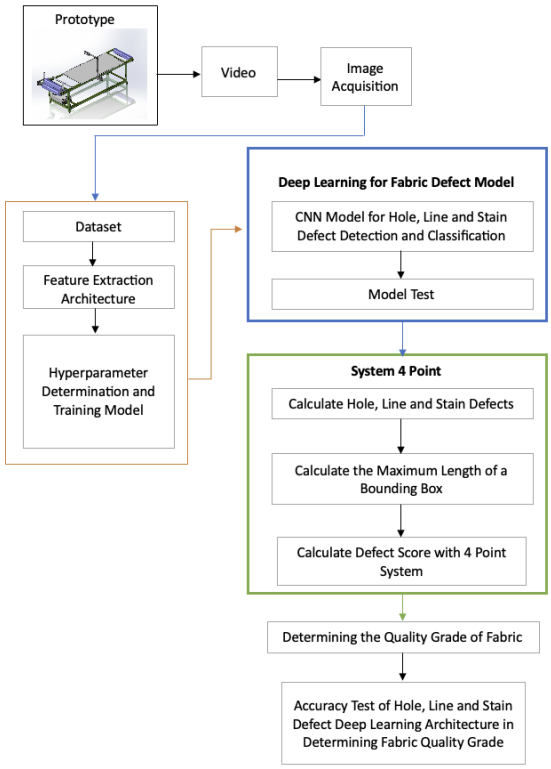


Figure 1: Framework's research

3.1 System design

Prototyping of fabric automation tools connected to an Intel RealSense D435i camera produces video and is processed for image acquisition.

3.2 Creating a classification model

3.2.1 Dataset acquisition

The image obtained from the camera acquisition with fabric from INDORAMA SYNTHETICS LTD. having been adjusted to the system produces video with a duration of 1.57 minutes converted into images producing 3,330 images. The 3,330 images were then manually selected by selecting images with hole, line and stain defects, resulting in 1,456 images which were used as datasets for the model training and testing processes. The images are augmented to make the dataset balanced in each defect class. The techniques used are as follows:

a) Flip: Horizontal and vertical
b) 90° Rotation:Clockwise and counterclockwise
c) Rotation 15° to - 15
d) Exposure Bounding Box: -10% to 10%
e) Bounding Box Brightness: -15% to 15%

3.2.2 Feature extraction architecture

Replacing the YOLOv8 backbone with ResNet-50 enhances feature representation, particularly for inputs with complex textures such as fabric defects. The ResNet-50 blocks are integrated before the neck module, allowing deeper semantic features to be propagated to the YOLOv8 detection head while maintaining inference efficiency, this can be seen in Figure 11. This architectural modification improves generalization, especially in detecting subtle texture anomalies, without compromising detection speed or overall performance.

3.2.3 Hyperparameters

Determination of Hyperparameters and Training the Determination of the hyperparameters used is:

a) Batch size 32
b) Learning Rate 0.001429
c) Epoch 50, 100, 150 and Early Stop
d) IoU 0.3

3.2 Detection and classification machine learning for hole, line and stain defects

Although at first glance Algorithm 1 may appear to reiterate standard YOLO detection procedures, it actually integrates essential post-processing stages that are critical to this study, particularly within the context of the textile industry. The algorithm not only counts the number of each defect type (holes, lines, and stains) but also calculates severity levels based on the 4-point fabric grading system, including the assignment of weight to specific defect categories. Moreover, it computes the maximum defect length by analyzing the bounding box coordinates, which serves as a key reference in the quantification and classification of fabric quality.

Algorithm 1: Defect detection and counting

Step 1: Input: Model (img, stream=True, iou = 0.3)
Step 2: Calculation hole, line, stain:
classCount = class name:[] for class name in classNames
Count hole = 0
Count Line = 0
Count stain = 0
Step 3: Count Defects
if currentClass = 'Hole' then
count_hole += 1
total_score += 1
else if currentClass = 'Line' then
count_line += 1
weight ← calc_weight(defect_length(cm))
total_score += weight
else if currentClass = 'Stain' then
count_stain += 1
weight ← calc_weight(defect_length(cm))
total_score += weight
end if
Step 4: Calculate Maximum Bounding Box Length
w, h = x2 - x1, y2 - y1
defect_length_pixel = max(w, h) = 0

4 Scheme research

The system of the Deep Learning-based fabric quality level detection automation tool begins with the initial step of building a system of the defect detection image capture hardware. The integration of the hardware systems can be seen in Figure 2,

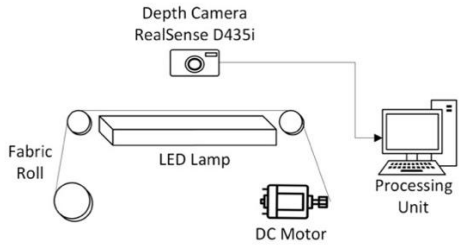


Figure 2: Scheme research

This scheme shows an automated fabric inspection system consisting of several main components. A RealSense D435i Depth Camera is used to capture precise images of the fabric. The camera is connected to a Processing Unit (computer) that processes the image data to detect the defects. The LED Lamp provides even lighting on the fabric to improve the accuracy of the camera's capture. Fabric Roll rolls the fabric, while DC Motor moves it continuously so that its entire surface can be inspected. This system is designed to automatically detect defects in fabric.

4.1 System design of automatic device capturing fabric images

The system device has dimensions and design as shown in Figure 3 which is used for video acquisition. The size or dimensions of the device have a device length of 100 cm and a device width of 30 cm and a device height of 25 cm. The camera position is perpendicular to the cloth with a height of 25 cm and the cloth is rolled at the same speed from the shaft driven by the electric motor, which is 4.07 meters / minute.

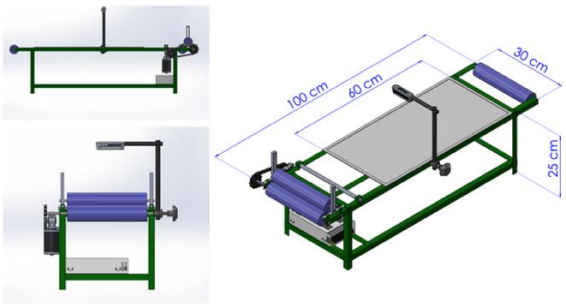


Figure 3: Device design

4.2 Manufacturing system of fabric image capturing device

The automatic image capturing device moves the fabric with a DC electric motor, while the fabric is rolled using a roller shaft. The camera is placed above the device while the fabric running to capture the fabric images and send them to the machine learning model for hole, line and stain defect detection and classification.

The sample of fabric obtained from INDORAMA SYNTHETICS LTD. is 5 meters long and 1.8 meters wide and it is adjusted to the system device which is 60 cm long and 30 cm wide. The results of the fabric adjusted to the system are 2 rolls of fabric of 15 meters each.

Figure 4: Aqquisition instrument

Table 1: Instrument design

Parameter	Value
Tool length	100 cm
Tool width	30 cm
Fabric board length	60 cm
Tool height	25 cm
Light width	30 cm
Sample Fabric width	1.8 meters = 180 cm
Sample Fabric length	5 meters
The ratio of fabric width to lamp width	180/30 = 6
Length of dataset fabric	0.6 m × 5 m = 30 m
Bearing type	Plain Bearing, diameter 40 mm
Power supply	220V to 12V 20A
Motor shaft diameter	9 mm
End cylinder shaft diameter	19 mm
Frame material	Angle Iron size 4×4 cm
RPM	35 revolutions in 1 minute
Time	3 minutes
Roll circumference	π × Roller diameter
Circumference	π × 3.7 cm = 11.64 cm

4.3 Video

During the fabric rolling process, video acquisition is performed using an Intel RealSense D435i camera mounted vertically above the fabric surface at a fixed height of 30 cm to ensure consistent top-down imaging. The resulting video has a duration of 1 minute 51 seconds with 30 frames per second.

4.4 Primary dataset

Prior to conducting the video analysis, a dataset was prepared using defective fabric samples provided by PT. INDORAMA. These fabric samples varied in length but consistently maintained a width of 1.8 meters. For systematic data acquisition, the fabric width was divided into six equal sections, each measuring 30 centimeters. These segmented widths were then reassembled longitudinally to form a continuous dataset. This configuration ensured consistent coverage and alignment of the defective areas along the entire fabric length.

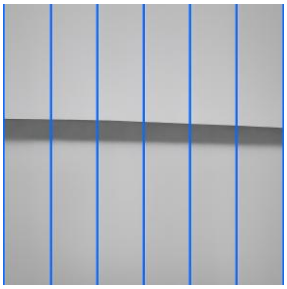


Figure 5: Symmetric sectioning of fabric image with indexed columns

Figure 5 illustrates the segmentation of fabric into six symmetrical vertical sections, each labeled sequentially from left to right. Following this segmentation, each section is vertically concatenated along the fabric's longitudinal axis to reconstruct a complete and continuous dataset image. This vertical merging process allows for a comprehensive and systematic analysis of defects distributed throughout the entire fabric surface the video results are frame extracted and converted from RGB format to Greyscale. The results of frame extraction from the video are the number of frames in the video of 3,330 images with a size of 1,568 pixels x 880 pixels.

4.5 Dataset preparation and processing

To improve the replicability of this study, we provide a detailed overview of the dataset preparation process. First, image data were captured by framing video segments from a real-time fabric inspection system. These images were then annotated using Roboflow, where preprocessing steps such as resizing, format conversion, and augmentation were applied as described in Subsection 3.2.1. The dataset was subsequently divided into training and validation sets using an 80:20 ratio through a custom Python script in Figure 6 ensuring no overlap with test data. For model testing, we used a different fabric roll than the one used in the training and validation process, ensuring independent evaluation.

```
# Determine the proportion of 80% for training and 20% for validation.
train_size = int(0.80 * len(all_images))
train_images = all_images[:train_size]
val_images = all_images[train_size:]

def copy_files(images_list, dest_images_path, dest_labels_path):
    for image_name in images_list:
        # Check if the image file is already in the test folder
        if os.path.exists(os.path.join(test_images_path, image_name)):
            print(f"Skipping {image_name} as it already exists in the test folder.")
            continue

        # Copy image files
        shutil.copy(os.path.join(images_path, image_name), os.path.join(dest_images_path, image_name))

        # Copy the appropriate label file if it is not already in the test folder.
        label_name = os.path.splitext(image_name)[0] + '.txt'
        label_source_path = os.path.join(labels_path, label_name)
        if os.path.exists(os.path.join(test_labels_path, label_name)):
            print(f"Skipping label {label_name} as it already exists in the test folder.")
            continue

        if os.path.exists(label_source_path):
            shutil.copy(label_source_path, os.path.join(dest_labels_path, label_name))
        else:
            print(f"Label for {image_name} not found!")

# Copy data to train folder, if it is not in test
copy_files(train_images, output_train_images, output_train_labels)

# Copy data to validation folder, if it is not in test
copy_files(val_images, output_val_images, output_val_labels)

print("Dataset successfully split!")
print(f"Train set: {len(train_images)} images")
print(f"Validation set: {len(val_images)} images")
```

Figure 6: Split data

To enhance feature representation, particularly for subtle defect textures, the default YOLOv8 backbone was replaced with the pretrained ResNet-50 architecture. This modification leverages the residual learning capability of ResNet-50 to extract more discriminative and hierarchically rich features at the early stages of the network. Integration was performed by directly feeding the output of ResNet-50 into a series of convolutional layers designed to mimic the detection head of YOLOv8. These layers were configured to preserve spatial resolution and detection accuracy, especially for detecting small-scale defects such as holes and stains. This backbone replacement proved to yield more accurate bounding box predictions and clearer class separation, as reflected in the evaluation results on the validation data.

```
# Create an input layer for the new model
input_layer = layers.Input(shape=(640, 640, 3))

# Put input into the ResNet-50 backbone
x = resnet_backbone(input_layer, training=False)

# Adding YOLOv8 head
# YOLOv8 usually uses multiple convolutional layers, we can emulate a similar architecture

x = layers.Conv2D(1024, (3, 3), padding='same', activation='relu')(x)
x = layers.MaxPooling2D((3,3), strides=(2,2), padding='same')(x)
x = layers.Conv2D(512, (1, 1), padding='same', activation='relu')(x)
x = layers.MaxPooling2D((3,3), strides=(2,2), padding='same')(x)
x = layers.Conv2D(256, (1, 1), padding='same', activation='relu')(x)
x = layers.Conv2D(128, (1, 1), padding='same', activation='relu')(x)
output_layer = layers.Conv2D(3, (1, 1), activation='softmax')(x)

# Combine inputs and outputs into a model
model_with_resnet = models.Model(inputs=input_layer, outputs=output_layer)

# Model Compilatio
model_with_resnet.compile(optimizer='adam', loss='crossentropy')
dataset_path = '/content/data.yaml'
model_with_resnet.summary()
model_with_resnet.fit(data=dataset_path, epochs=100, imgsz=640, lr0=0.0001)

```

Figure 7: YOLOv8 with ResNet-50 Backbone

This Figure 7 illustrates the architecture and training configuration of a deep learning model that combines the ResNet-50 backbone with a YOLOv8 detection head. The model is built using TensorFlow, starting with an input layer of 640 x 640 pixels, followed by multiple convolutional layers that emulate the YOLOv8 structure for object detection. The model is then compiled using the Adam optimizer and categorical crossentropy loss function, and trained for 100 epochs using a dataset in YAML format. This approach is designed to support efficient and automated fabric defect classification.

4.6 Classification model creation

Dataset acquisition is prepared before the data is processed in the learning model. The dataset images are selected manually from 3,330 existing images to become 1,456 bounding box images, divided into 80% training data, namely 1,165 images and 20% validation data, namely 291 images. The image size is resized to 640 pixels x 640 pixels in all dataset images.

The architecture can be seen in Figure 12 and explained as follows. The images pass through a convolution layer with Batch Normalization, ReLu activation function and Maxpooling, which reduces the image dimension to 320 x 320 with 64 features. The images go through several ResNet blocks to extract the features, with the first block producing an output of 160 x 160 with 256 features and the last block producing 20 x 20 with 2048 features, showing a decrease in dimension and an increase in feature depth. After the ResNet-50 backbone, down-sampling is performed through the convolution layer, reducing the features from 2,048 to

1,024, then the C2F block is applied to produce an output of 1024 features, followed by down-sampling to 512 features. The SPPF block then combines spatial information from multiple scales, producing an output of 1,024 features. The final detection layer predicts bounding boxes, objectness scores, and class labels, allowing the model to efficiently detect objects at multiple scales and produce accurate predictions.

ResNet is realized by adding skip connections between layers. These connections are element-by-element additions between the input and output blocks and do not add extra parameters or computational complexity to the network [23].

Determination and Training Epoch is used to see how many times the model sees the entire dataset during the training. The number of epochs is how many times the entire dataset is used to train the model. In each epoch, the model learns from the data repeatedly to improve the performance [30], [32]. The epochs used with 3 variations are epochs 50, 100 and 150. All three epochs were performed at a learning rate of 0.001429.

4.7 Machine learning detection and classification of hole, line and stain defects

a) CNN Model for Hole, Line and Stain Defect Detection and Classification
The model obtained from the classification model creation process is used to detect and classify the results of processing fabric image acquisition. The model results are detected images with an IOU value of 0.3 for the classification of 3 classes of fabric defects, namely lines, stain and holes. YOLO with its advantages can detect and classify objects [33].

b) Model test
A model testing uses the mean average precision (mAP) formula in the object detection by predicting the bounding box produced by the model compared to the ground truth box using an intersection over Union (IoU) [30], [31]. For each object class, the average precision (AP) value is calculated by taking the average of Precision over various Recall values. This AP is the main metric used to evaluate the performance of the model in detecting objects of a particular class. After AP is calculated for each class, mAP is calculated by taking the average of all these AP values. This mAP value reflects how well the model can detect objects, combining the information about precision and sensitivity in the object detection at various scales and classes, mAP provides a comprehensive picture of the model's performance in detecting the objects accurately and consistently across the datasets.

c) Calculation of Hole, Line and Stain Defects
The model can be used to detect and classify hole, line and stain defects. The next step is to calculate each defect. This calculation aims to process the calculation into a 4-point system which is an accumulation of the total defects that occur with the dimensions of the defects. Based on the total and dimensions of the defects, they are entered into a scale hole with a value of 1 to 4.

d) Calculation of the Maximum Length of the Bounding Box
Bounding box is an important element in the object detection in the form of hole, line and stain defect detection in the fabric images. Bounding box is calculated following the original size of the fabric to represent the size of the defect in the real image. The calculation that converts from pixels to the actual defect size uses DPI (Dots per Inch). The width w and height h values of the bounding box are calculated based on the difference in coordinates between the bottom-right and top-left holes of the bounding box. The max (w, h) function is then used to obtain the larger value between the width and height, which is called the maximum length of the bounding box in pixels. The function of converting pixel values to cm uses the formula (pixel length / DPI) x 2.54, where DPI (Dots Per Inch) is the scale factor used to determine the conversion from pixels to physical size [34].

e) Calculating Handicap Score With 4-Point System
The defect score according to the classification found is calculated using a 4-point system. Inspection in the research focuses on the inspection of the construction quality which is sourced from how the fabric itself is made. Fabric inspection is based on its production. With a 4-point system, defects in the warp/weft direction or strip/wale direction in a fabric will be given penalty holes using the criteria based on the American Society for Testing and Materials (ASTM) D5430-93 standard, as in Table 2.

Table 2: Penalty defect 4-point system

No	Defect Size	Penalty Point
1	Defect less than 3 inches	1
2	Defect more than 3 inches, but not more than 6 inches	2
3	Defect more than 6 inches, but not more than 9 inches	3
4	Defect more than 9 inches	4

f) Determining the Fabric Quality Level
Calculation of total holes per yard in the 4-point system, the quality of the fabric is evaluated in holes/100 square yards. The quality grade of the fabric is based on the number of holes against fabric defects with a length of Holes per 100·yard^2. This rule in attachment by INDORAMA SYNTHETICS LTD.

$$\text{Point} = \frac{\text{Total roll point x 6 x 100}}{\text{Fabric length (yard)x fabric weidht (inches)}} \quad (4)$$

The results of calculating the score using a 4-point system for fabric defects are used in the process of determining the fabric quality level (grade) by following the rules in Table 3.

Table 3: Fabric quality grade

No	Fabric Quality Grade	Total Defect
1	Grade AE	<30
2	Grade A	30 < A<40
3	Grade BE	>40

g) Accuracy Testing of Machine Learning Architecture for Detecting Hole, Line, and Stain Defects in Fabric Quality Grade

A system learning machine for detecting dot, line and soil defects in fabric has been completed and now requires testing to evaluate the accuracy of prediction and determination of fabric quality grade. The testing steps include calling the confusion_matrix (y_true, y_pred, labels) function to calculate the confusion matrix based on three main parameters, namely y_true (the actual label), y_pred (the label predicted by the model), and labels (the list of all classes). The calculation results of the function are stored in the variable cm, and the cm value is returned for further analysis.

5 Result

The Intel RealSense D435i camera captures fabric images in RGB format, generating video files. In this research, the recorded video has a duration of 1 minute and 51 seconds, with the automation tool operating at a speed of 4.07 meters per minute. The resulting video resolution is 1,568 × 880 pixels.

5.1 Results of classification model creation

The acquisition of an image dataset with a size of 640 pixels x 640 pixels is divided into 80% training data and 20% validation data from 1456 bounding box images. The distribution of the images in the training and validation processes can be seen in Figure 9.

Figure 8: Sample images

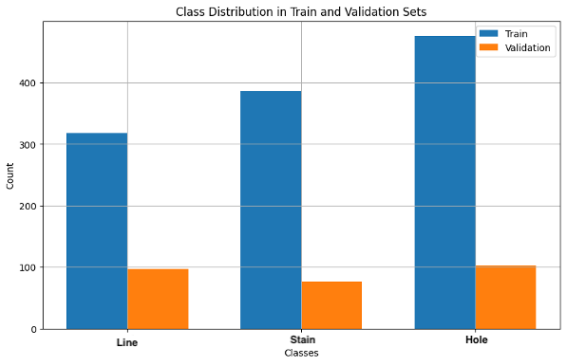


Figure 9**:** Distribution data

Table 4: Distribution results of training data and validation data

Class	Total Training Data	Total Validation Data	Total data	Training Data (%)	Validation Data (%)
Hole	476	102	578	82.4	17.6
Stain	386	77	463	83.4	16.6
Line	318	97	415	76.6	23.4

Table 4 shows the results of the distribution of training data and validation data on 3 defect classes, namely Hole, Line and Stain. Line defects, the number of training data used was 318 images, validation 97 images, the total image data used was 415 images, or 76.6% training data and 23.4% validation data if expressed in a percentage. Stain Defects the number of training data was 386 and validation 77, so that the total images used are 463, the percentage of training data is 83.4% and validation data is 16.6%. Hole Defects used 476 training data and 102 validation data so that the total is 578 so the percentage of training data is 82.4% and validation data is 17.6%.

Table 5: Impact of data split proportions on precision, recall, and mAP Values

Train Comp	Val. Comp	Precision	Recall	mAP	mAP50-95
70%	30%	0.718	0.647	0.705	0.462
80%	20%	0.729	0.646	0.704	0.464
90%	10%	0.743	0.562	0.658	0.413

The results presented in Table 5 demonstrate that variations in the proportion of training and validation data significantly influence the model's performance. As the proportion of training data increases from 70% to 90%, precision improves steadily (from 0.718 to 0.743), indicating the model's growing confidence in its predictions.

However, this comes at the cost of recall, which declines from 0.647 to 0.562, suggesting that the model becomes increasingly conservative and may fail to detect some true positives. Similarly, while mAP50 remains relatively stable between the 70:30 and 80:20 splits (0.705 and 0.704), it drops to 0.658 in the 90:10 split, implying reduced generalization when the validation set becomes too small. Notably, mAP50-95 shows a consistent downward trend as the validation portion decreases, from 0.462 at 70:30 to 0.413 at 90:10. These findings suggest that the 80:20 split achieves the best trade-off between precision, recall, and generalization. Conversely, the 70:30 split may be more appropriate when prioritizing recall and broader generalization performance.

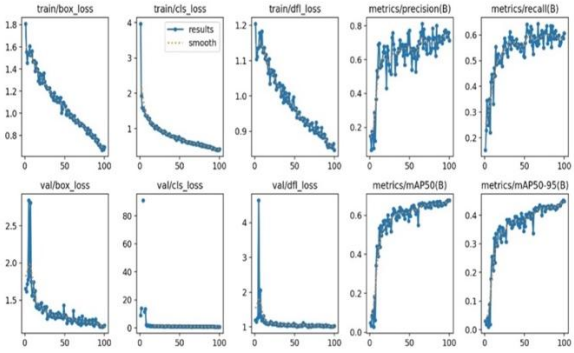


Figure 10: Performance graph at epoch 100

Figure 10 provides a comprehensive visualization of the model's training and validation behavior over 100 epochs. The loss curves covering box loss, classification loss, and DFL (distribution focal loss) exhibit a consistent and smooth downward trend in both training and validation phases, indicating effective convergence and stability of the learning process. The sharp decline in validation losses during the initial epochs reflects rapid learning, while the subsequent plateau suggests the model has reached an optimal learning state with minimal overfitting. The performance metrics precision, recall, mAP50, and mAP50-95 all show a gradual improvement, particularly precision and mAP50, which stabilize above 0.75 and 0.70, respectively, by the end of training. However, recall and mAP50-95 demonstrate more fluctuation, suggesting sensitivity to class imbalance and the presence of harder-to-detect defect types. This graphical evidence reinforces earlier tabular analysis, validating that the model's performance peaks around epoch 100 with the 80:20 data split, achieving strong precision while balancing generalization. The curve patterns also highlight that extending training beyond 100 epochs may not yield significant gains, thus confirming epoch 100 as an optimal stopping point.

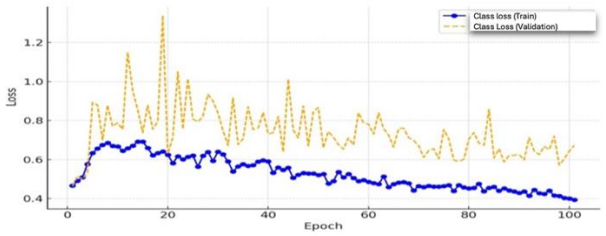


Figure 11: Loss train and valid graphs

Figure 11 shows the training and validation class loss curves. The training loss consistently decreases across epochs, indicating effective learning and convergence. However, the validation loss fluctuates with noticeable peaks, suggesting difficulty in generalization possibly due to data distribution differences or early signs of overfitting. The consistently higher validation loss compared to training loss also reflects a performance gap. This instability indicates the model has not yet fully generalized and could still be improved.

5.2 Architectural results of feature extraction

This architecture processes 640x640 pixel RGB fabric images using a ResNet-50 backbone [35]. The input passes through an initial convolutional layer and ResNet blocks 1-4 to extract features at various scales, with output dimensions ranging from 320x320x64 to 20x20x2048. The features are compacted through the convolution and Cross Stage Partial (C2F) blocks, followed by Spatial Pyramid Pooling (SPPF) to fuse the features from different scales. The final detection layer predicts bounding boxes, objectness scores and class labels, producing ready-to-use features for the training fabric defect detection models such as holes, lines, and stains.

Input Size: (640, 640, 3) (RGB Image)

Backbone: ResNet-50 (Pretrained)

- **Layer 1:** Convolution + BatchNorm + ReLU + MaxPooling (Input: 640x640x3, Output: 320x320x64)
- **Layer 2:** ResNet Block 1 (Convolution Layer) — (Output: 160x160x256)
- **Layer 3:** ResNet Block 2 (Convolution Layer) — (Output: 80x80x512)
- **Layer 4:** ResNet Block 3 (Convolution Layer) — (Output: 40x40x1024)
- **Layer 5:** ResNet Block 4 (Convolution Layer) — (Output: 20x20x2048)
- **Layer 6:** Convolution Block (Downsample: 2048 -> 1024)
- **Layer 7:** C2f Block (Output: 1024)
- **Layer 8:** Convolution Block (Downsample: 1024 -> 512)
- **Layer 9:** C2f Block (Output: 512)
- **Layer 10:** SPPF Block (Spatial Pyramid Pooling) (Output: 1024)
- **Layer 11:** Detection Layer — Predicting bounding boxes, objectness scores, and class labels.

Figure 12: Feature extraction architecture

5.3 Results of hyperparameter determination

The hyperparameters applied use the epoch trial values of 50, 100, 150 and early stopping so that it can be used to choose a good combination between epoch and learning rate 0.001429 and iou 0.3. The hyperparameter results in Table 6 show the model performance which can be seen in the precision and recall values.

Table 6: Hyperparameter experiment result

Epoch	Learning Rate	Precision	Recall	mAP 50	mAP 50-95
50	0.001429	0.701	0.632	0.671	0.44
100	0.001429	0.813	0.577	0.686	0.439
150	0.001429	0.702	0.63	0.664	0.413
Early stop at (67)	0.001429	0.720	0.574	0.663	0.413

The hyperparameter experiment results presented in Table 6 indicate a non-monotonic relationship between the number of training epochs and key evaluation metrics, such as precision, recall, and mAP. While the precision significantly increased from 0.701 at 50 epochs to 0.813 at 100 epochs, this gain was accompanied by a notable decline in recall (0.577) and minimal improvements in mAP50 (0.686) and mAP50-95 (0.439). A further increase to 150 epochs led to a regression in precision and map scores, suggesting the onset of overfitting, particularly when training is prolonged without adequate regularization mechanisms.

More critically, the downward trend in recall and the inconsistent mAP50-95 values can be attributed to the underrepresentation of the 'hole' defect class in the training data. This class imbalance hinders the model's ability to generalize well across all defect types, especially for small-scale or visually subtle defects such as holes, which are often harder to localize due to their limited spatial footprint and low texture contrast. The consistently low recall scores, even with increased epochs, reflect the model's bias toward the majority classes (e.g., lines and stains), which dominate the training set. This asymmetry in class representation has a compounding effect: while the model becomes more precise (fewer false positives), it fails to detect minority class instances, leading to false negatives that are critical in high-stakes quality assessment applications. In the context of fabric inspection, misclassification or omission of small yet significant defects like holes could result in unacceptable quality decisions downstream.

5.4 Machine learning results of detection and classification of hole, line and stain defects

The results of the process of calling the model and detecting defects in the images are the steps in the CNN model for detecting and classifying hole, line and stain defects can be seen in Figure 13.

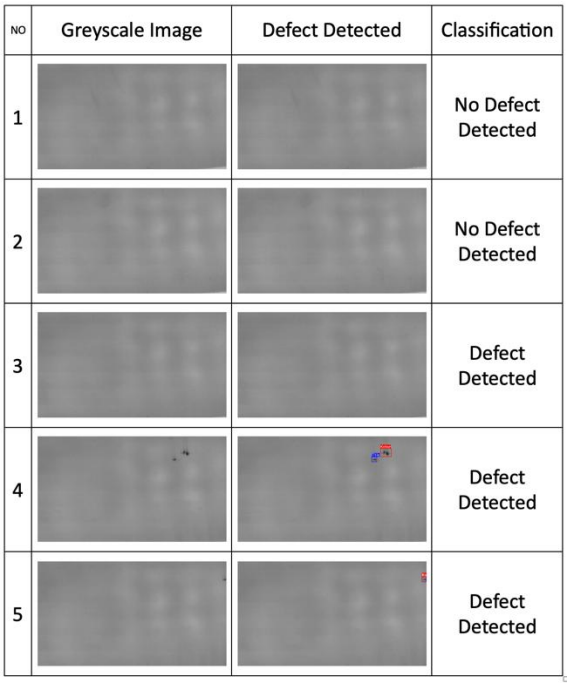


Figure 13: Detection model results

Model testing is done to test the model in recognizing defective objects and recognizing the types of fabric defects. The defects are in the form of holes, lines and stain.

Table 7: Model testing result on proposed model

Class	Precision	Recall	mAP50	mAP 50-95	F1-Score
All	0.789	0.554	0.675	0.461	0.61
Line	0.781	0.639	0.744	0.536	0.66
Stain	0.829	0.720	0.839	0.647	0.62
Hole	0.758	0.302	0.442	0.201	0.52

Table 7 presents the model testing results for three defect types: line, stain, and hole. Overall, the model achieved a precision of 0.789 and recall of 0.554, with a mAP50 of 0.675 and mAP50-95 of 0.461. The highest performance was observed in the *stain* class, with a precision of 0.829 and mAP50 reaching 0.839, indicating strong detection capability for stain defects. In contrast, the lowest performance was seen in the *hole* class, where recall dropped to 0.302 and mAP50-95 was only 0.201, suggesting that the model struggles to detect hole-type defects effectively.

Table 8: Model testing result on YOLOv8 Basic

Class	Precision	Recall	mAP50	mAP 50-95	F1-Score
All	0.652	0.421	0.531	0.336	0.51
Line	0.643	0.498	0.591	0.412	0.56
Stain	0.699	0.578	0.682	0.489	0.63
Hole	0.621	0.218	0.357	0.161	0.32

Based Table 7 and 8 compared to the YOLOv8 Basic model with the Proposed Model shows notable improvements across all evaluation metrics. Overall precision increased from 0.652 to 0.789, and recall improved from 0.421 to 0.554. The mAP50 and mAP50-95 also improved significantly, from 0.531 and 0.336 in YOLOv8 Basic to 0.675 and 0.461 in the Proposed Model, respectively.

The hole class, while still the most challenging, saw modest gains in mAP50 (from 0.357 to 0.442) and mAP50-95 (from 0.161 to 0.201). However, its recall remained low in both models, indicating that further enhancement is needed to detect hole-type defects effectively.

Although the Proposed Model demonstrates overall performance improvement compared to the YOLOv8 Basic model, the results for the hole class remain relatively low, particularly in terms of recall and mAP50-95. This indicates that the model struggles to consistently detect most hole-type defects. The low recall is likely due to the limited number of training samples and the high variability in the shape, size, and position of the defects, which constrain the model's ability to generalize effectively. This issue persists despite the application of data augmentation techniques intended to balance the distribution across defect types. Therefore, further strategies such as improving annotation quality, increasing real-world data for minority classes, or applying techniques like focal loss may be necessary to enhance the model's sensitivity to underrepresented defect categories.

Based on mAP@0.5 performance, the proposed model achieves a score of 0.675, representing a significant improvement over DA-YOLOv8 (0.446) and confirming the effectiveness of incorporating ResNet as the backbone in enhancing feature representation. Although it does not outperform FF-YOLO (0.755) or YOLOv5 + ResNet (0.951), the proposed model offers a strong balance between detection accuracy and computational efficiency. The YOLOv5 + ResNet architecture, despite its high accuracy, relies on a two-stage detection process that may introduce latency and increased complexity, making it less suitable for real-time applications. In contrast, the proposed model retains a single-stage detection framework while leveraging the enriched semantic features provided by the ResNet backbone, enabling more accurate object localization and classification without sacrificing inference speed. These results indicate that the proposed model is a promising solution for industrial settings that require both high precision and real-time performance.

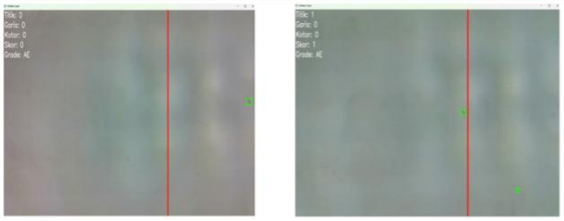


Figure 14: Objects before (left) and after (right) crossing the threshold

The hole, line and stain defect calculations are carried out after the detected object passes the bias line, so that they are calculated according to the defect classification of the object. Figure 14 (a) the object is detected before passing the bias line and all the assessment elements still have a value of 0. Figure 14 (b) shows that the detected object has passed the bias line, so that the assessment elements have changed according to the classification of the detected object.

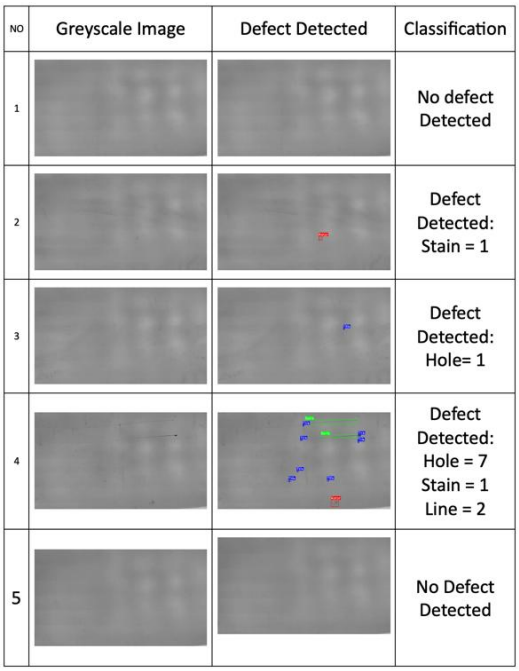


Figure 15: Computation of defect detection based on classification

5.5 Calculation results of maximum bounding box length

The bounding box result is a box used to mark the detected objects. Figure 16 shows the object detection with a bounding box, the object can be detected by issuing the maximum bounding box length value.

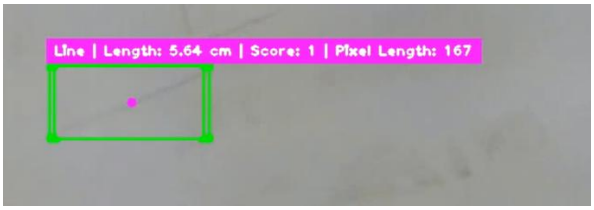


Figure 16: Bounding box display showing maximum length

The calculation of pixels to cm is as follows: There is a Bounding box with a pixel length of 167 Count Pixels to cm.

The bounding box result is a box used to mark the detected objects. Figure 16 shows the already object detection with a bounding box. The object can be detected by issuing the maximum length value of the bounding box. The calculation of pixels to cm is as follows:

1. There is a bounding box with a pixel length of 167.
2. Calculate Pixels to cm
3. Defect Length Result
4. The score obtained is 1 because it only meets the score limit of 1 to 8 cm.

5.7 Results of calculating defect score with 4-point system

The calculation of the defect scores is based on the number of defects in each class with the dimensions of the defects according to the categories listed in Table 2. Every time a defect passes the threshold, it is locked and the defect score is calculated based on the bounding box dimensions. The process can be seen in Figure 17.

5.8 Results of fabric quality determination

The fabric quality ranking (grade) based on a 4-point system has been successfully carried out by a system device integrated with a learning model for detecting hole, line and stain defects.

Figure 17: Defect score calculation view

Figure 18: Results of defect detection interface and fabric quality grading

The fabric that runs on the automation tool and is captured by the camera into a frame is detected, classified and calculated for each defect found according to the type of defect. The rules used to determine the quality of the fabric use a 4-point system. The Grade result based on Figure 18 is BE.

5.9 Accuracy test results of learning machine architecture for hole, line and stain defects

The performance results of the proposed model, as summarized in Table 7 and further illustrated in the confusion matrix in Figure 19, provide critical insight into its classification capabilities across the three fabric defect categories: Line, Stain, and Hole. The aggregated performance metrics indicate that the model achieved a precision of 0.789, recall of 0.554, mAP50 of 0.675, mAP50-95 of 0.461, and an F1-score of 0.61. These values offer a coherent and consistent summary, resolving previously reported discrepancies in overall performance statistics. Class-wise, the model exhibited high confidence in detecting Line defects, as evidenced by 24 true positives and a competitive F1-score of 0.66. For the Stain class, the model achieved the highest recall (0.720) and mAP50 (0.839), albeit with notable misclassification into the Line class. The Hole class, however, posed the greatest challenge, with the lowest recall (0.302) and F1-score (0.52), suggesting difficulties in distinguishing its features due to potential class imbalance or inadequate discriminatory patterns. These findings highlight the need for data balancing and enhanced feature engineering, particularly for underrepresented classes, to improve the model's robustness and ensure consistent performance across all defect categories.

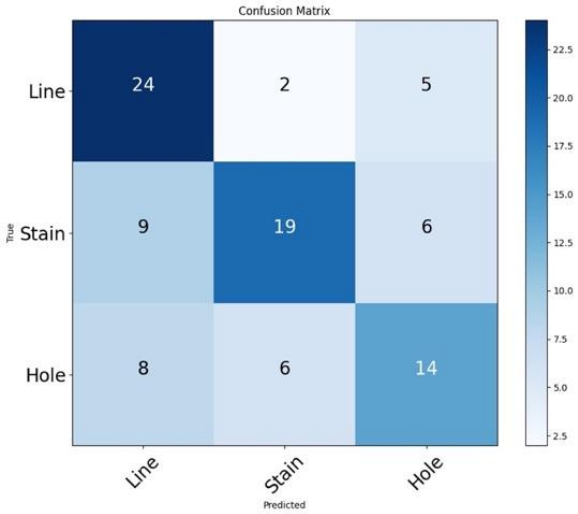


Figure 19: Results of confusion matrix

The analysis based on Figure 19 has good a model performance for the "Line" class with a few classification errors, while the performance in the "Stain" and "Holes" classes needs to be improved, especially to reduce errors between classes that have similar features.

6 Conclusion

This study successfully developed an automated system integrating the Intel RealSense D435i camera and a Convolutional Neural Network (CNN) model to detect and classify three types of fabric defects hole, line, and stain while simultaneously evaluating fabric quality based on the 4-point grading system. The hardware system, built with dimensions of 100 cm × 30 cm × 25 cm, enables real-

time defect detection. The implemented CNN model demonstrated competitive performance, achieving a precision of 0.789, recall of 0.554, mAP50 of 0.675, mAP50-95 of 0.461, and an F1-score of 0.61. These results indicate a reasonable balance between classification accuracy and generalization capability on unseen data.

The primary advantage of this research lies in the seamless integration of hardware and software components, enabling the system to operate fully autonomously without human intervention. Notably, the implementation of the 4-point grading system represents a significant contribution, as it allows for the automated calculation of defect scores and determination of fabric grade without the need to stop the fabric rolling machine. This marks a substantial improvement over previous approaches that required manual inspection and temporary halting of the production process to perform grading. As a result, the proposed approach not only enhances inspection speed and classification accuracy but also significantly reduces production downtime, improves operational efficiency, and supports the automation of quality control in textile manufacturing.

Despite its promising results, the system presents several limitations. The training dataset is relatively limited in size and diversity, which may affect the model's ability to generalize to more complex or varied defect patterns. Additionally, the system has not yet been tested under real industrial conditions, leaving uncertainties regarding its stability and reliability in actual production environments.

Future research should focus on expanding the dataset with more diverse and representative defect types, employing data augmentation techniques to enrich model robustness, and conducting field trials in industrial settings to validate the system's performance under real-world operational constraints.

In summary, the automated system developed in this study not only demonstrates strong potential in improving the accuracy and speed of fabric quality inspection but also provides a scalable and practical solution for advancing quality control processes within the modern textile industry.

7 Acknowledgment

The authors would like to express their deepest gratitude to the Rector and Vice Rector II of Gunadarma University for their support has been given, especially in financing the publication of this research. The support and contribution given are very meaningful in realizing this work. Their sincere gratitude also goes to INDORAMA SYNTHETICS LTD. who has provided samples of fabric defects.

References

[1] M. Ateeq, A. P. P. Abdul Majeed, H. Hafizh, M. A. Mohd Razman, I. Mohd Khairuddin, and N. H. Noordin, "A Feature-Based Transfer Learning Method for Surface Defect Detection in Smart Manufacturing," in *Intelligent Manufacturing and Mechatronics*, W. H. Mohd. Isa, I. Mohd. Khairuddin, Mohd. A. Mohd. Razman, S. 'Atifah Saruchi, S.-H. Teh, and P. Liu, Eds., Singapore: Springer Nature Singapore, 2024, pp. 455–461.

[2] K. Gopalakrishnan and P. T. Vanathi, "Fabric Defect Detection Using Deep Learning Techniques," in *Ubiquitous Intelligent Systems*, P. Karuppusamy, F. P. García Márquez, and T. N. Nguyen, Eds., Singapore: Springer Nature Singapore, 2022, pp. 101–113.

[3] Q. Dong, "Surface defect detection algorithm for aluminum profiles based on deep learning," *Informatica*, vol. 48, no. 13, pp. 1–14, 2024.

[4] F. K. Konstantinidis, I. Kansizoglou, K. A. Tsintotas, S. G. Mouroutsos, and A. Gasteratos, "The role of machine vision in industry 4.0: A textile manufacturing perspective," in *IST 2021 - IEEE International Conference on Imaging Systems and Techniques, Proceedings*, 2021. doi: 10.1109/IST50367.2021.9651459.

[5] M. P. Sikka, A. Sarkar, and S. Garg, "Artificial intelligence (AI) in textile industry operational modernization," 2024. doi: 10.1108/RJTA-04-2021-0046.

[6] Y. A. Karayiannis *et al.*, "Defect detection and classification on web textile fabric using multiresolution decomposition and neural networks," in *Proceedings of the IEEE International Conference on Electronics, Circuits, and Systems*, 1999. doi: 10.1109/ICECS.1999.813221.

[7] E. Shady, Y. Gowayed, M. Abouiiana, S. Youssef, and C. Pastore, "Detection and Classification of Defects in Knitted Fabric Structures," *Textile Research Journal*, vol. 76, no. 4, 2006, doi: 10.1177/0040517506053906.

[8] C. H. Chan, "Fabric defect detection by Fourier analysis," *IEEE Trans Ind Appl*, vol. 36, no. 5, 2000, doi: 10.1109/28.871274.

[9] S. Das, M. K. Selvaraj, S. Shanmugavelu, and S. Jayaram, "Deep learning neural networks for knitted fabric defect identification and classification," *Man-Made Textiles in India*, vol. 50, no. 12, 2022.

[10] L. Shi, J. Song, Y. Gao, G. Cheng, and B. Hao, "YOLO-GFD: A Fast and Accurate Fabric Defect Detection Model," in *2023 4th International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering, ICBAIE 2023*, 2023. doi: 10.1109/ICBAIE59714.2023.10281292.

[11] M. Li, D. Pi, and S. Qin, "An efficient single shot detector with weight-based feature fusion for small object detection," *Sci Rep*, vol. 13, no. 1, p. 9883, 2023, doi: 10.1038/s41598-023-36972-x.

[12] S. N. Appe, G. Arulselvi, and G. N. Balaji, "CAM-YOLO: tomato detection and classification based on improved YOLOv5 using combining attention mechanism," *PeerJ Comput Sci*, vol. 9, 2023, doi: 10.7717/peerj-cs.1463.

[13] Y. Liu, P. Sun, N. Wergeles, and Y. Shang, "A survey and performance evaluation of deep learning methods for small object detection," *Expert Syst Appl*, vol. 172, 2021, doi: 10.1016/j.eswa.2021.114602.

[14] N. D. Nguyen, T. Do, T. D. Ngo, and D. D. Le, "An Evaluation of Deep Learning Methods for Small Object Detection," *Journal of Electrical and Computer Engineering*, 2020, doi: 10.1155/2020/3189691.

[15] B. Liu *et al.*, "PRC-Light YOLO: An Efficient Lightweight Model for Fabric Defect Detection," *Applied Sciences (Switzerland)*, vol. 14, no. 2, 2024, doi: 10.3390/app14020938.

[16] X. Jun, J. Wang, J. Zhou, S. Meng, R. Pan, and W. Gao, "Fabric defect detection based on a deep convolutional neural network using a two-stage strategy," *Textile Research Journal*, vol. 91, no. 1–2, 2021, doi: 10.1177/0040517520935984.

[17] P. Bory, "Deep new: The shifting narratives of artificial intelligence from Deep Blue to AlphaGo," *Convergence*, vol. 25, no. 4, 2019, doi: 10.1177/1354856519829679.

[18] A. Zahra, M. Amin, F. E. A. El-Samie, and M. Emam, "Efficient utilization of deep learning for the detection of fabric defects," *Neural Comput Appl*, vol. 36, no. 11, pp. 6037–6050, 2024, doi: 10.1007/s00521-023-09137-0.

[19] G. Revathy and R. Kalaivani, "Fabric defect detection and classification via deep learning-based improved Mask RCNN," *Signal Image Video Process*, vol. 18, no. 3, pp. 2183–2193, 2024, doi: 10.1007/s11760-023-02884-6.

[20] A. International, "Standard Test Methods for Visually Inspecting and Grading Fabrics." [Online]. Available: https://www.astm.org/d5430-93r00.html

[21] J. Jing, D. Zhuo, H. Zhang, Y. Liang, and M. Zheng, "Fabric defect detection using the improved YOLOv3 model," *J Eng Fiber Fabr*, vol. 15, 2020, doi: 10.1177/1558925020908268.

[22] X. Yue, Q. Wang, L. He, Y. Li, and D. Tang, "Research on Tiny Target Detection Technology of Fabric Defects Based on Improved YOLO," *Applied Sciences (Switzerland)*, vol. 12, no. 13, 2022, doi: 10.3390/app12136823.

[23] X. Luo, Q. Ni, R. Tao, and Y. Shi, "A Lightweight Detector Based on Attention Mechanism for Fabric Defect Detection," *IEEE Access*, vol. 11, 2023, doi: 10.1109/ACCESS.2023.3264262.

[24] Q. Liu, C. Wang, Y. Li, M. Gao, and J. Li, "A Fabric Defect Detection Method Based on Deep Learning," *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2021.3140118.

[25] H. Jung and J. Rhee, "Application of YOLO and ResNet in Heat Staking Process Inspection," *Sustainability (Switzerland)*, vol. 14, no. 23, 2022, doi: 10.3390/su142315892.

[26] C. Chen, "FF-YOLO: Fashion Fabric Detection Algorithm Based on YOLOv8," *IEEE Access*, vol. 13, pp. 2298–2312, 2025, doi: 10.1109/ACCESS.2024.3524618.

[27] W. Song, D. Lang, J. Zhang, M. Zheng, and X. Li, "Textile Defect Detection Algorithm Based on the Improved YOLOv8," *IEEE Access*, vol. 13, pp. 11217–11231, 2025, doi: 10.1109/ACCESS.2025.3528771.

[28] A. Das Gupta *et al.*, "An approach to automatic fault detection in four-point system for knitted fabric with our benchmark dataset Isl-Knit," vol. 10, Mar. 2024, doi: 10.1016/j.heliyon.2024.e35931.

[29] Y. Matsuzaka and R. Yashiro, "AI-Based Computer Vision Techniques and Expert Systems," 2023. doi: 10.3390/ai4010013.

[30] N. T. Allo, Z. Zainuddin, and others, "A Novel Approach of Hybrid Bounding Box Regression Mechanism to Improve Convergency Rate and Accuracy.," *International Journal of Intelligent Engineering & Systems*, vol. 17, no. 2, 2024.

[31] W. Xiong, Y. Wu, J. Cao, and R. Li, "IB-CBB: an improved spatial index considering intersection based on clipped bounding boxes," *Ann GIS*, vol. 30, no. 2, pp. 233–250, 2024.

[32] M. Sobirov, N. Sharibaev, A. Kayumov, and K. Musayev, "Method of assessment of structural properties of knitted fabrics based on image analysis," in *E3S Web of Conferences*, 2024, p. 3020.

[33] Q. and Z. J. and Y. S. Zhang, "Enhancing YOLOv8 Object Detection with Shape-IoU Loss and Local Convolution for Small Target Recognition," *Informatica*, vol. 49, no. 21, pp. 105–120, 2025.

[34] M. L. Swartz, "Managing digital images," *American Journal of Orthodontics and Dentofacial Orthopedics*, vol. 118, no. 3, pp. 354–358, Sep. 2000, doi: 10.1067/mod.2000.110525.

[35] B. Koonce, "ResNet 50," in *Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization*, Berkeley, CA: Apress, 2021, pp. 63–72. doi: 10.1007/978-1-4842-6168-2_6.

Hybrid Optimized Machine Learning Models for Forecasting Electricity Demand and Generation: A Comparative Analysis

Liandian Jiang[1], Pengfei Huang[1], Junyang Tian[1], Haiyong Li[1], Bin Liu[1], Shan Liang[1], Jian Wang[2,*]
[1]Guangxi Power Grid Co., Ltd, Nanning 530023 China
[2]Shandong University, Jinan 250100, China
E-mail: wjianvip@126.com
[*]Corresponding author

Keywords electricity demand forecasting, maximum electricity generation, support vector regression, catboost model, metaheuristic optimization algorithm.

Increasingly complicated and volatile electricity demand considerably complicates the job of operators and planners of energy systems. For enhancing grid reliability and efficiency in resource allocation, realistic forecasting is thus in high demand regarding overall electricity demand and peak generation capacity. The study will introduce a new approach by adopting ML techniques, namely CatBoost and support vector regression, for predicting electricity demand and maximum generation capacity. Optimization algorithms for metaheuristics, such as phasor particle swarm optimizer and chaos game optimizer, are used to improve model performance. To this end, four hybrid models were developed: namely, CatBoost-CGO, CatBoost-PPSO, SVR-CGO, and SVR-PPSO. The performance was evaluated using statistical indices such as MAE, RMSE, MAPE, and R^2. Among the models, CatBoost-PPSO demonstrated the highest accuracy with $R^2 = 0.955$ for total demand and $R^2 = 0.958$ for maximum generation in the test set, outperforming other combinations. These results validate the effectiveness of hybrid optimization in improving prediction accuracy.

Povzetek:

1 Introduction

Electricity demand forecasting helps in keeping the operation and planning of power systems effective. Correct forecasts of overall electricity demand and peak generation of electricity are required in order for the utility to be aware, in terms of decisions on resource planning, grid management, and infrastructure plans[1]. Due to increasing complexity and dynamism among energy markets, strong and reliable forecasting models are increasingly in demand, which, upon adaptation to changed conditions, will deliver proper predictions effectively[2].

In the last couple of years, ML methodologies have emerged as proficient tools in forecasting electricity demand, being able to uncover intrinsic patterns and interrelations present in the historical data. Of these, SVR and CatBoost have grown as two well-recognized approaches applied to problems related to extreme non-linearity and high-dimensionality. The performance of these models can be further improved using optimization techniques that are suitably tuned to improve their predictive accuracies. For example, PPSO and CGO represent the optimization methods giving the paths for improving parameters in ML models, thus enhancement of predictive accuracy and efficiency. These comparative insights have very important implications for energy providers, policy analysts, and researchers alike. Identifying the best combination of algorithms and optimization techniques greatly enhances the reliability of forecasting in terms of electricity demand, optimizes resource utilization, and contributes toward a much more robust and sustainable energy infrastructure.

1.1 Related studies

The importance of forecasting electricity and energy consumption has attracted several researchers to forecasting using various methods and models. A wide range of ML-based models has been applied in different forms with supervised learning algorithms, including decision trees, SVM, linear, and logistic regression, along with neural networks.

[3] Proposed an NN-based grey forecasting model, namely the NNGM, to project electricity consumption patterns. The present study investigates the prediction performance of the NNGM for electricity consumption using datasets from the Turkish governmental database. From the empirical results, one could perceive that the NNGM performed remarkably well in the forecasting of electricity consumption. [4] This section presents the application of ANN to forecast long-term energy usage in Greece, using the MLP model with different trials of

architecture. Real input data and real output data are used to influence the energy consumption for training, validating, and testing purposes. The ANN model predicts energy consumption for the years ranging from 2005 to 2015. Comparison analysis underlines the accuracy of the ANN method, as compared to other methods like linear regression and SVM. The study pinpoints that correct forecasting of energy consumption is imperative for the development of an appropriate energy policy and to satisfy socio-economic factors, underlining the relevance of the method for long-term energy consumption prediction in Greece.[5] It proposes an improved ANN model for the hourly forecasting of building electricity consumption. The parameters of the ANN structure have been optimized by using an improved version of the Particle Swarm Optimization algorithm, iPSO. The important input variables are selected and the model is simplified with the help of PCA. To assess the performance of this model, 2 datasets have been used: one from the Energy Prediction Shootout contest and the other from a campus building in East China. Comparative analysis with conventional ANN and hybrid Genetic Algorithm-ANN (GA-ANN) models shows both versions of iPSO-ANN and GA-ANN with higher accuracy. Besides, it is observed that iPSO-ANN performs with a higher modeling time compared to the GA-ANN. Hopefully, the proposed model will be able to demonstrate a very feasible approach toward real-time prediction of building electricity consumption.

In a study done by [6] a novel approach to predict natural gas consumption is introduced, employing BMA to effectively handle uncertainties related to model structure and parameters, consequently improving prediction accuracy. 6 variables, such as GDP, urban population, energy consumption structure, industrial composition, energy efficiency, and exports of goods and services, are selected for prediction purposes. Comparative evaluation against the gray prediction model, Linear regression model, and ANNs demonstrates the versatility and reliability of BMA in forecasting the expected substantial increase in natural gas consumption. Studies of [7] evaluate various regression analyses using different time intervals of hourly and daily data from a research house. While multiple linear regression models incorporating outdoor temperature and solar radiation improve the R2, they also worsen the RMSE. The paper aims to serve as a comprehensive resource for future research in energy forecasting, suggesting a move towards personalized models for households based on smart meter data and user-friendly software tools. [8] explores 3 unique approaches of the ARIMA-ANFIS model to predict annual energy consumption in Iran. Initially, the ARIMA model is applied to 4 input features, and its nonlinear residuals are predicted using 6 diverse ANFIS structures. The second approach combines ARIMA predictions with the 4 input features for ANFIS prediction. To tackle data scarcity, the third approach integrates AdaBoost data diversification. Results indicate substantial enhancements in prediction accuracy across all hybrid patterns, particularly with the third approach, leveraging the diversification model. [9] presents a method for predicting electricity consumption in 28 commercial buildings in Seo-gu, Gwangju, South Korea, using publicly available monthly and daily consumption data. It addresses the challenge of accurately selecting parameters that influence energy usage, proposing a 2-step approach. Firstly, a customized electricity consumption model is developed based on individual building characteristics within the building community. Secondly, additional data such as building features and energy equipment types are integrated using dynamic-time-warping-based clustering classification. In comparison with conventional time-series analysis approaches, the suggested strategy performs better as it uses ML techniques for both model building and optimization. This development has the potential to improve the accuracy of electricity consumption estimates for commercial buildings and offer insightful information for controlling the supply and demand of power in both urban and building contexts.

[10] introduces a new stacking model aimed at predicting building energy consumption. By blending different base prediction algorithms and generating "meta-features," this model provides a comprehensive analysis of datasets from various angles. The practical application of the stacking model is demonstrated through 2 engineering case studies. Comparative evaluation against established prediction models, such as Random Forest and Gradient Boosted Decision Tree, reveals the superior performance of the stacking method in terms of accuracy, generalization, and robustness. In essence, this model diversifies available prediction techniques and enhances the algorithmic arsenal for empirical models. [11] presents a short-term electricity consumption prediction model using LSTM with an attention mechanism. Initially, the attention mechanism assigns weight coefficients to input sequence data. Following this, the output of each LSTM cell is computed using forward propagation, and errors between predicted and actual values are determined using back-propagation. Model weights are adjusted through gradient descent to minimize errors. Experimental results across various electricity consumption types demonstrate the effectiveness of the proposed model in improving prediction accuracy compared to existing approaches. [12] presents an ML-driven approach for forecasting power usage, exploring a range of techniques including linear regression, K Nearest Neighbors, XGBOOST, random forest, and ANNs. The study leverages historical electricity usage data obtained from a power utility company, which undergoes preprocessing to handle outliers and missing values. Evaluation of model performance involves various assessment measures, revealing the efficacy of the proposed method in accurately predicting power usage. Notably, the K-Nearest Neighbors model emerges as the top performer, demonstrating particularly high accuracy in predicting agricultural production. [13] explores the application of deep learning in forecasting annual energy consumption for buildings and conducts a comparative evaluation of model performance. An innovative aspect of this study is the development of a model trained on data from multiple buildings, allowing building designers to input key features early in the development process to predict annual average energy consumption. Results show that the ANN

method surpasses other models in accurately predicting annual energy consumption.

Several studies have been conducted on electricity demand and energy consumption forecasting using machine learning and statistical approaches. Table 1 presents a comparative summary of selected state-of-the-art methods, highlighting their employed techniques, reported metrics, and identified limitations. Most of these works focused either on specific case studies such as buildings or used traditional machine learning techniques without hybrid optimization. Moreover, several studies lack detailed statistical performance metrics or comparisons across multiple algorithms. This study addresses these limitations by proposing hybrid models combining advanced regressors (CatBoost and SVR) with novel metaheuristic optimizers (PPSO and CGO), demonstrating significant improvements in forecasting accuracy.

Table 1: Summary of Related studies

Study	Forecasting Method	Reported Metrics	Accuracy/Highlights	Identified Limitations
Hu (2017)	Neural Network + Grey Forecasting	MAPE	Good predictive behavior	No optimization; lacks statistical metrics
Ekonomou (2010)	ANN	Error (%)	Low error for long-term prediction	No hybrid methods or optimization
Li et al. (2015)	ANN + PCA	MAE, RMSE	Improved short-term accuracy	Building-level focus only
Fumo & Biswas (2015)	Linear Regression	MAE, RMSE	Simple model; acceptable performance	Lacks adaptability to nonlinear patterns
Barak & Sadegh (2016)	ARIMA–ANFIS hybrid	MSE = 0.026	Low MSE (0.026%)	Model complexity; no R^2 given
Hwang et al. (2020)	ML approach for commercial buildings	RMSE, MAE	Effective for commercial data	Application-limited; no comparative benchmarks
Wang et al. (2020)	Model Stacking	RMSE, Accuracy	9.5%–31.6% improvement over baselines	High model complexity
Lin et al. (2020)	LSTM + Attention	RMSE	Strong performance on time series	Requires large datasets
Vijendar Reddy et al. (2023)	ML Techniques	Accuracy	Broad machine learning application	No benchmarking or R^2 data

1.2 Main novelty

Motivated by the limitations observed in earlier research, such as insufficient adaptability to nonlinear patterns and lack of effective parameter optimization, the proposed approach aims to address these gaps. The investigation focuses on evaluating the comparative performance of the hybrid models and determining which configuration offers the optimal balance between predictive accuracy and computational efficiency. This paper considers 2 metaheuristic optimizers including PPSO and CGO to integrate them with SVR and CatBoost. Thien the developed hybrid models are compared based on several metrics, in order to present an efficient ML-based algorithm to forecast the electricity consumption by computing maximum generation and total demand. The aim of this study is for the proposed model to be prominent and to overcome existing models. It is hypothesized that the integration of CatBoost with PPSO will result in superior forecasting performance due to the model's advanced feature handling capabilities and the optimizer's enhanced global search behavior.

The rest of the paper is organized as follows. Section 2, contains a brief description of the compared models such as SVR and CatBoost. Also, the utilized metaheuristic optimizers are described. then the presented hybrid models are introduced. Section 3, presents the results of the numerical analysis of the hybrid models. Section 4 includes the results of the model's comparison, based on the statistical evaluation indices. Finally, section 5 concludes the comparison and discusses the selected hybrid model for predicting electricity consumption.

2 Algorithm's description & method of study

The descriptions of the ML models utilized in this comparison including SVR and CatBoost, are provided concisely. Also, the selected optimizers are introduced along with the hybrid models developed through integrating optimizers and ML models.

2.1 CatBoost algorithm

CatBoost (short for Category Boosting) is a powerful ML algorithm developed by Yandex [14], specifically designed to tackle complex predictive modeling tasks with efficiency and accuracy. It fundamentally employs, at a base level, the gradient boosting technique, which is one of the most popular ensembles learning methods that fits a sequence of decision trees in improving a previous model's prediction. What actually makes CatBoost

different is its unique capability to handle categorical features out-of-the-box: unlike classical methods, which require explicit preprocessing steps like one-hot encoding, CatBoost handles categoricals internally during the training phase, saving time and quite often, at the same time, improving model performance. It also contains many optimization strategies to enhance learning: ordered boosting, which trains instances in order of gradient; oblivious trees, special structure of trees that simplify computations. Besides that, prevention of overfitting is implemented in CatBoost by means of regularization methods, such as L2 regularization and tree depth regularization. So, CatBoost balances out model complexity and generalization; therefore, robust models are obtained that have the capability to give accurate predictions for data that has not been seen before. Mathematically, CatBoost optimizes an objective function, usually based on MSE for regression or cross-entropy for classification, via gradient boosting. The model parameters are optimized in order to minimize an objective function L (y, F(x)) with respect to the goal variable y and the input features x. Optimization of the objective function is performed in iterations through a gradual addition of decision trees in the ensemble, with each added tree aiming at a minimal residual of the previous ensemble predictions [14], [15]. In the context of this study, since the forecasting task is regression-based, the specific loss function employed is the Mean Squared Error (MSE), formally defined as:

$$\boldsymbol{L_{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i)^2} \quad \textbf{(1)}$$

where y_i represents the true target value, y_i is the model's predicted value, and n denotes the number of samples.

2.2 Support vector regression algorithm

SVR is one of the supervised learning algorithms for regression, introduced by [16]. It belongs to a family of SVMs, which is mainly applied to classification. However, the extension of this to the domain of regression where one tries to predict continuous outcomes, rather than discrete classes, is known as SVR.

The basic idea behind the work of SVR is to find that function which does the job of approximating mappings from input variables to continuous output variable with minimal error. To show how the relationship of input and output variables is modeled, a hyperplane or set of hyperplanes is constructed in a high-dimensional space. The hyperplane(s) is/are chosen such that it/they possess maximum margin from the data points. The margin of the hyperplane is defined as the distance between the hyperplane and the nearest data point(s).

Unlike other traditional regression methods, the objective of SVR is to capture the points that are within a boundary of a specified margin from the hyperplane(s)in other words, support vectorsrather than precisely fitting all data. Because of this, SVR is more resistant against noise and outliers. The mathematical description of the SVR can be visualized with:

Given a training dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ where x_i represents the input features and y_i represents the corresponding output values, SVR aims to find a function $f(x)$ that approximates the relationship between the input features and the output values. The primal optimization problem for SVR can be formulated as follows:

$$\textbf{MINIMIZE: } \boldsymbol{\frac{1}{2}\parallel w \parallel^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*)} \quad \textbf{(2)}$$

Subject to:

$$\begin{aligned} &\boldsymbol{y_i - \mathbf{w}^T\phi(\mathbf{x}_i) - b \le \varepsilon + \xi_i} \\ &\boldsymbol{\mathbf{w}^T\phi(\mathbf{x}_i) + b - y_i \le \varepsilon + \xi_i^*} \\ &\boldsymbol{\xi_i, \xi_i^* \ge 0, i = 1, 2, \ldots, N} \end{aligned} \quad \textbf{(3)}$$

where $\boldsymbol{w}$ is the weight vector. $\phi(x)$ is a mapping function to transform the input features x into a higher-dimensional space. b denotes the bias term. ε shows the epsilon-insensitive loss function parameter, which defines the size of the margin around the predicted values where no penalty is incurred.

ξ_i and ξ_i^* are slack variables, allowing for deviations from the margin and controlling the trade-off between the margin and the training error. C is a regularization parameter that balances the trade-off between maximizing the margin and minimizing the training error. A smaller value of C encourages a larger margin but may allow more training errors, while a larger value of C penalizes errors more heavily, potentially leading to a narrower margin.

The dual optimization problem, derived from the primal problem using Lagrange duality, involves finding the Lagrange multipliers α_i and α_i^* for each training sample using the equation below:

$$\boldsymbol{f(x) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\langle\phi(x_i)\cdot\phi(x)\rangle + b} \quad \textbf{(4)}$$

where $0 \le \alpha_i, \alpha_i^* \le C$.

The final regression function is given by:

$$\boldsymbol{f(x) = W^T\phi(x) + b} \quad \textbf{(5)}$$

SVR often uses kernel tricks, where the inner product $\langle\phi(x_i), \phi(x_j)\rangle$ is replaced with a kernel function $K(x_i, x_j)$ to implicitly map the input features into a higher-dimensional space without explicitly calculating $\phi(x_i)$ and $\phi(x_j)$.

By using this formulation, SVR is able to minimize training error and maximize margin while identifying a hyperplane or group of hyperplanes that best match the connection between the input features and the output values [16], [17].

2.3 Chaos game optimization algorithm

The Chaos Game Optimization is a metaheuristic optimization technique inspired by the dynamic's principles of the "chaos game" concept originating from fractal theory. Talataheri & Azizi [18], in the year 2020, proposed the CGO and developed the algorithm to emulate the dynamics of chaos games in finding the optimum solutions for optimization problems. Chaos theory focuses on regular patterns that may come out from systems' behavior, like fractals, cycles that repeat themselves, and subsystems that repeat. The systems are shown by these patterns as dynamic, self-similar, and self-organized entities. Chaos theory explains how even small

changes to a dynamic system's starting circumstances can result in radically different behavior later on. This dependency on initial conditions underscores the system's sensitivity and its potential for diverse and unpredictable trajectories.

The CGO is a method for generating geometric patterns using randomness, famously demonstrated by the Sierpinski Triangle. Each vertex of the triangle corresponds to a color on a dice, determining the starting point. Rolling the dice guides subsequent movements towards vertices, marking points halfway. Despite randomness, the pattern converges into a self-similar fractal resembling the Sierpinski Triangle, showing order emerging from randomness. Fig. 1 illustrates this process, resulting in the Sierpinski triangle's creation. The self-similarity of the triangle is evident in the final shape.

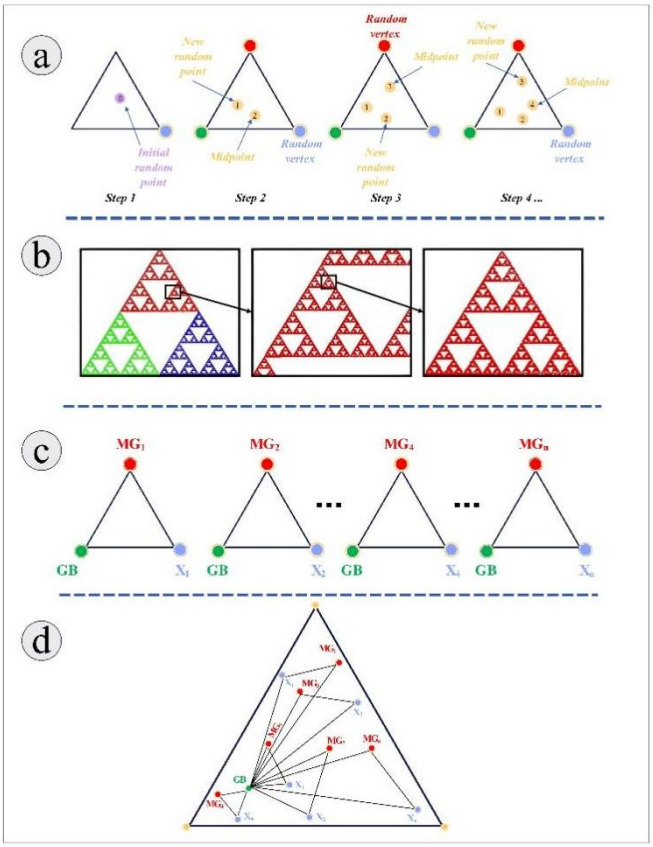


Figure 1: The chaos game process a) generating the Sierpinski triangle b) The self-similarity of the Sierpinski triangle c) the formation of temporary triangles d) multiple triangles in the search area

According to the mathematical representation of the CGO algorithm in Fig. 1, the CGO delves into a variety of potential solution (X), each representing potential seeds within the Sierpinski triangle. Each parameter x_{ij} in Eq. (5) is a decision variable inside each potential solution X_i. Also, the Sierpinsky triangle is the search space of the CGO algorithm and each potential solution inside the search space is called a qualified seed.

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{X_1} \\ \boldsymbol{X_2} \\ \vdots \\ \boldsymbol{X_i} \\ \vdots \\ \boldsymbol{X_n} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x_1^1} & \boldsymbol{x_1^2} & \cdots & \boldsymbol{x_1^j} & \cdots & \boldsymbol{x_1^d} \\ \boldsymbol{x_2^1} & \boldsymbol{x_2^2} & \cdots & \boldsymbol{x_2^j} & \cdots & \boldsymbol{x_2^d} \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x_i^1} & \boldsymbol{x_i^2} & \cdots & \boldsymbol{x_i^j} & \cdots & \boldsymbol{x_i^d} \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x_n^1} & \boldsymbol{x_n^2} & \cdots & \boldsymbol{x_n^j} & \cdots & \boldsymbol{x_n^d} \end{bmatrix} \quad \textbf{(6)}$$

where $\begin{cases} i = 1, 2, \ldots, n \\ j = 1, 2, \ldots, d \end{cases}$.

Also, n represents the total number of qualified seeds or potential solutions. d shows the dimension of each seed.

Each qualified seed has an incipient position where it is selected randomly using the equation below:

$$\boldsymbol{x_i^j(0) = x_{i,\,\mathrm{MIN}}^j + \mathbf{r} \cdot \left(x_{i,\,\mathrm{MAX}}^j - x_{i,\,\mathrm{MIN}}^j\right)} \quad \textbf{(7)}$$

where the expression $x_i^j(0)$ denotes the incipient position of each qualified seed. The minimum and maximum allowed values are $x_{i,\,\min}^j$ and $x_{i,\,\max}^j$, respectively, for the j-th decision variable and i-th potential solution. r is a number that is selected randomly in the range of 0 to 1.

Chaos theory investigates the innate patterns seen in dynamic systems, describing them as self-similar and self-organizing systems. When modeling these patterns as potential solutions (X) in optimization problems, initial seeds are essential. Seed eligibility is determined by fitness values, with the highest indicating optimal candidates. The goal of the mathematical model is to produce suitable seeds in the search space so that a Sierpinski triangle is formed. Temporary triangles are created for each eligible seed, using Global Best (GB),

Mean Group (MG_i), and chosen seed (Xi) as vertices. New eligible seeds are introduced via CGO, with movements guided by dice rolls and probabilities towards GB or MG_i. Constraints on seed movement are imposed through randomly generated factorials to adhere to CGO's principles. Mathematically, Eq. (7) demonstrates the aforesaid process for the seed number one.

$$\boldsymbol{seed_i^1 = X_i + \alpha_i \times (GB \times \beta_i - MG_i \times \gamma_i)\ ,\ i = 1\ to\ n} \quad (8)$$

where GB and MG_i are the best global solutions found so far, and the average values derived from a subset of chosen qualified seeds. α_i denotes a factorial generated randomly used to simulate movement constraints for the seeds. γ_i and β_i are 2 integer parameters for modeling the dice rolling probabilities, which can be selected between 0 and 1.

3 blue and 3 red sides on a dice are used by the second seed (GB). The seed travels in the direction of Xi (blue) or MGi (red), according to the color rolled. Movement is modeled similarly to the first seed, limited by randomly generated factors. This process is formulated using the following equation:

$$\boldsymbol{seed_i^2 = GB + \alpha_i \times (GB \times \beta_i - MG_i \times \gamma_i)\ ,\ i = 1\ to\ n} \quad (9)$$

where α_i, γ_i and β_i are the same parameters used in Eq. (7).

The third seed referred to as MG_i, employs a die featuring both blue and green faces. Its movement direction is determined by the color rolled, with blue directing it towards Xi and green towards GB. A random number-generating function is used to do this, and its result is either 0 or 1, representing the option of blue or green faces. Most importantly, the seed can move along the lines that join GB and Xi. Eq. (9) presents the mentioned procedure for the third seed:

$$\boldsymbol{seed_i^3 = MG_i + \alpha_i \times (GB \times \beta_i - MG_i \times \gamma_i)\ ,\ i = 1\ to\ n} \quad (10)$$

The 4th seed is created by a different procedure in order to incorporate mutation during position updates of qualifying seeds inside the search space. This seed's position adjustments are executed by introducing random changes to randomly selected decision variables. The mathematical expression of the aforesaid process can be demonstrated as follows:

$$\boldsymbol{seed_i^4 = X_i\left(x_i^k = x_i^k + R\right),\quad k = 1\ to\ d} \quad (11)$$

where k is an integer number selected in the range of 0 to d. R represents a random number uniformly distributed within the range of 0 to 1.

There are 4 different formulations for the variable α_i, which controls the restrictions on seed movement, in order to control and fine-tune the CGO's exploration and exploitation rates.

$$\begin{cases} \boldsymbol{\alpha_i = r} \\ \boldsymbol{\alpha_i = 2 \times r} \\ \boldsymbol{\alpha_i = 1 + (\delta \times r)} \\ \boldsymbol{\alpha_i = \sim\varepsilon + (\varepsilon \times r)} \end{cases} \quad (12)$$

where r is a random number that is distributed uniformly in the range 0 to 1, and ε and δ are 2 integer numbers selected in the range of 0 to 1.

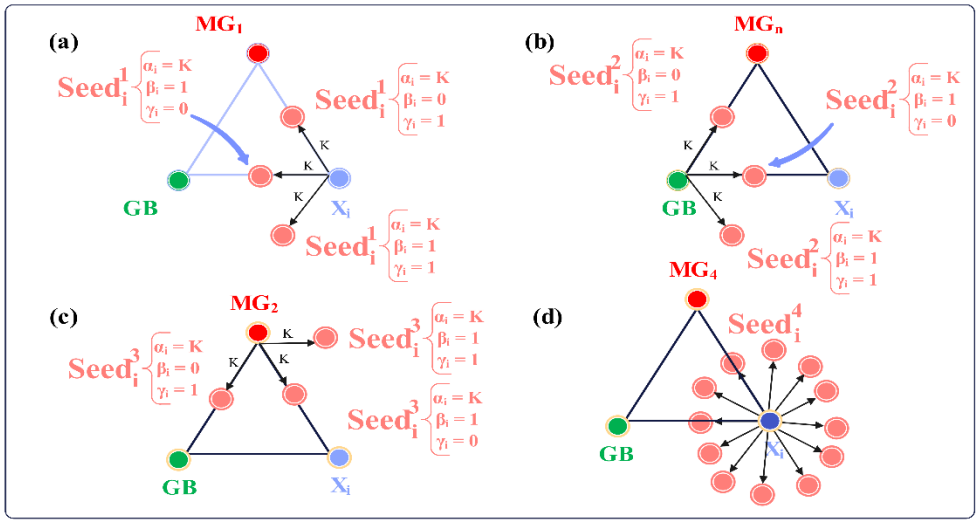


Figure 2: The process of position updating in the search space for a) first seed b) second seed c) third seed d) 4th seed.

To manage self-similarity in fractals, evaluating both existing and new seeds is crucial to determine their appropriateness for the search space. New solutions are compared with current ones, retaining superior ones while discarding those with lower fitness values, indicative of higher self-similarity. The substitution process simplifies model complexity in mathematical approaches. In practice, all eligible seeds found are used to construct the complete Sierpinski triangle shape (Fig. 2). [18], [19].

2.4 Particle swarm optimization algorithm

PSO is a computational optimization technique inspired by the social behavior of bird flocking or fish schooling. It was originally developed by Dr. Eberhart and Dr.

Kennedy in 1995 [20]. PSO is a population-based optimization algorithm that iteratively improves a set of candidate solutions (particles) according to a fitness function, aiming to find the optimal solution.

In PSO, each potential solution to the optimization problem is represented as a "particle" in an n-dimensional search space. Each particle has a position and a velocity vector, which are updated iteratively based on its own best-known position and the best-known position in the entire swarm. The position of a particle corresponds to a potential solution to the optimization problem, and the fitness of that solution is evaluated using an objective function.

At each iteration, particles adjust their velocities based on their previous velocities, where the equation below expresses it mathematically:

$$\boldsymbol{v_{id}^{f+1} = v_{id}^{f} + C_1 r_1^f \left(Pb_{id}^{f} - x_{id}^{f}\right) + C_2 r_2^f \left(Gb_{id}^{f} - x_{id}^{f}\right)} \quad \textbf{(13)}$$

where the current velocity of the particle is shown as v_{id}^{f}, with f times of iterations in a search space with d number of dimensions, and v_{id}^{f+1} is the updated velocity of the particle. Pb_{id}^{f} is the best-known position of particle i (personal best), and Gb_{id}^{f} is the best-known position of any particle in the swarm (global best). C_1 and C_2 are denoting the acceleration coefficients, and r_1^f and r_2^f are random numbers uniformly distributed in the range of 0 to 1.

According to the PSO algorithm, each particle updates its position through the following equation:

$$\boldsymbol{x_{id}^{f+1} = x_{id}^{f} + v_{id}^{f+1}} \quad \textbf{(14)}$$

x_{id}^{f+1} is the updated position of particle i. x_{id}^{f} is referred to the current position of particle i. v_{id}^{f+1} denotes the velocity of particle i, which can be calculated using Eq. (12) [21].

It iterates until any of the termination criteria, such as maximum number of iterations or satisfactory solution quality, are reached. The advantage of PSO is that it is very efficient in exploring complicated search areas and it is easy and convenient to use [20], [21].

2.5 Phasor particle swarm optimizer

PPSO is a PSO algorithm variant that has been proposed by Ghasemi et al. [22]. The basic idea of PPSO introduces a new mechanism that was inspired by the phasor theory in mathematics.Every particle in the PPSO algorithm is assigned an important parameter: phase angle, θ. It acts as an interface towards controlling and steering the movement of particles within the search space. In other words, the phase angle is one of the main ingredients of the whole control process in the behavior of the particles during the process of optimization.

Phase angle is one of the crucial parameters in PPSO, as it bears the signature of direction and amplitude over the particle velocities. The manipulation of phase angle may change the velocities of PPSO particles. This adjustment is typically achieved through an equation that incorporates the phase angle into the calculation of the particle velocities. [23]:

$$\boldsymbol{V_i^T = p(\theta_T) \times (Pb - X_T) + g(\theta_T) \times (Gb - X_T)} \quad \textbf{(15)}$$

The mechanism of position updating in PPSO is presented using the following mathematical equation:

$$\boldsymbol{X_{T+1} = X_T + V_T} \quad \textbf{(16)}$$

where all the parameters are similar to the parameters in the basic PSO. T is the current iteration and the parameters $p(\theta_T)$ and $g(\theta_T)$ can be calculated through the following equations:

$$\boldsymbol{p(\theta_T) = |\cos(\theta_T)|^{2\times \sin(\theta_T)}} \quad \textbf{(17)}$$

$$\boldsymbol{g(\theta_T) = |\sin(\theta_T)|^{2\times \cos(\theta_T)}} \quad (18)$$

Alongside updating the position and velocity, the phasor angle will also be adjusted using the equation below:

$$\boldsymbol{\theta_{T+1} = \theta_T + |\cos(\theta_T) + \sin(\theta_T)| \times 2 \times \pi} \quad \textbf{(19)}$$

In other words, in PSO, the phase angle is a guiding parameter that orients the flight of particles toward regions of interest in the search space. Since the particle's velocity is dynamically updated according to this phase angle, it is then possible for the algorithm to explore complex optimization surfaces in the search for optimal or near-optimal solutions.

To clarify the relation between standard PSO and its phasor-based variant (PPSO), it is important to note that the phase angle θ in PPSO can be interpreted as an implicit generalization of the stochastic influence mechanisms present in classical PSO. While traditional PSO relies on random multipliers r_1 and r_2 to modulate cognitive and social components of particle motion, PPSO encapsulates such modulation into deterministic trigonometric functions of the phase angle, i.e., $p(\theta_T)$ and $g(\theta_T)$. These functions control the intensity and direction of movement toward both personal best (Pb) and global best (Gb) positions. Thus, the phase angle serves not as a completely external addition, but as a mathematically grounded and dynamic extension of existing update rules in PSO. This formulation offers more control over convergence behavior and exploration patterns, especially in complex or multimodal search spaces.

2.6 PAWN sensitivity index

Pianosi and Wagener (2015) [24] Introduce a new method for defining and computing a density-based sensitivity index, PAWN. If some uncertainty about an input is vanished, the output *CDF* variance represents the impact of that input. Precisely, the computation of KS between the conditional distribution and the empirical unconditional *CDF* of model output, when x_i is kept constant, is required.

From these individual results, the maximum, median, or other quantile of the sensitivity index statistic can be obtained, while the KS statistic is computed under various conditioning scenarios.

This index makes it possible to filter out non-influential inputs and rank inputs according to how much they contribute to output uncertainty.

The benefit of PAWN is that it may be used with any kind of output distribution, including ones that are extremely skewed. Also, researches show that the PAWN index is highly sensitive to design parameters in models, compared to other indices [25]. Furthermore, it is adaptable to concentrate on particular output sub-ranges, such as extreme values, which is very helpful for applications like natural hazard models [24].

2.7 Hybrid models & dataset description

Aiming to forecast electricity consumption, 2 ML-based algorithms are adopted including CatBoost and SVR. To increase the performance of the algorithms, the adopted models are integrated with 2 optimizers. Hence, optimizers such as CGO and PPSO are selected for this purpose. The hyperparameters for ML models used, with search spaces defined as follows: for CatBoost – learning rate ∈ [0.01, 0.3], depth ∈ [3, 10], iterations ∈ [100, 1000]; for SVR – C ∈ [0.1, 100], epsilon ∈ [0.01, 0.5], gamma ∈ [0.001, 1]. Finally, 4 hybrid models such as CatBoost-CGO, CatBoost-PPSO, SVR-CGO, and SVR-PPSO are created and compared.

Several input data points were selected to train and test the hybrid models. The datasets contain various types of information and raw data, including the date of data collection, weather conditions, temperature, and more. The dataset was divided into 80% training and 20% testing sets using time-based chronological splitting, ensuring no data leakage between phases and preserving the temporal dependencies essential. This information was adapted from a study conducted by Salehin et al. in 2024 [26], where the datasets were sourced from Bangladesh's governmental website. Table 2 introduces the datasets used as input for the hybrid models, along with the expected outputs required from the models.

Table 2: Input and output data description [26]

Name of the data	Description
Day	Implies the day on which the data is recorded.
Month	Denotes the month in which data is recorded.
Year	Shows the year that the data is recorded.
Temp (C)	Temperature at a specific location using the sensor. (Celsius ℃)
Temp_max (C)	The maximum temperature recorded during a particular timeframe. (Celsius ℃)
Temp_min (C)	Minimum temperature recorded during a particular timeframe. (Celsius ℃)
Temp_ave (C)	Average temperature recorded during a particular timeframe. (Celsius ℃)
Surface pressure (Pa)	The air pressure measured at ground level.
Wind_speed_max (m\s)	Maximum wind speed measured at 50 m above the earth’s surface.
Wind_speed_min (m\s)	Minimum wind speed measured at 50 m above the earth’s surface.
Wind_speed_ave (m\s)	Average wind speed measured at 50 m above the earth’s surface.
Prectotcorr	Measurement associated with precipitation or correlation involving precipitation.
Total demand (MW)	Total quantified electricity demand. (Megawatts)
Max generation (MW)	Maximum electricity generation documented. (Megawatts)

The aforementioned datasets are in different levels of correlation with each other. Some of the data are more correlated with the total demand and max generation, i.e., they have more influence on the measured amount of electricity demand and electricity generation. The result of correlation analysis on the variables of input data is illustrated as a heat map which is presented in Fig. 3:

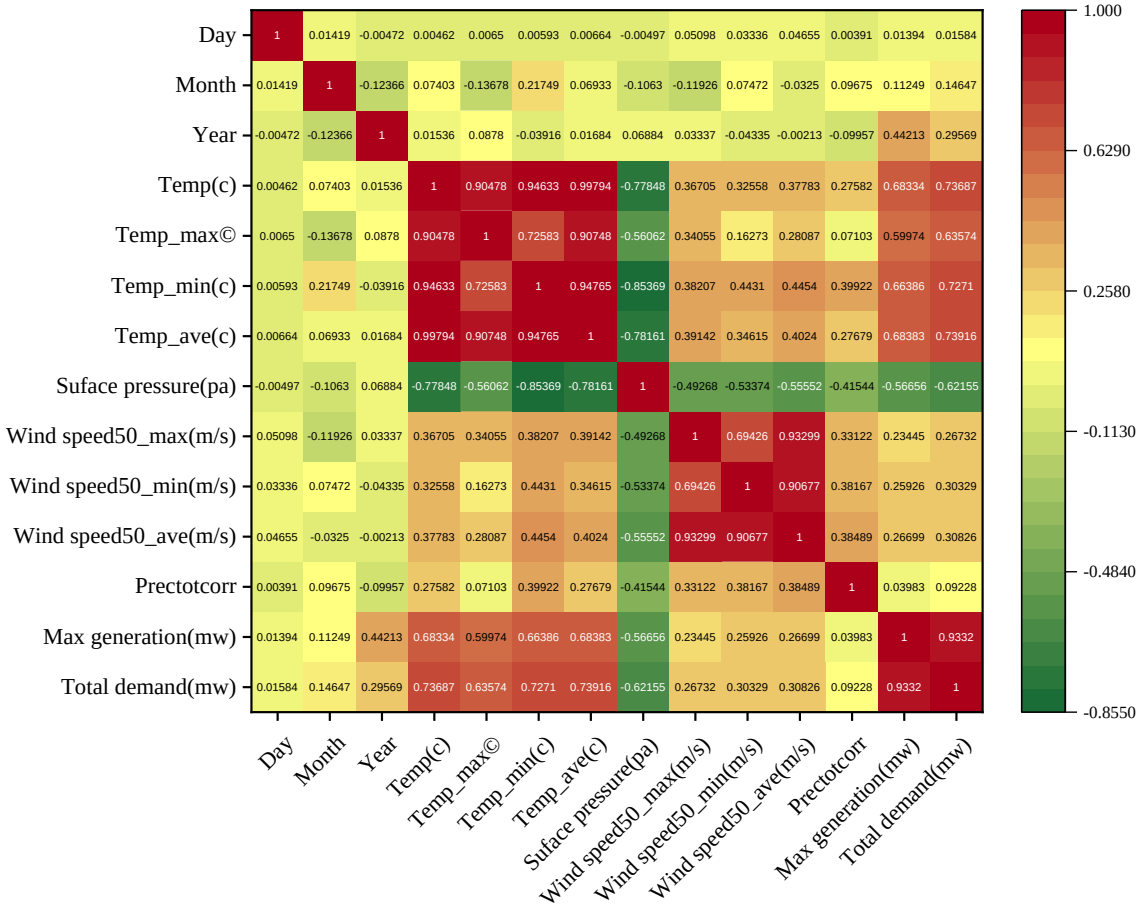


Figure 3: The correlation among variables of input data

According to Fig. 3, the heat map determines the correlation among the variables, with warmer colors denoting high correlation and colder colors denoting low correlation between the 2 variables. Based on this, maximum generation and total electricity demand are correlated highly with each other. Both maximum generation and total demand show a high correlation with variables related to temperature, while their correlation with surface pressure is relatively the lowest. As a matter of fact, the surface pressure demonstrates a low correlation with most of the variables.

Moreover, the heat map implies that wind speed has a higher correlation with electricity generation and electricity demand, compared to precipitation. Also, the correlation between total demand and temperature is partly higher than the correlation between maximum generation and temperature.

In order to determine how uncertainties in input data, affect the outputs of the forecasting models, a sensitivity analyze implemented. By executing the PAWN sensitivity analysis on the inputs of Table 1, the following results are achieved.

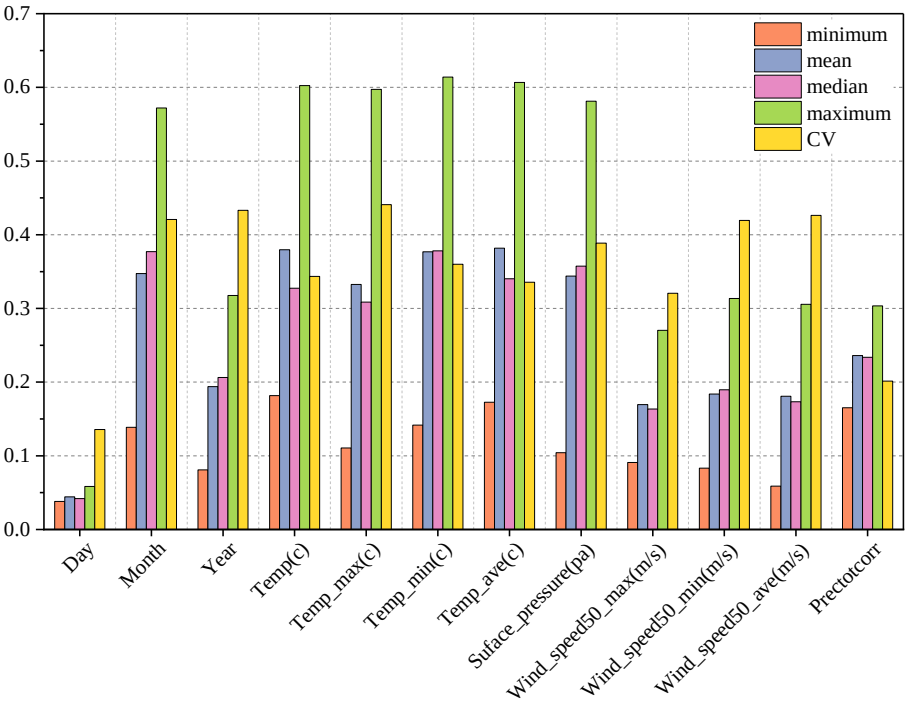


Figure 4: PAWN sensitivity analysis for predicting total demand

Fig. 4 Implies that the variables related to the temperature are showing higher sensitivity to the model, i.e. predicting total electricity demand by the hybrid models is mostly under the influence of temperature. According to the graph, the month of the recording of the data has a high impact on outputs, while the day of the recording data is not effective on the outputs.

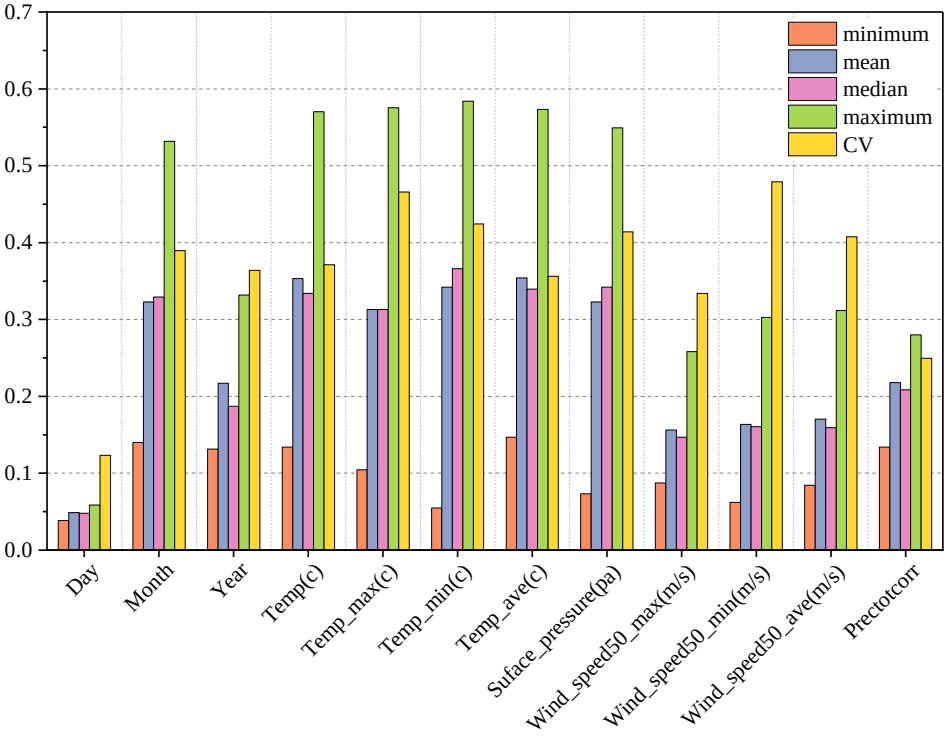


Figure 5: PAWN sensitivity analysis for predicting maximum generation

The results of the PAWN sensitivity analysis for predicting the maximum electricity generation (Fig. 5), demonstrate similar results as the total demand in Fig. 4. The variables related to the temperature, contribute more to the prediction values of prediction of maximum generation. Also, the month of the recording data has a higher impact on the outputs, rather than the variables such as recording day and year. Accordingly, the precipitation is more effective than the wind speed, to the maximum electricity generation.

3 Numerical analysis results

The results of the performance of the models in forecasting total electricity demand and maximum electricity generation are stated in the following. Furthermore, the error values of each hybrid model are compared and datasets division to test and train is noted.

3.1 Total demand forecasting results

The 4 hybrid models including CatBoost-PPSO, CatBoost-CGO, SVR-PPSO, and SVR-CGO are trained and tested through various input data, aiming to predict the electricity demand. Fig. 6 displays the comparison between the target value and predicted value accomplished by the hybrid models, along with the obtained error rates.

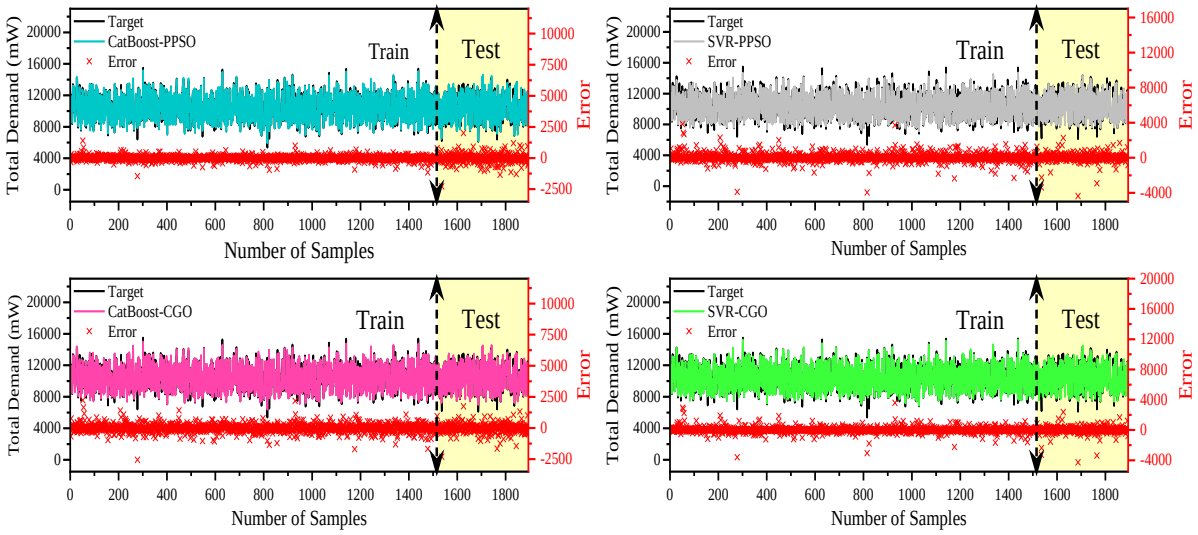


Figure 6: Comparison of target values and predicted values by hybrid models

Fig. 6 denotes how coincided are the prediction values of models are, with the expected values. This figure clearly distinguishes between the training and testing periods, and presents the predicted and actual electricity demand for all hybrid models. The result of the testing process shows that the prediction values of the CatBoost-PPSO are on the same wavelength as the target values, i.e., the CatBoost-PPSO performed more effective in forecasting the electricity demand, compared to the other hybrid models. The red curve represents the prediction error. the graphs of error rates indicate lower error values for CatBoost-PPSO for training samples, while according to Fig. 6, the error values of testing samples are almost the same (PPSO stands for Phasor Particle Swarm Optimizer, and CGO stands for Chaos Game Optimization).

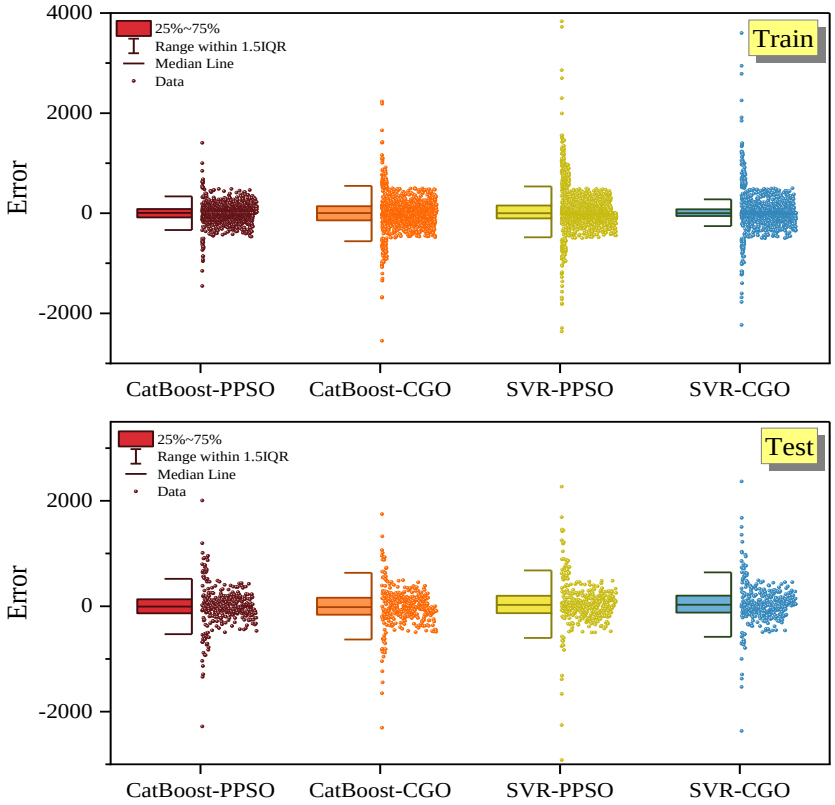


Figure 7: Range of error values for hybrid models on forecasting total demands

Fig. 7 demonstrates the dispersion of obtained error rates during the training and testing process. Accordingly, The CatBoost-PPSO reveals that the error rates of test samples are concentrated more tightly around 0, indicating more accurate forecasting by the model. CatBoost-CGO, SVR-PPSO, and SVR-CGO are in the next places, respectively. As can be seen in Fig. 7, although the SVR-CGO displays satisfying results in training, the error rates of this model in testing are comparatively higher than the other models.

The distribution of error values of each hybrid model is illustrated in more detail in Fig. 8:

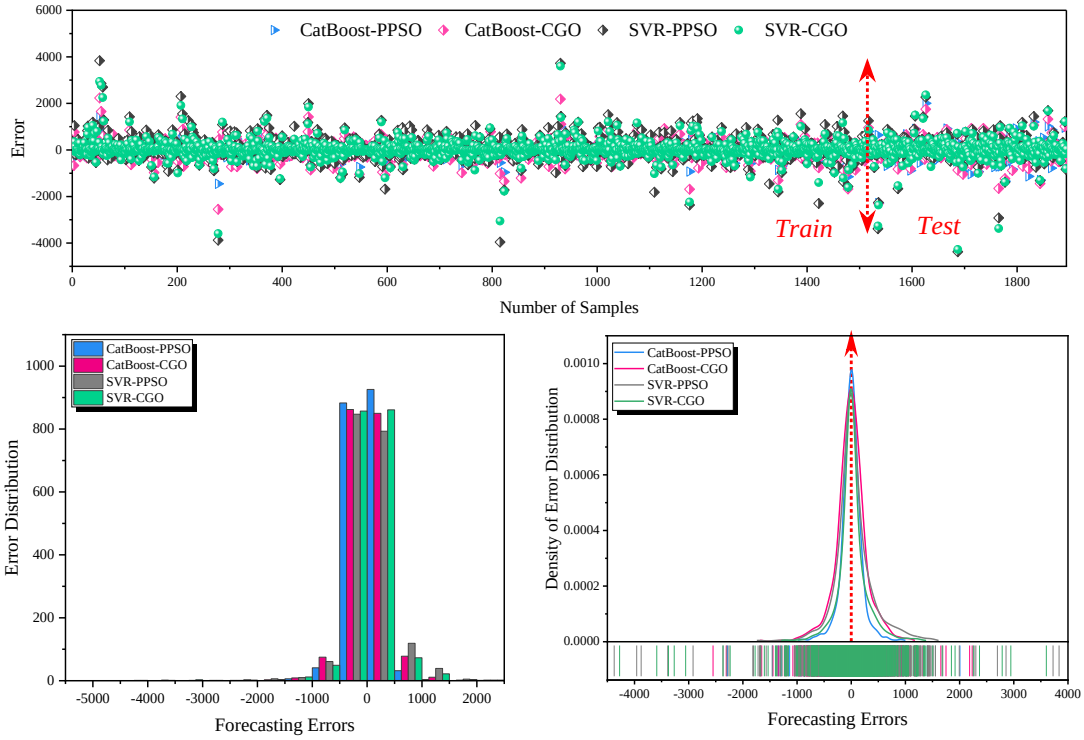


Figure 8: Comparison of error rates of the hybrid models on predicting total demand

The graphs in Fig. 8 demonstrate the comparison of the aforementioned error rates. Fig. 8 clarifies that the CatBoost-PPSO is more precise in prediction since its graph is more convergent to 0 and the obtained error values of samples are less dispersed.

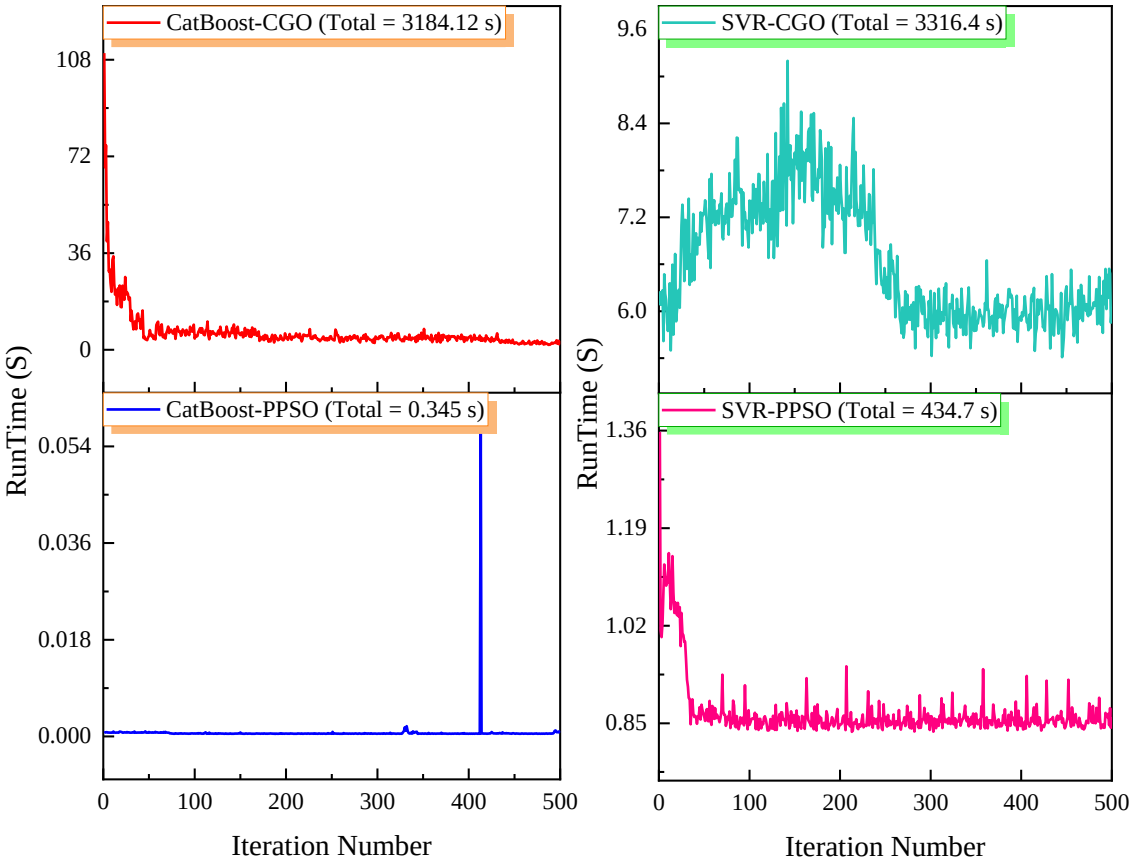


Figure 9: Comparing the runtime of the hybrid models during the prediction of total demand

According to Fig. 9, the CatBoost-PPSO performed the prediction with the shortest runtime. The models such as CatBoost-CGO and SVR-CGO, displayed disappointing results based on runtime. The runtime of the CatBoost-PPSO has remarkably outperformed the rest of the models, indicating the robustness and precision of the hybrid model.

3.2 Maximum electricity generation prediction results

The 4 aforementioned hybrid models have performed a prediction on maximum electricity generation where the results and error rates are presented in the following.

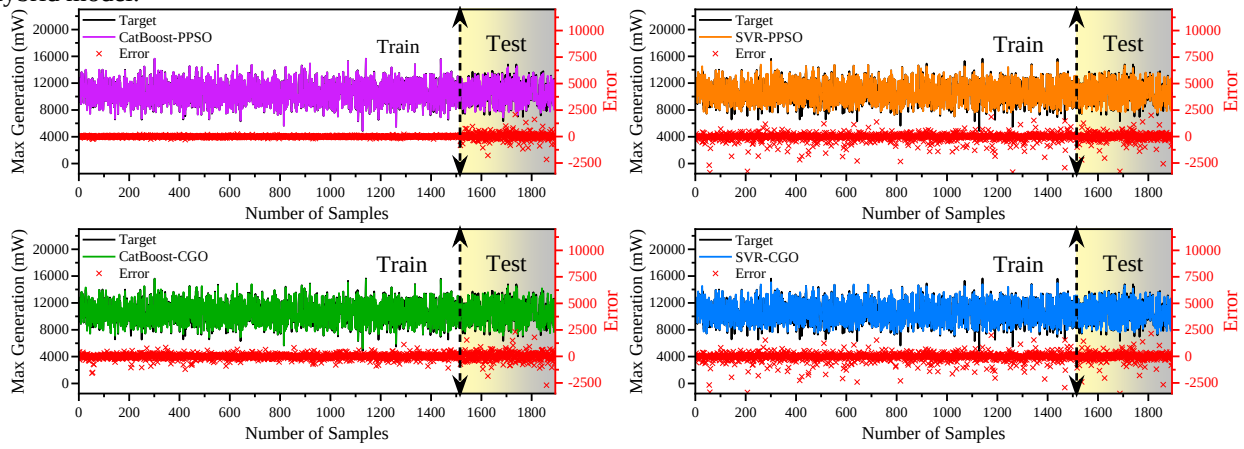


Figure 10: Comparison of target values and predicted values by hybrid models

The performance of the hybrid models in forecasting maximum generation is compared to the expected values and illustrated in Fig. 10. This visualization compares actual and predicted maximum electricity generation for different hybrid models, over both training and test datasets. Each subplot includes the target values, predicted

outputs, and associated error plots. As can be seen, the CatBoost-PPSO shows notable results in predicting the training samples. Accordingly, the CatBoost-PPSO shows comparatively lower error values during the testing process.

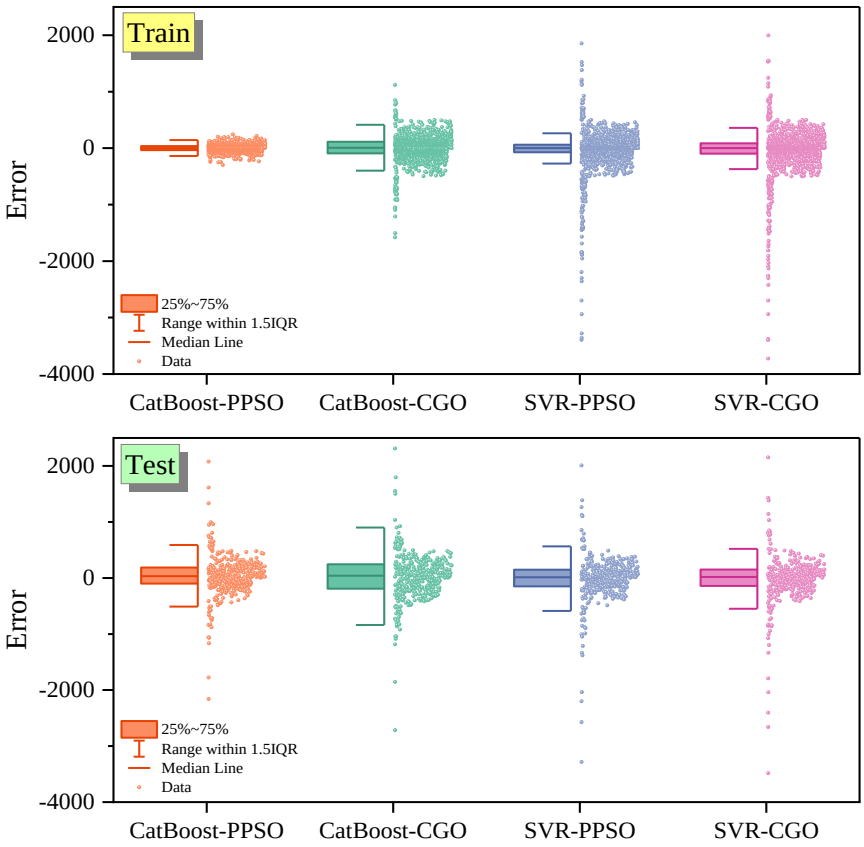


Figure 11: Range of error values for hybrid models on forecasting max generation

The dispersion of error rates of the hybrid models in Fig. 11, is indicating that the errors produced by CatBoost-PPSO during maximum generation forecasting, are so low. On the other hand, the error values of CatBoost-CGO during the test, are more dispersed and there are more outliers, indicating the low precision of CatBoost-CGO in forecasting max generation.

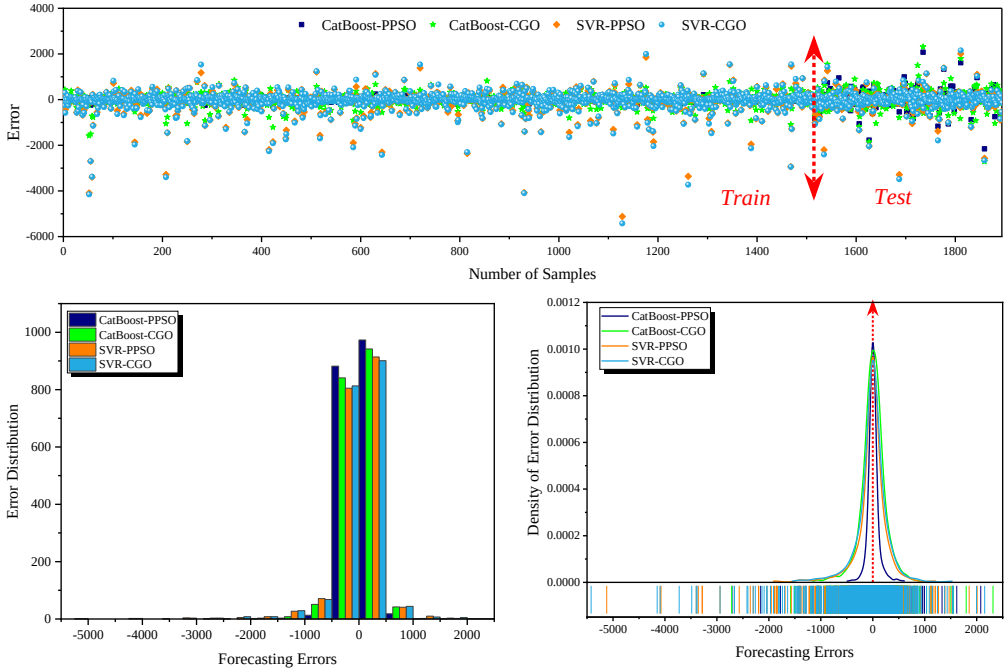


Figure 12: Comparison of error rates of the hybrid models on predicting max generation

Fig. 12 compares the distribution and density of obtained error values by the hybrid models in one graph. Evidently, the error values of CatBoost-PPSO are concentrated to 0 and less distributed, admitting the fact that the CatBoost-PPSO shows outstanding performance in forecasting the maximum electricity generation.

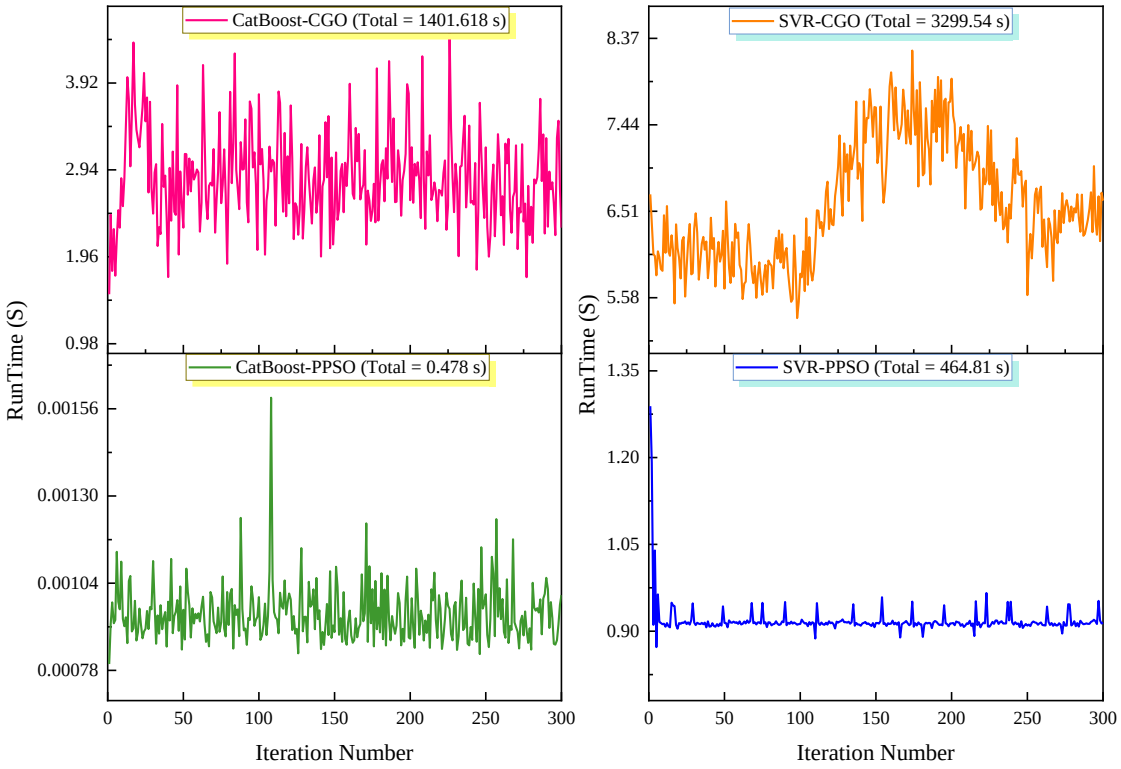


Figure 13: Comparing the runtime of the hybrid models during the prediction of max generation

Comparing the runtime of the models during the prediction of the maximum electricity generation denotes the best performance of the CatBoost-PPSO. Fig. 13 indicates that this hybrid model is significantly faster than the remaining models with a runtime of 0.478s after 300 iterations.

The models are analyzed according to several statistical evaluation indices the description of the metrics along with the results of the analysis are presented in the next section.

4 Analysis of the models based on the statistical evaluation indices

The hybrid ML-based models are assessed using several statistical evaluation indices. The accuracy of the models in predicting the total electricity demand and maximum electricity generation can be obtained using various metrics. Hence, the following indices are selected for this purpose. As shown in Table 3, the proposed statistical evaluation indices are detailed through specific equations.

Table 3: Equations of the proposed statistical evaluation indices

Statistics	Name	Equation
MSE	Mean Squared Error	$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
RMSE	Root Mean Square Error	$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$
NRMSE	Normalized Root Mean Square Error	$NRMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i)^2}}$
MAPE	Mean Absolute Percentage Error	$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left\|\frac{y_i - \hat{y}_i}{y_i}\right\|$
MAE	Mean Absolute Error	$MAE = \frac{\sum_{i=1}^{n}\|y_i - \hat{y}_i\|}{n}$
RAE	Relative Absolute Error	$RAE = \frac{\sum_{i=1}^{n}\|y_i - \hat{y}_i\|}{\sum_{i=1}^{n}\|y_i - \bar{y}\|}$
R2	Coefficient of Determination	$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$
n is the number of data points, y_i and $\hat{y}_i$ are i-th actual value and i-th estimated value, respectively, and $\bar{y}$ is the mean of the actual values.		

Each index, evaluates the performance of predictive models, particularly in regression analysis and forecasting. The metrics assessed the performance of the models in predicting total electricity demand and maximum electricity generation, separately.

4.1 Total demand forecasting results

To measure the difference between the predicted values of total demand and the real total demand, the R^2 is exerted. The fitness of the hybrid models to the data is illustrated in the following figure.

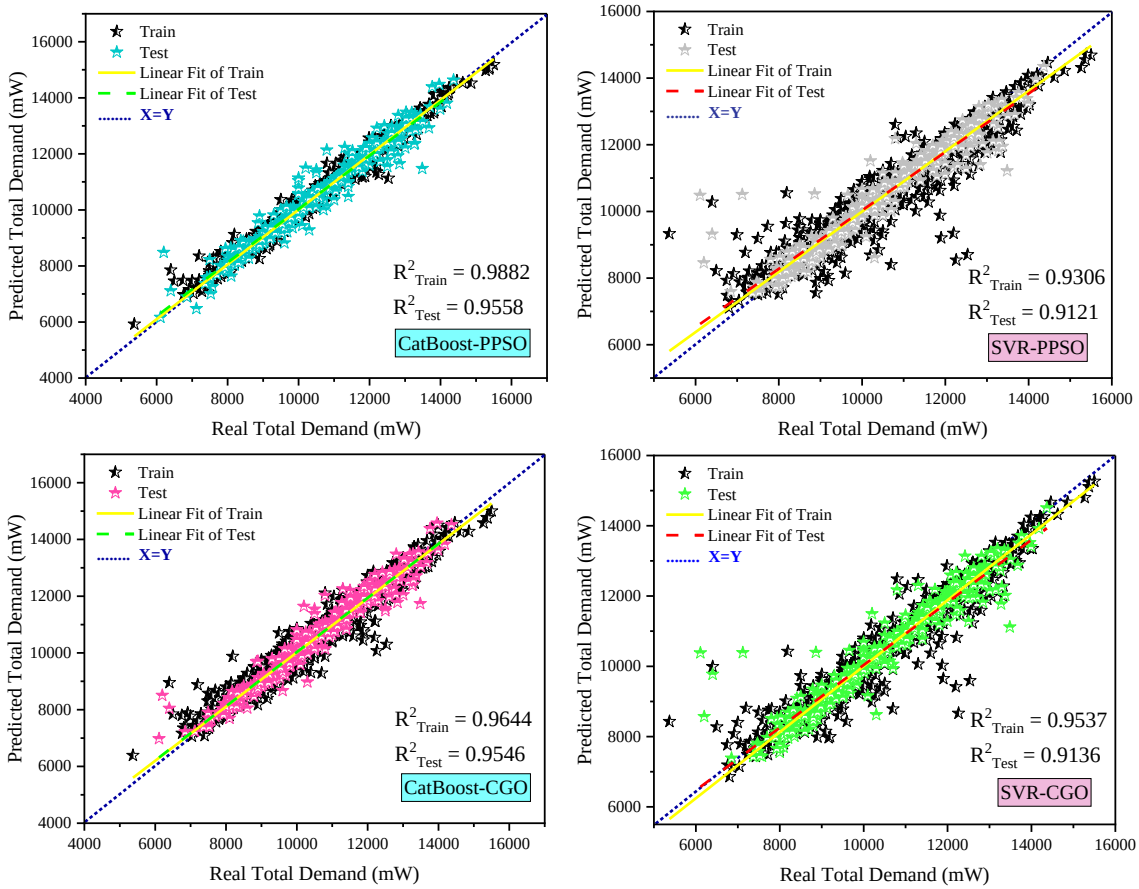


Figure 14: Comparing the goodness of fit of a model to the real data of total demand

The regression analysis using R2 in Fig. 14, indicates the promising result of CatBoost-PPSO for the data of test. The CatBoost-CGO also demonstrates satisfying results, denoting the distinction of CatBoost to the SVR in the given task.

The results of the evaluation based on MAE, MAPE, RAE, RMSE, and NRMSE are presented in Fig. 14. The obtained R2 values are also compared in a singular graph which is presented.

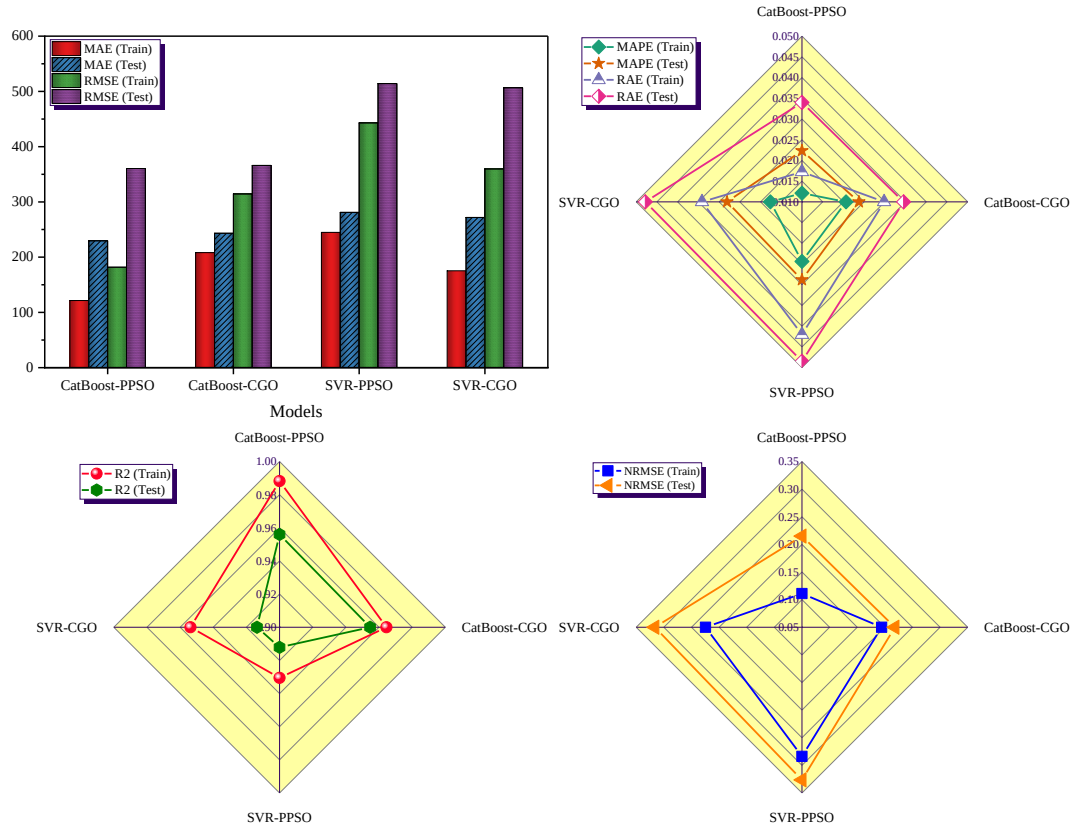


Figure 15: Comparing the obtained values of R2, MAE, MAPE, RAE, RMSE, and NRMSE for total demand forecasting

According to Fig. 15, the results of the testing process imply the low error rates and precise performance of the CatBoost compared to the SVR. Furthermore, the integration of CatBoost with PPSO outperforms the CatBoost-CGO, relatively. The CatBoost-PPSO exceeds the rest of the models in performance and accuracy.

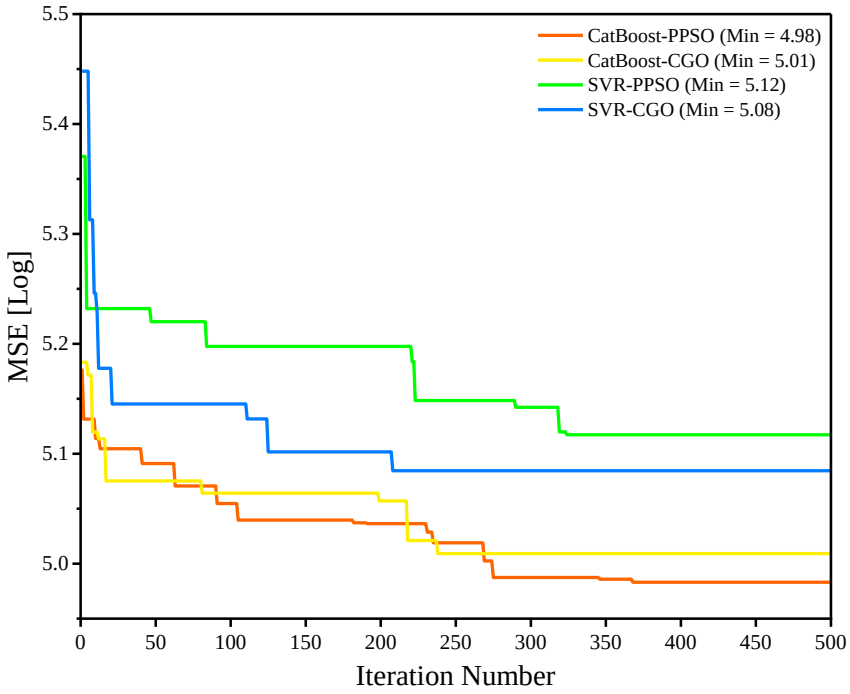


Figure 16: The obtained MSE values after 500 iterations in forecasting the total demand

The Fig. 16 shows the obtained MSE values after 500 epochs. According to the results, the CatBoost-PPSO displays fewer MSE values at earlier iterations. Hence, the CatBoost-PPSO model performs more beneficial in predicting the target values.

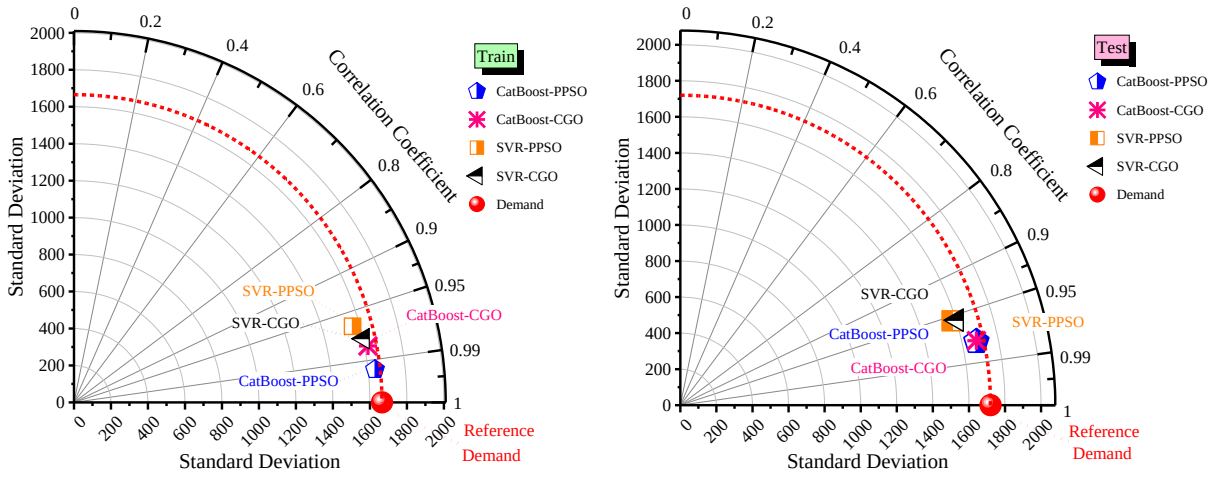


Figure 17. The Taylor diagrams of the hybrid models in predicting the total demand

Fig. 17, determines the correlation between the target value and the predicted values by the models. according to Fig. 17, the prediction values of the CatBoost-PPSO are closer to the line of target values (demand). This indicates the accuracy of the models in the prediction of the total electricity demand.

The obtained values of the evaluation indices are presented in Table 4, which summarizes the aforementioned figures.

Table 4: The obtained values of the statistical evaluation indices for predicting total demand

CatBoost-PPSO	Tain		Test		CatBoost-CGO	Tain		Test	
	MAE	121.4943	MAE	229.6541		MAE	208.2137	MAE	243.2856
	RMSE	181.7452	RMSE	360.4203		RMSE	314.3134	RMSE	366.0203
	MAPE	0.012091	MAPE	0.02234		MAPE	0.020638	MAPE	0.023736
	R2	0.988195	R2s	0.955997		R2	0.964452	R2s	0.954626
	RAE	0.017242	RAE	0.034001		RAE	0.029818	RAE	0.034529
	NRMSE	0.111016	NRMSE	0.214936		NRMSE	0.194235	NRMSE	0.218172
SVR-PPSO	Tain		Test		SVR-CGO	Tain		Test	
	MAE	244.8789	MAE	281.0936		MAE	175.2804	MAE	271.7475
	RMSE	443.0239	RMSE	514.0837		RMSE	359.6045	RMSE	506.6743
	MAPE	0.024369	MAPE	0.028811		MAPE	0.017614	MAPE	0.028079
	R2	0.930567	R2s	0.912098		R2	0.953707	R2s	0.913594
	RAE	0.042029	RAE	0.048497		RAE	0.034115	RAE	0.047798
	NRMSE	0.283801	NRMSE	0.326346		NRMSE	0.224504	NRMSE	0.316315

To assess the significance of model performance differences in the Total Demand objective, the Wilcoxon signed-rank test was applied using Absolute Error (AE) values. As shown in Table 5, the comparison between Catboost-PPSO and SVR-PPSO revealed a statistically significant advantage for Catboost-PPSO ($p = 0.0014$), indicating better predictive accuracy. Conversely, the comparison between Catboost-CGO and SVR-CGO did not result in a statistically significant difference ($p = 0.9496$), suggesting similar performance between the two models for this objective.

Table 5: Wilcoxon signed-rank test results for model comparison in total demand

Algorithm	Model 1	Model 2	Median Difference (AE_M1 - AE_M2)	P-value	Conclusion
PPSO	Catboost-PPSO	SVR-PPSO	-28.7344	0.0014	Catboost-PPSO Significantly Better (Lower AE)
CGO	Catboost-CGO	SVR-CGO	12.3338	0.9496	No Significant Difference

4.2 Maximum generation forecasting results

The foregoing statistical evaluation indices evaluate the maximum electricity generated along with the total demand. The results of the prediction accomplished with the hybrid models are compared based on the error rates and the difference between the prediction and the real values.

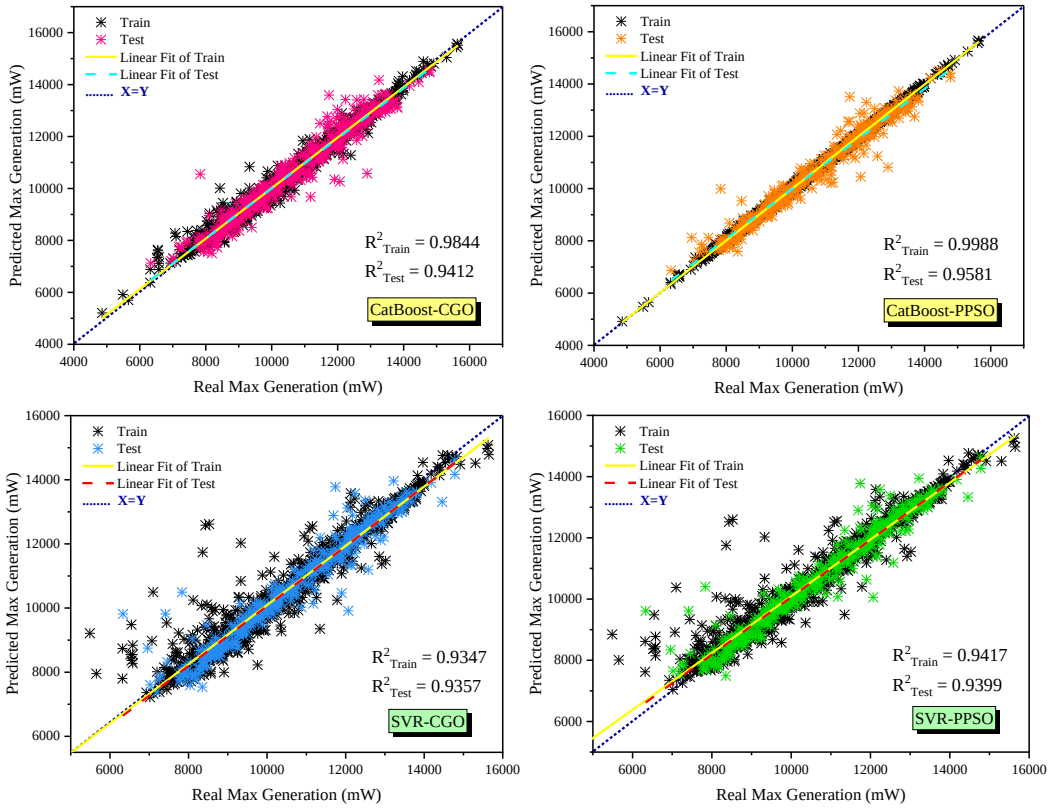


Figure 18: Comparing the goodness of fit of a model to the real data of the max generation

The regression graphs in Fig. 18 demonstrate that the predicted max generation by the CatBoost- PPSO is coincident with the real values with an R2 test value of 0.9581 which outperforms the other models.

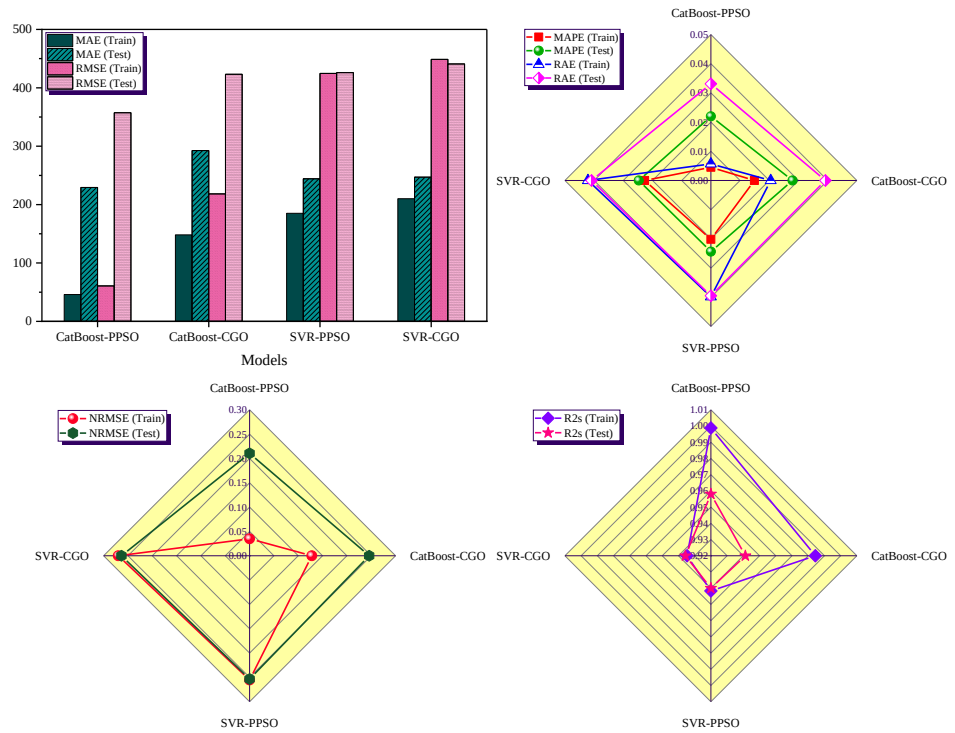


Figure 19: Comparing the obtained values of R2, MAE, MAPE, RAE, RMSE, and NRMSE for the max generation forecasting

By comparing the error values of the hybrid models illustrated in Fig. 19, the CatBoost-PPSO model has lower error values during the prediction of the maximum electricity generation. In addition to CatBoost-PPSO, the SVR-PPSO also showed promising results, particularly in terms of MAE. However, the R2 value for SVR-PPSO is relatively low.

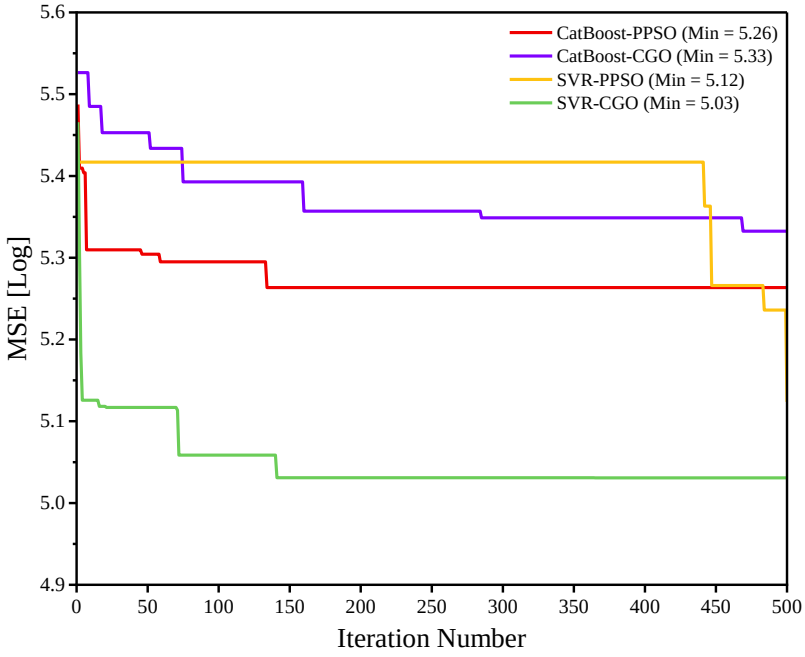


Figure 1: The obtained MSE values after 500 iterations in forecasting the max generation

By comparing the obtained MSE values in Fig. 20, the SVR-CGO displays notably lower MSE rates after 500 iterations. In addition, SVR-PPSO and then CatBoost-PPSO are in the next places.

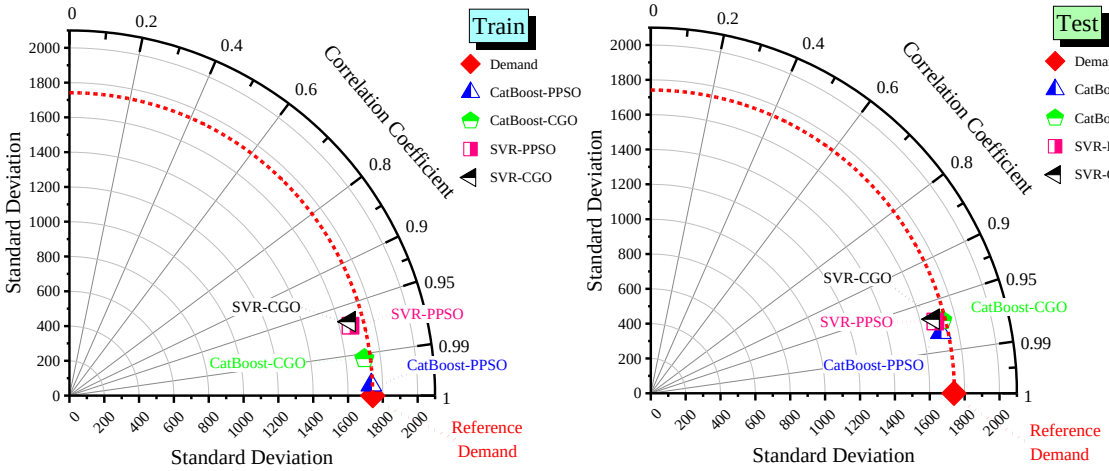


Figure 2: The Taylor diagrams of the hybrid models in predicting the max generation

According to Fig. 21, the standard deviation graphs denoting the performance of the models on predicting the max generation indicate better performance of the CatBoost-PPSO on data of train. The results of the testing process show that the models almost performed similarly and there is no noticeable difference, as can be deduced from the graph. Although, the numerical values in Table 5, denote better results for CatBoost-PPSO.

Table 5 presents the obtained values of the indices and summarizes the figures above:

Table 5: The obtained values of the statistical evaluation indices for predicting the maximum generation

CatBoost-PPSO	Tain		Test		CatBoost-CGO	Tain		Test	
	MAE	45.9977	MAE	229.3393		MAE	148.1471	MAE	292.5621
	RMSE	60.69213	RMSE	357.2295		RMSE	218.3039	RMSE	423.2141
	MAPE	0.004515	MAPE	0.02197		MAPE	0.014922	MAPE	0.02796
	R2s	0.998796	R2s	0.958087		R2s	0.984419	R2s	0.94124
	RAE	0.005696	RAE	0.033127		RAE	0.020489	RAE	0.039245

	NRMSE	0.034997	NRMSE	0.210455		NRMSE	0.127899	NRMSE	0.245781
SVR-PPSO	Tain		Test		SVR-CGO	Tain		Test	
	MAE	185.0795	MAE	244.2409		MAE	210.0944	MAE	247.0029
	RMSE	424.6411	RMSE	425.9695		RMSE	448.6983	RMSE	440.8853
	MAPE	0.020161	MAPE	0.024377		MAPE	0.022707	MAPE	0.024697
	R2	0.941689	R2s	0.940021		R2	0.934701	R2s	0.93571
	RAE	0.039855	RAE	0.039501		RAE	0.042112	RAE	0.040884
	NRMSE	0.25535	NRMSE	0.253392		NRMSE	0.269826	NRMSE	0.263338

Tables 3 and 4 indicate the substantial performance of the CatBoost-PPSO model, with the highest R^2 values on the test set and the lowest error metrics among the compared hybrid configurations. This superior performance can be attributed to the gradient boosting structure of CatBoost, which effectively handles complex feature interactions and minimizes overfitting, in conjunction with the exploration-exploitation balance provided by the phasor-enhanced swarm optimization mechanism of PPSO. The consistent reduction in RMSE and MAPE across both total demand and maximum generation forecasts further reinforces its robustness. In addition to predictive accuracy, runtime performance also plays a crucial role in practical deployments. Models with faster convergence and lower computational cost, such as CatBoost-PPSO, are more suitable for real-time or large-scale forecasting environments where rapid decision-making is essential. Hence, CatBoost-PPSO is selected as the preferred model for the given task, balancing both accuracy and applicability in real-world energy systems.

For the Max Generation objective, the Wilcoxon signed-rank test was conducted to evaluate the statistical differences in model performance. As shown in Table Y, SVR-CGO significantly outperformed Catboost-CGO ($p < 0.0001$), achieving lower AE values. However, in the PPSO-based comparison, no statistically significant difference was observed between Catboost-PPSO and SVR-PPSO ($p = 0.7646$).

Table 7: Wilcoxon signed-rank test results for model comparison in max generation

Algorithm	Model 1	Model 2	Median Difference (AE_M1 - AE_M2)	P-value	Conclusion
PPSO	Catboost-PPSO	SVR-PPSO	4.0609	0.7646	No Significant Difference
CGO	Catboost-CGO	SVR-CGO	48.8007	1.356e-06	SVR-CGO Significantly Better (Lower AE)

5 Discussion

The literature on electricity demand and generation forecasting reveals several recurring limitations. Many prior studies have relied on conventional machine learning or statistical methods (e.g., ANN, ARIMA, linear regression) without incorporating robust optimization techniques. Consequently, these models often suffer from limited adaptability to nonlinear demand patterns, lack of generalizability across contexts, and suboptimal parameter tuning resulting in reduced predictive accuracy. Furthermore, a substantial number of prior works failed to provide comprehensive statistical benchmarking using indices like R^2 or NRMSE, limiting comparability.

To address these shortcomings, we proposed four hybrid machine learning models that integrate CatBoost and Support Vector Regression (SVR) with advanced metaheuristic optimizers—Phasor Particle Swarm Optimizer (PPSO) and Chaos Game Optimizer (CGO). These hybridizations were designed to enhance model accuracy, stability, and generalization by leveraging both the learning capabilities of the base models and the exploratory efficiency of the optimizers. All experiments were conducted on a personal computing system running Windows 10 Pro (64-bit, build 19045.2311), equipped with a 12th Gen Intel® Core™ i5-12400F CPU @ 2.50 GHz and 32.0 GB DDR4 RAM. The experimental results, presented in Tables 4 and 5, strongly support the effectiveness of the proposed framework. Among all models, CatBoost-PPSO consistently delivered the best performance, achieving the lowest test RMSE values (360.42 for demand, 357.23 for generation) and the highest R^2 scores (0.956 for demand and 0.958 for generation). Notably, its test MAPE values (0.02234 for demand and 0.02197 for generation) were also significantly lower than those of all competing models. By comparison, even other hybrid combinations such as SVR-PPSO or CatBoost-CGO showed relatively higher error margins. For example, SVR-CGO exhibited a test RMSE of 506.67 for total demand and 440.88 for maximum generation, alongside lower R^2 values (0.913 and 0.935 respectively), confirming the superior synergy between CatBoost and PPSO.

In conclusion, the results indicate that the integration of CatBoost with PPSO not only mitigates the limitations of earlier models—such as underfitting, local optima, and high sensitivity to data noise—but also offers a more scalable and accurate solution for forecasting in complex, real-world electricity networks. Despite the results derived from the suggested hybrid models are, there are some limitations to consider. For one, the models were trained

and validated using a defined dataset that is unique to a specific geographic and temporal context, limiting the applicability of the results to other areas or periods. Second, while metaheuristic optimization methods like PPSO and CGO enhance performance, they are computationally intensive and will not perform well in real-time or high-frequency forecasting setups. Additionally, the models were evaluated in relation to benchmark statistical measures, but additional experiments in extreme demand settings or unusual events could provide more insight regarding robustness and pragmatic reliability. These findings underscore the value of hybrid optimization strategies and provide a strong foundation for future research in energy analytics.

6 Conclusion

Electricity demand and peak generation are some of the major factors that need to be forecasted, as this helps optimize resource allocation, infrastructure planning, and policymaking in energy management. The effective prediction, therefore, of utilities and policy-makers finds its basis in handling issues related to supply-demand imbalance, integration of renewable energy sources, and stability of the grid. Such forecasting is performed by using modern approaches like ML and metaheuristics algorithms that help in attaining better efficiency and robustness in energy systems for supporting sustainable energy transitions.

Precisely, this paper performed a deep forecast of total electricity demand and highest electricity generation using a hybrid approach by combining ML models with metaheuristic algorithms. More precisely, CatBoost and SVR have been combined with PPSO and CGO in order to build 4 hybrid models: namely Catboost-CGO, CatBoost-PPSO, SVR-CGO, and SVR-PPSO.

Further, performance evaluation in an extensive statistical basis using metrics like MSE, MAE, MAPE, RAE, RMSE, NRMSE, and R2 was performed.

It is clearly obvious that the hybrid CatBoost-PPSO model presents a higher predictive ability than the other examined hybrid models in all the considered indices. CatBoost-PPSO achieved the lowest test RMSE (360.42 for total demand and 357.23 for maximum generation), along with the highest R^2 scores (0.955997 and 0.958087, respectively), confirming its superior predictive performance over the other hybrid configurations. This is indicative of how effective the incorporation of CatBoost and PPSO will be in terms of forecasting electricity demand and generation. The obtained error values are considerably lower for the hybrid CatBoost-PPSO metrics.

These results not only underscore the importance of leveraging hybrid approaches but also highlight the significance of selecting appropriate combinations of ML models and metaheuristic algorithms for enhanced forecasting accuracy. Such insights could prove invaluable for practitioners and researchers in the field of energy forecasting, facilitating more precise decision-making and resource allocation in electricity management systems.

Nomenclature

Abbreviation	Description	Abbreviation	Description
SVR	Support Vector Regression	d	Dimension of seeds
SVM	Support Vector Machine	$x_i^j(0)$	The incipient position of each qualified seed
PSO	Particle Swarm Optimizer	$x_{i,\min}^j$	The minimum allowed value for the j-th decision
PPSO	Phasor Particle Swarm Optimizer	$x_{i,\max}^j$	The minimum allowed value for the j-th decision
CGO	Chaos Game optimizer	r	Randomly selected number
ML	Machine Learning	GB	Global best
NNGM	Neural Network-based grey forecasting model	MG_i	Average value
ANN	Artificial Neural Network	γ_i and β_i	Integer parameters for modeling the dice rolling probabilities
BMA	Bayesian Model Averaging	X_i	The chosen seed
LSTM	Long Short-Term Memory	R	Uniformly distributed random number
x_i	Input feature of SVR	v_{id}^f	The current velocity in PSO
y_i	Corresponding output of SVR	v_{id}^{f+1}	Updated velocity of the particle
w	Weight Factor	Pb_{id}^f	The best-known position of the particle so far
$\phi(x)$	Mapping function	Gb_{id}^f	Global best position of the particle
ε	epsilon-insensitive loss function	C_1 and C_2	Acceleration coefficients
b	Bias term	x_{id}^f	Current position of the particle
ξ_i and ξ_i^*	Slack variables	x_{id}^{f+1}	The updated position of the particle
C	Regularization parameter	θ	Phase angle of the PPSO
α_i and α_i^*	Lagrange multiplier	T	The current iteration
$K(x_i, x_j)$	Kernel function	PAWN	Probability of Absolute Worthlessness of Nodes
n	Total number of qualified seeds of CGO		

Acknowledgment:

This work was supported by research and engineering applications of panoramic display and decision analysis technology for power grid fault relay protection based on artificial intelligence.

Competing of interests

The authors declare no competing of interests.

Authorship contribution statement

Jian Wang: Writing-Original draft preparation, Conceptualization, Supervision, Project administration.
Liandian Jiang: Methodology
Pengfei Huang: Validation.
Junyang Tian: Language review
Haiyong Li: Software
Bin Liu: Formal Analysis
Shan Liang: Data Curation

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Author statement

The manuscript has been read and approved by all the authors, the requirements for authorship, as stated earlier in this document, have been met, and each author believes that the manuscript represents honest work.

Ethical approval

All authors have been personally and actively involved in substantial work leading to the paper, and will take public responsibility for its content.

References

[1] J. Tian and S. Cui, "Analysis and Calculation of Marginal Electricity Price of Nodes With Network Loss From the Perspective of Intelligent Robot Considering Digital Signal Processing Technology," *Informatica (Slovenia)*, vol. 48, no. 14, pp. 97–106, Oct. 2024, doi: 10.31449/inf.v48i14.6066.

[2] J. Zupančič, M. Gams, and J. Stefan, "Dynamic Protocol for the Demand Management of Heterogeneous Resources with Convex Cost Functions," 2017.

[3] Y. C. Hu, "Electricity consumption prediction using a neural-network-based grey forecasting approach," *Journal of the Operational Research Society*, vol. 68, no. 10, pp. 1259–1264, Oct. 2017, doi: 10.1057/s41274-016-0150-y.

[4] L. Ekonomou, "Greek long-term energy consumption prediction using artificial neural networks," *Energy*, vol. 35, no. 2, pp. 512–517, 2010, doi: 10.1016/j.energy.2009.10.018.

[5] K. Li, C. Hu, G. Liu, and W. Xue, "Building's electricity consumption prediction using optimized artificial neural networks and principal component analysis," *Energy Build*, vol. 108, pp. 106–113, Dec. 2015, doi: 10.1016/j.enbuild.2015.09.002.

[6] W. Zhang and J. Yang, "Forecasting natural gas consumption in China by Bayesian Model Averaging," *Energy Reports*, vol. 1, pp. 216–220, Nov. 2015, doi: 10.1016/j.egyr.2015.11.001.

[7] N. Fumo and M. A. Rafe Biswas, "Regression analysis for prediction of residential energy consumption," 2015, *Elsevier Ltd*. doi: 10.1016/j.rser.2015.03.035.

[8] S. Barak and S. S. Sadegh, "Forecasting energy consumption using ensemble ARIMA-ANFIS hybrid algorithm," *International Journal of Electrical Power and Energy Systems*, vol. 82, pp. 92–104, Nov. 2016, doi: 10.1016/j.ijepes.2016.03.012.

[9] J. Hwang, D. Suh, and M. O. Otto, "Forecasting electricity consumption in commercial buildings using a machine learning approach," *Energies (Basel)*, vol. 13, no. 22, Nov. 2020, doi: 10.3390/en13225885.

[10] R. Wang, S. Lu, and W. Feng, "A novel improved model for building energy consumption prediction based on model integration," *Appl Energy*, vol. 262, Mar. 2020, doi: 10.1016/j.apenergy.2020.114561.

[11] Z. Lin, L. Cheng, and G. Huang, "Electricity consumption prediction based on LSTM with attention mechanism," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 15, no. 4, pp. 556–562, Apr. 2020, doi: 10.1002/tee.23088.

[12] G. Vijendar Reddy, L. J. Aitha, C. Poojitha, A. Naga Shreya, D. Krithika Reddy, and G. Sai Meghana, "Electricity Consumption Prediction Using Machine Learning," in *E3S Web of Conferences*, EDP Sciences, Jun. 2023. doi: 10.1051/e3sconf/202339101048.

[13] R. Olu-Ajayi and H. Alaka, "Building energy consumption prediction using deep learning."

[14] A. V. Dorogush, V. Ershov, and A. Gulin, "CatBoost: gradient boosting with categorical features support," Oct. 2018, [Online]. Available: http://arxiv.org/abs/1810.11363

[15] G. Huang *et al.*, "Evaluation of CatBoost method for prediction of reference evapotranspiration in humid regions," *J Hydrol (Amst)*, vol. 574, pp. 1029–1041, Jul. 2019, doi: 10.1016/j.jhydrol.2019.04.085.

[16] H. Drucker·, C. J. C. Burges, L. Kaufman, A. Smola··, and V. Vapoik, "Support Vector Regression Machines."

[17] B. Üstün, W. J. Melssen, and L. M. C. Buydens, "Facilitating the application of Support Vector Regression by using a universal Pearson VII function based kernel," *Chemometrics and Intelligent Laboratory Systems*, vol. 81, no. 1, pp. 29–40, Mar. 2006, doi: 10.1016/j.chemolab.2005.09.003.

[18] S. Talatahari and M. Azizi, "Chaos Game Optimization: a novel metaheuristic algorithm," *Artif Intell Rev*, vol. 54, no. 2, pp. 917–1004, Feb. 2021, doi: 10.1007/s10462-020-09867-w.

[19] A. Ramadan, S. Kamel, M. M. Hussein, and M. H. Hassan, "A new application of chaos game optimization algorithm for parameters extraction of

three diode photovoltaic model," *IEEE Access*, vol. 9, pp. 51582–51594, 2021, doi: 10.1109/ACCESS.2021.3069939.

[20] J. Kennedy, R. Eberhart, and bls gov, "Particle Swarm Optimization."

[21] D. N. Wilke, M. Khajehzadeh, and Q. Bai, "Analysis of Particle Swarm Optimization Algorithm Swarm Int elligence based Soft Comput ing Techniques for t he Solut ions t o Mult iobject ive Op… hifza afaq A Survey of t he St at e of t he Art in Part icle Swarm Opt imizat ion A Review of Populat ion-based Met a-Heurist ic Algorit hms Analysis of Particle Swarm Optimization Algorithm."

[22] M. Ghasemi, E. Akbari, A. Rahimnejad, S. E. Razavi, S. Ghavidel, and L. Li, "Phasor particle swarm optimization: a simple and efficient variant of PSO," *Soft comput*, vol. 23, no. 19, pp. 9701–9718, Oct. 2019, doi: 10.1007/s00500-018-3536-8.

[23] khatsu Siseyiekuo, Srivastava Abhishek, and Kumar Das Dushimanta, "An Adaptive Phasor Particle Swarm Optimization to Solve Economic Load Dispatch and Combined Emission Economic Load Dispatch Problem," 2019.

[24] F. Pianosi and T. Wagener, "A simple and efficient method for global sensitivity analysis based oncumulative distribution functions," *Environmental Modelling and Software*, vol. 67, pp. 1–11, May 2015, doi: 10.1016/j.envsoft.2015.01.004.

[25] A. Puy, S. Lo Piano, and A. Saltelli, "A sensitivity analysis of the PAWN sensitivity index," *Environmental Modelling and Software*, vol. 127, May 2020, doi: 10.1016/j.envsoft.2020.104679.

[26] I. Salehin, S. M. Noman, and M. M. Hasan, "Electricity energy dataset 'BanE-16': Analysis of peak energy demand with environmental variables for machine learning forecasting," *Data Brief*, vol. 52, Feb. 2024, doi: 10.1016/j.dib.2023.109967.

Model-Agnostic Explainability for Multi-Label Classification via Formal Concept Analysis

Hakim Radja[1], Yassine Djouadi[2], and Karim Tabia[3]
[1]Computer Science Department, BP 17, RP, Tizi-Ouzou, Mouloud Mammeri University of Tizi-Ouzou, Algeria
[2]RIIMA Laboratory, University of Algiers 1, Algeria
[3]CNRS, CRIL UMR 8188, Université d'Artois, Lens, France.
E-mail: hakim.radja@ummto.dz, y.djouadi@univ-alger.dz, tabia@cril.fr

Keywords: Explainable AI, formal concept analysis (FCA), multi-label classification

Received: July 1, 2025

Multi-label classification refers to a supervised learning problem where a single data instance can correspond to several labels simultaneously. While these models achieve high prediction accuracy, they share some limitations with single-label classifiers, particularly in terms of interpretability and explainability. In this work, we introduce a model-agnostic explainability method that enhances the interpretability of multi-label classification models using Formal Concept Analysis (FCA). Our approach aims to provide users with clearer insights into how these models make predictions, helping them better understand the rationale behind the decisions. Specifically, we address key questions such as: What is the smallest set of features needed for the multi-label classifier f to make a prediction? and Which features are relevant to a specific prediction? By answering these questions, our method seeks to improve user confidence in the model's decisions and foster a deeper understanding of multi-label classification. We introduce the key concepts of the Decisive Attribute Set (DAS) and the Significant Attribute Set (SSA). A DAS is the smallest set of features that can independently lead to a prediction, while an SSA includes all the features that influence the prediction of one or more labels. Additionally, we introduce a dedicated class-specific importance score that quantifies the role of each attribute, based on its frequency and specificity across formal concepts. Using these concepts, we generate clear, interpretable patterns in the form of rules, referred to as "DAS-rules", which provide straightforward explanations for individual predictions. Our method achieves high local fidelity, generates compact explanations, and successfully captures feature interactions, as demonstrated through extensive experiments on three benchmark datasets (Stack Overflow, Yelp, and TMC2007-500). These results demonstrate the novelty and practical effectiveness of our FCA-based framework for multi-label explainability.

Povzetek:

1 Introduction

In machine learning (ML), the primary goal of predictive models is often to achieve high accuracy. However, as model complexity increases especially with architectures like deep learning, a major drawback emerges: opacity. These models are frequently described as "black boxes," meaning their decision-making processes remain hidden from users. This lack of transparency presents a significant challenge, particularly in sensitive applications where understanding the reasoning behind a prediction is crucial. Explainability is essential because it allows users to interpret, verify, and comprehend how a model reaches its decisions. Without such insights, it becomes difficult to trust or validate the model's outcomes, leaving users uncertain about the factors driving specific decisions [9].
As noted by Golovin [10], any sufficiently complex system tends to behave like a black box when its ease of use surpasses the effort required to fully understand it. In fact, many of today's top-performing machine learning models, including deep neural networks, fall into this category [11, 18]. While these models exhibit impressive accuracy, their lack of interpretability limits their broader adoption, especially in high-stakes fields like healthcare, finance, and autonomous systems. As machine learning models become increasingly integral to decision-making in these sectors, users are left grappling with crucial questions: Can we trust these predictions? How can we meaningfully interpret the decisions made by these models? The black-box nature of these systems not only erodes trust but also poses a significant barrier to their widespread application. This challenge has led to the rise of Explainable AI (XAI) and interpretable machine learning, fields dedicated to improving model explainability and fostering user confidence [12].
Our work adds to the expanding body of research on explainability by focusing specifically on multi-label classification models, an area that has received comparatively less attention. Unlike single-label classifiers, which pre-

dict only one label per data sample, multi-label models can assign multiple labels to each instance, introducing an additional layer of complexity in their interpretation. To tackle this challenge, we leverage Formal Concept Analysis (FCA), a powerful mathematical framework that enables us to break down and explain the predictions made by multi-label classifiers. At the heart of our approach are two novel concepts aimed at addressing key questions regarding the role of features in predictions:

- **Decisive Attribute Set ($\mathcal{DAS}$)**: This represents the minimal subset of features required for the model to make a specific prediction. A $\mathcal{DAS}$ identifies the fewest features that, once known, are sufficient to prompt the classifier to predict a particular label or a sub-set of labels. Unlike traditional feature relevance methods, which rank features by importance but do not ensure their sufficiency for a specific prediction, this approach guarantees that the identified features are sufficient for that prediction.
- **Significant Attribute Set ($\mathcal{SSA}$)**: This includes all the features that appear in at least one Decisive Attribute Set. The intuition behind $\mathcal{SSA}$ is that, while not every feature is decisive on its own, these attributes collectively contribute meaningfully to the model's overall predictive process.
- Building on these two concepts, we introduce an *Attribute importance measure* that quantifies the contribution of each feature to the classifier's predictions. This measure reflects both the frequency and specificity with which a given attribute is involved in the sufficient conditions for predicting a particular label. By assigning a numerical score to each attribute with respect to each class, this metric complements the symbolic explanations provided by $\mathcal{DAS}$ and $\mathcal{SSA}$, offering a richer and more comprehensive understanding of the model's decision-making process.

In addition, we propose explanation rules derived from the $\mathcal{DAS}$ to explain individual predictions, similar to anchor rules [41]. These rules are complemented by numerical scores, which help identify and filter the most commonly occurring rules to explain a prediction. Our methodology is model-agnostic, meaning it can be applied to explain the predictions of any black-box model.

The paper introduces several key contributions in the context of explainable multi-label classification using Formal Concept Analysis (FCA). The main contributions of the paper are the following:

1. The paper defines two novel concepts Decisive Attribute Sets $\mathcal{DAS}$ and Significant Attribute Sets $\mathcal{SSA}$ and relates them to two key types of explanations in multi-label classification.
2. Model-Agnostic Framework for Explainability: The proposed method is designed to work with any black-box multi-label classification model.
3. Flattening and Use of FCA for Explanations: The paper develops a novel way to transform multi-label data into a formal context (a binary relation) that facilitates the application of FCA using off-the-shelf FCA tools. The transformation involves converting a three-dimensional (3D) relationship (instances, features, and labels) into a two-dimensional (2D) context.
4. Reduction Algorithm for Formal Concepts: A reduction algorithm is proposed to reduce the number of formal concepts derived during analysis. This optimization balances computational efficiency with the relevance of explanations, making the approach scalable.
5. Experimental Validation: The approach is evaluated using case studies and experiments on multi-label datasets like Stack Overflow, Yelp, and Aviation Safety Reporting System. These experiments validate the utility of the approach in extracting meaningful explanations and understanding classifier behaviors.

These contributions advance the field of explainable AI by providing a structured, mathematically grounded and model-agnostic method for enhancing explainability in multi-label classifiers.
The structure of the paper is as follows: Section 2 briefly presents multi-label classification and the methodologies used to address its challenges. Section 3 presents a detailed review of existing approaches in explainable AI. In Section 4, we discuss the fundamentals of formal concept analysis. Section 5 and Section 6 outlines our proposed methods and algorithms. Section 7 focuses on empirical evaluations and provides an in-depth analysis of the findings. Section 8 highlights the novelty of our approach and compares it with existing explainability techniques in the literature. Finally, Section 9 summarizes the key contributions of this study and provides concluding insights.

2 Multi-label classification

Multi-label classification assigns multiple labels to a single data instance, extending the traditional single-label classification. This approach is widely used in applications like text categorization [19], image classification [20], and bioinformatics [21]. Formally, a data instance $x = (a_1, a_2, \ldots, a_n)$, where each a_i is a binary attribute, is assigned a subset of labels $y \in 2^{\mathcal{C}}$, given a label set $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$.

Example 1. *Table 1 illustrates predictions from a multi-label classifier for 6 instances described by three attributes (a_1, a_2, a_3, also denoted as $\mathcal{P}$ in FCA notation). The label space $\mathcal{C}$ includes three labels (c_1, c_2, c_3). For instance, x_1 is predicted to belong to c_1 and c_3.*

Multi-label classification [57, 42, 34] has gained significant attention, leading to the development of various algorithms categorized into three types: problem transforma-

Table 1: Example of multi-label classifier predictions

$\mathcal{O}$	$\mathcal{P}$			$\mathcal{C}$		
	a_1	a_2	a_3	c_1	c_2	c_3
x_1	1	1	0	1	0	1
x_2	1	0	0	1	0	0
x_3	1	0	1	1	0	0
x_4	0	0	1	0	1	0
x_5	1	1	1	1	0	1
x_6	1	1	0	1	0	1

tion methods, algorithm adaptation methods, and ensemble methods.

1. **Problem Transformation Methods:** These methods convert multi-label classification problems into single-label equivalents. Prominent techniques include: Binary Relevance, Label Powerset and Label Ranking

2. **Algorithm Adaptation Methods:** These approaches modify existing algorithms to handle multi-label problems. Examples include multi-label decision trees, and adaptations of AdaBoost and neural networks for multi-label scenarios.

3. **Ensemble Methods:** These methods combine problem transformation and algorithm adaptation techniques to improve classification performance. An example is Random k-label (RAKEL), which constructs classifiers using random subsets of labels and combines their predictions through a voting mechanism.

Although machine learning algorithms appear powerful in terms of prediction accuracy, they have their own limitations. One of these is opacity or the lack of transparency, which by nature characterizes *black-box systems*. This means that the internal logic of these systems and the internal workings are hidden from the user, which is a serious disadvantage because it does not allow a human (expert or non-expert) to verify, interpret and understand the reasoning of the system and the way particular decisions are made [9]. In other words, any sufficiently complex system acts as a black-box when it becomes easier to run than to understand [10]. Many machine learning models, including the best-performing models in various fields, belong to this group of black-box systems [11] such as deep artificial neural networks [18].This has led to the emergence of the domain of interpretable and explainable machine learning.

3 Explainable AI

Can we always trust a high-performing model? Such models often make critical decisions, such as approving or rejecting mortgage applications or diagnosing severe medical conditions, where the stakes are high. Even when predictions are accurate, they demand clear explanations. A model that delivers both accurate predictions and understandable, relevant explanations holds significantly greater value. The growing use of machine learning in critical domains has generated significant interest in improving model explainability. Explainable ML aims to ensure transparency and interpretability [1], detect biases [2, 3], assess the safe and reliable operation of models [4, 5, 6], offer clear explanations for the underlying causal mechanisms [7], and provide tools for model improvement and debugging [8]. Existing explainability methods are typically categorized as follows [46]:

- **Intrinsic vs Post-hoc**: Intrinsic methods focus on constructing models that are inherently interpretable, whereas post-hoc methods generate explanations for pre-trained black-box models.

- **Local vs Global**: Local methods explain specific individual predictions, while global methods aim to provide insights into the overall behavior of the model.

- **Agnostic vs Specific**: Agnostic methods are model-independent and can be used across various models, while specific methods are designed exclusively for certain model architectures.

The primary focus in explainable ML research has been on how to provide explanations (for an overview, see [12]). Common techniques for explanation include ranking feature contributions [13], generating prototypes [14], and using decision trees [15] or decision rules [16]. Popular methods like LIME [17] and SHAP [18] are widely used to explain predictions. A crucial element of explainability in machine learning is assessing the importance of the features involved in making predictions. Saarela and Jauhiainen [27] provide an overview of explainability methods focused on feature importance. These methods range from providing basic insights to analyzing more complex models. Generally, simpler models, like linear classifiers, are easier to interpret [28, 29], whereas non-linear models pose greater challenges [30, 31]. The importance of attributes differs across models. For instance, an attribute that is critical for one model may be insignificant for another [32]. Turini et al. [12] categorize different approaches that link explainability to specific types of black-box models. Darwiche [33] introduces concepts such as "sufficient" and "necessary" reasons, which help define the minimal set of attribute values required for a prediction, categorizing attributes into these sets according to their importance. Comprehensive surveys further discuss the diversity of explainability techniques and their underlying assumptions [53, 54, 55].

Several recent studies have focused on explainability for multi-label classification. Ciaperoni et al. [49] propose a symbolic multi-label classification model that outputs a small set of interpretable IF-THEN rules, offering a balance between predictive performance and human readability. Ayad et al. [50] extend Shapley values to classifier

chains in a multi-label setting, capturing both direct and indirect feature contributions through their *Shapley Chains* method. Hemal and Saha [51] propose an explainable deep learning-based meta-classifier tailored for the multi-label classification of retinal diseases, integrating interpretation at the output level for each predicted label. Mylonas et al. [52] adapt local explanation techniques to random forest classifiers in a multi-label context, generating label-wise explanations for individual instances. However, these approaches are often tied to specific model architectures and deliver only particular types of explanations, which limits their general applicability across different classifiers and explanation needs.

The challenge in explaining black-box models lies in generating interpretable outputs that clarify a specific prediction without fully revealing the model's internal logic. This paper focuses on symbolic explanations, especially those that provide "sufficient reasons" to justify predictions. These explanations identify the factors that are necessary or sufficient for a particular prediction to occur. Our approach is grounded in formal concept analysis, presented in the next section.

R	hunt	2legs	fly	4legs
lion	X			X
duck		X	X	
eagle	X	X	X	
cow				X

Figure 1: Illustration of a formal context

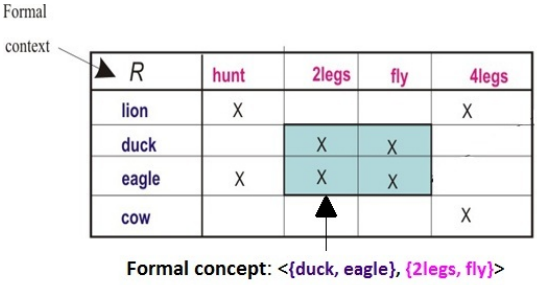


Figure 2: An example of a formal concept

4 Formal concept analysis

Formal Concept Analysis (FCA) [22], introduced by Wille in 1982, is an algebra-based approach that facilitates the discovery of potentially useful abstractions. It is applied in various fields:

- **In data mining**: To identify groups of products frequently purchased together, such as in consumer basket analysis.
- **In information retrieval**: To exploit an analogy between the object-property relationship and the document-term relationship for query refinement.
- **In knowledge representation**: To build clusters of knowledge known as *formal concepts*.

This method is based on a **formal context** $K = (O, P, R)$, where:

- O is the set of objects,
- P is the set of properties,
- $R \subseteq O \times P$ is a Boolean relation between certain objects in O and certain properties in P.

Formal concepts are induced using a set-theoretic operator called the **Galois derivation operator** (Δ):

- For a subset $X \subseteq O$, X^Δ represents the properties shared by all objects in X:

$$X^\Delta = \{p \in P \mid \forall o \in X,\ (o,p) \in R\}.$$

Example: $\{DUCK, EAGLE\}^\Delta = \{2legs, fly\}$.

- For a subset $Y \subseteq P$, Y^Δ represents the objects sharing all properties in Y:

$$Y^\Delta = \{o \in O \mid \forall p \in Y,\ (o,p) \in R\}.$$

Example: $\{2legs, fly\}^\Delta = \{DUCK, EAGLE\}$.

A **formal concept** is a pair $\langle X, Y\rangle$ such that $X^\Delta = Y$ and $Y^\Delta = X$.

- X is the *extent* of the concept (the maximal set of objects).
- Y is the *intent* of the concept (the maximal set of properties).

Example: $\langle\{DUCK, EAGLE\}, \{2legs, fly\}\rangle$ is a formal concept where $\{DUCK, EAGLE\}$ is the **extent**, and $\{2legs, fly\}$ is the **intent**.

The set of all formal concepts obtained is organized into a **complete lattice**, based on a partial order $\leq$, defined as follows:

$$\langle X_1, Y_1\rangle \leq \langle X_2, Y_2\rangle \iff X_1 \subseteq X_2 \text{ (equivalently, } Y_2 \subseteq Y_1)$$

Formal Concept Analysis (FCA) offers several advantages for data analysis. One of its key benefits is the visual representation of the concept lattice, which facilitates an intuitive exploration of the data and its underlying structure. In multi-label classification, FCA has been widely studied to enhance the performance of classification models. Research has demonstrated that FCA effectively addresses challenges specific to multi-label classification, such as handling noisy labels, reducing dimensionality, and learning from limited datasets. Li et al. [37] proposed a framework based on Formal Concept Analysis

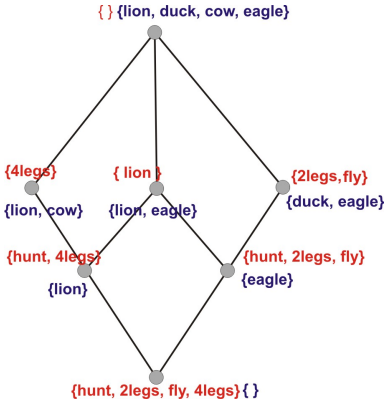


Figure 3: The concept lattice corresponding to the formal context of Figure 2

(FCA) for addressing multi-label classification, leveraging the conceptual structure of the data and achieving promising results in terms of precision and recall. Zhang et al. [38] proposed an innovative approach that combines FCA with deep learning techniques to capture highly discriminative features in multi-label data. Wang et al. [39] extended FCA to model label relationships, thereby improving the representation of complex class interactions. Additionally, Chen et al. [40] explored the use of fuzzy FCA to manage uncertainty and ambiguity in data annotations.

FCA is also used in the area of explainability of machine learning models. For example, in [43] the authors explore the application of Shapley values, derived from cooperative game theory, to assess the contribution of individual attributes in classification and clustering based on formal concepts. In [44] the authors explore how FCA can offer techniques to provide plausible explanations for model predictions, focusing on the importance of individual attributes in classification and clustering. Similarly, in [45], the authors propose a framework based on FCA to explain deep learning models that are considered as "black boxes". The authors apply FCA to analyze anomalies in data, which is then used to explain the results of machine learning models, focusing on binary classification tasks.

Building on this background, the next section outlines our approach to explaining multi-label classification using FCA.

5 FCA Representation for Multi-Label Classification Predictions

Explanations aim to clarify how specific input features influence a model's output, fostering trust and enabling users to validate its decisions.

In our approach, explanations are based on *decisive attributes* and *formal concepts*, which are used to identify the key features that drive a prediction.

Our approach to explaining the predictions of a multi-label classification model starts by associating a formal context with the model's prediction(s). Specifically, we use a standard formal context to represent instances, attributes, and labels. This raises an important question: why choose traditional Formal Concept Analysis (FCA) over triadic[1] FCA for this task? Classical FCA focuses on binary relationships between objects and attributes, making it algorithmically simpler and computationally more efficient than triadic FCA. In contrast, triadic FCA provides greater expressiveness by extending the representation to ternary relationships involving objects, attributes, and attribute values. However, this added expressiveness comes at the cost of increased computational complexity, even with smaller datasets [35] [36]. Classical FCA strikes a better balance between expressiveness and computational feasibility, particularly when applied to large-scale datasets.

We now provide an overview of our FCA-based explainability approach:

FCA-based explanation process: step-by-step overview

To provide a clearer understanding of the thinking behind our approach, we provide a concise five-step overview of how symbolic explanations are derived from a multi-label classifier using Formal Concept Analysis (FCA). This illustration is complemented by a simplified diagram (Figure 4) and a simplified example for illustration purposes.

Step 1: Classifier predictions We consider a dataset with three instances (x_1, x_2, x_3), three binary attributes (a_1, a_2, a_3), and two class labels (c_1, c_2). The predictions of a multi-label classifier f are as follows:

Object	a_1	a_2	a_3	c_1	c_2
x_1	1	0	1	1	0
x_2	0	1	0	1	1
x_3	1	1	1	1	1

So we have: $f(x_1) = \{c_1\}$, $f(x_2) = \{c_1, c_2\}$, $f(x_3) = \{c_1, c_2\}$.

Step 2: Ternary relation construction Each instance is represented as a set of active features and predicted classes. We build a ternary relation $R \subseteq O \times P \times C$, where a triple (x_i, a_j, c_k) indicates that attribute a_j is active in instance x_i and contributes to the prediction of class c_k.

- $x_1 \rightarrow (x_1, a_1, c_1), (x_1, a_3, c_1)$
- $x_2 \rightarrow (x_2, a_2, c_1), (x_2, a_2, c_2)$
- $x_3 \rightarrow$ all six possible triplets: $(x_3, a_1, c_1), (x_3, a_2, c_1), (x_3, a_3, c_1)$, $(x_3, a_1, c_2), (x_3, a_2, c_2), (x_3, a_3, c_2)$

[1] A triadic context consists of sets objects $\mathcal{O}$, set of attributes $\mathcal{P}$ and a set of conditions, $\mathcal{C}$. It extends binary formal contexts to represent ternary relations encoding under which condition c and an object o have attribute a.

To apply FCA, we flatten each attribute-class pair into a compound attribute (c_k, a_j). The resulting binary context is:

Object	$(c_1 - a_1)$	$(c_1 - a_2)$	$(c_1 - a_3)$	$(c_2 - a_1)$	$(c_2 - a_2)$	$(c_2 - a_3)$
x_1	1	0	1	0	0	0
x_2	0	1	0	0	1	0
x_3	1	1	1	1	1	1

This transformation is crucial since it allows us to encode the ternary relation into a standard two-dimensional binary context. In this way, we can directly leverage existing Formal Concept Analysis (FCA) tools, which are inherently designed to operate on binary contexts defined in two dimensions.

Step 3: Formal concept extraction We compute formal concepts $\langle X, Y \rangle$ from the binary context, where X is a set of objects and Y a set of shared compound attributes. For example:

$$\langle \{x_1, x_3\}, \{(c_1 - a_1), (c_1 - a_3)\} \rangle$$

indicates that both x_1 and x_3 share these two conditions for predicting class c_1.

Step 4: Reducing redundant concepts In real-world applications, the number of extracted formal concepts can be very large, often with significant redundancy. To enhance interpretability, we apply a reduction step that retains only the most relevant concepts, ensuring that the resulting explanations are concise and non-repetitive.

Step 5: Deriving explanations From the reduced set of formal concepts obtained, we derive symbolic explanations that describe how features contribute to the predicted labels. The different types of explanations and their computation are detailed in the following sections.

The following figure summarizes the different steps of our FCA-based approach.

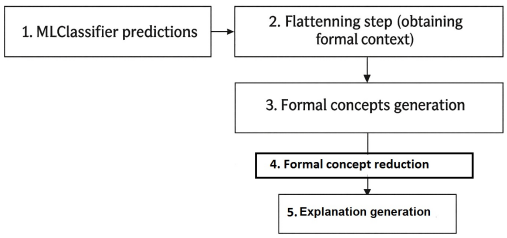


Figure 4: A step-by-step overview of the FCA-based explanation process

The classifier's predictions are transformed into a formal context using a flattening step. Formal concepts are then computed, from which symbolic explanations are derived.

From illustrative example to formal framework

In the following, we provide a formal and detailed description of the methodology outlined above. While the previous illustration was based on a simplified example, the next sections generalize the approach and define its components more rigorously. The five steps described earlier are synthesized into three main stages:

1. **Constructing a formal representation to encode classifier outputs**
   In multi-label classification, a classifier f assigns each data instance x, characterized by a feature vector $(a_1, a_2, \ldots, a_n)$, to a subset of labels y from the set $\mathcal{C} = \{c_1, c_2, \ldots, c_k\}$. This relationship can be represented as a 3D structure, including objects $\mathcal{O}$ (data instances), attributes $\mathcal{P}$, and labels $\mathcal{C}$, mathematically expressed as $\mathcal{R} \subseteq \mathcal{O} \times \mathcal{P} \times \mathcal{C}$. Converting this 3D structure into a 2D formal context is done in such a way that no information is lost.

2. **Deriving explanations through formal concepts**
   By transforming the 3D context into a 2D structure, formal concepts are generated, grouping objects with identical class predictions under the same conditions. To manage the potentially large number of concepts, a filtering process retains only the most relevant ones.

3. **Generating explanations**
   In this step, we present the various types of explanations derived from formal concepts. The *Decisive Attribute Set* ($\mathcal{DAS}$) is the smallest subset of features that allows the model f to predict a specific class for an instance, independently of the other attribute values. From the $\mathcal{DAS}$, we can derive other explanations, such as the *Significant Attributes Set* ($\mathcal{SSA}$), *necessary attributes set* ($\mathcal{NAS}$), and $\mathcal{DAS}$-rules.

The next sections detail the three stages: constructing the ternary relation, transforming it into a formal context, and extracting symbolic explanations.

5.1 Formal representation of classifier outputs

The process of generating explanations for multi-label classifier predictions begins with converting a multi-label dataset into a formal context. To start, let's define the key components of this transformation.

Let $\mathcal{MLD} = (\mathcal{O}, \mathcal{P}, \mathcal{C}, \mathcal{R})$ be a multi-label dataset, where $\mathcal{O}$ is a set of instances (or objects), $\mathcal{P}$ is a set of features (or attributes), $\mathcal{C}$ is a set of classes, and $\mathcal{R}$ is a ternary relation $\mathcal{R} \subseteq \mathcal{O} \times \mathcal{P} \times \mathcal{C}$, representing the relations between the instances, their features, and their associated class predictions. The goal is to transform this multi-label data, rep-

resented as a ternary relation $\mathcal{R} \subseteq \mathcal{O} \times \mathcal{P} \times \mathcal{C}$, into a binary relation $\mathcal{R}' \subseteq \mathcal{O} \times \mathcal{M}$, where $\mathcal{M}$ represents pairs of features and classes. This transformation, also called "flattening" of the data, is similar to what is done in [24]. As a result, a formal context $\mathcal{K}' = (\mathcal{O}, \mathcal{M}, \mathcal{R}')$ is created, where $\mathcal{O}$ represents the set of objects (instances), $\mathcal{M} = (\mathcal{C} \times \mathcal{P})$ is a set of pairs (c_k, a_j), where $a_j \in \mathcal{P}$ represents a feature, and $c_k \in \mathcal{C}$ represents a class, and $\mathcal{R}' \subseteq \mathcal{O} \times \mathcal{M}$ is a binary relation between objects and the set of pairs (c_k, a_j).
In simpler terms, this transformation associates each feature a_j with a class c_k to create a new set of combined attributes, allowing us to represent the original multi-dimensional data in a format compatible with Formal Concept Analysis (FCA). More specifically, for each instance $x_i \in \mathcal{O}$, the classifier f assigns predictions for one or more classes based on the feature values of x_i. The binary relation $\mathcal{R}'$ for an instance x_i and a pair (c_k, a_j) is defined as:

$$\mathcal{R}'(x_i, (a_j \times c_k)) = \begin{cases} 1 & \text{if } \mathcal{R}(x_i, a_j, c_k) = 1 \\ 0 & \text{otherwise} \end{cases}$$

By doing so, the ternary relationship between instances, features, and classes is flattened into a 2D relationship that can be directly processed using existing tools for FCA analysis.
Flattening the formal context is essential for applying FCA to generate explanations. By transforming the 3D data into a 2D binary context, we can identify formal concepts, which are the fundamental structures in FCA representing meaningful relationships between objects and attributes. In this context, a formal concept consists of a set of objects (instances) that share common attributes (feature–class pairs). Once the formal concepts are computed, they can be used to generate different types of explanations. Formal concepts naturally organize the data into clusters where instances display similar behavior, enabling us to derive explanations that are both interpretable and consistent with the classifier's predictions. In our approach, we use the Next Closure algorithm [26] to compute the formal concepts from the flattened formal context. This algorithm computes the closed itemsets based on the lexicographical order of attributes, making it efficient even for large datasets.

Example 2. *Table 2 provides an example of representing multi-label data as a formal context.*

5.2 Deriving explanations through formal concepts

To extract explanations from the formal context, the derivation operator is adapted to suit the context of multi-label classification. For a 2D formal context, the goal is to generate explanations for the classifier's predictions. Let $\mathcal{K}' = (\mathcal{O}, \mathcal{M}, \mathcal{R}')$ be the formal context, defined as follows:

- $\mathcal{O}$ represents the set of data instances,
- $\mathcal{M} = \mathcal{C} \times \mathcal{P}$ is the set of attribute-class pairs (c_k, a_j), where $a_j \in \mathcal{P}$ is an attribute, and $c_k \in \mathcal{C}$ is a class label,
- $(x_i, (c_k, a_j)) \in \mathcal{R}'$ indicates that the instance $x_i \in \mathcal{O}$ is associated with class c_k when the attribute a_j is present.

For any subset $X \subseteq \mathcal{O}$ and $Y \subseteq \mathcal{M}$, with $Y = \{(c_k, a_j) \mid a_j \in \mathcal{P}, c_k \in \mathcal{C}\}$, the Galois derivation operator $(.)^\Delta$ is defined as follows:

$$X^\Delta = \{(c_k, a_j) \in \mathcal{M} \mid X \subseteq \mathcal{R}'(c_k, a_j)\}$$

$$Y^\Delta = \{x \in \mathcal{O} \mid Y \subseteq \mathcal{R}'(x)\}$$

Here:

- X^Δ is the set of attribute-class pairs shared by all instances in X,
- Y^Δ is the set of instances that possess all attribute–class pairs in Y.

In this context, a formal concept is a pair $\langle X, Y \rangle$ such that $X \subseteq \mathcal{O}$, $Y \subseteq \mathcal{M}$, with the properties that $X^\Delta = Y$ and $Y^\Delta = X$. The set of all formal concepts is denoted by $\mathcal{L}(\mathcal{K}')$.

5.2.1 Example

Referring to the formal context from Table 2, the derived formal concepts are:

- $fc_1 : \langle \{x_1, x_2, x_3, x_4, x_5, x_6\}, \{\} \rangle$,
- $fc_2 : \langle \{x_4\}, \{(c_2, a_3)\} \rangle$,
- $fc_3 : \langle \{x_1, x_2, x_3, x_5, x_6\}, \{(c_1, a_1)\} \rangle$,
- $fc_4 : \langle \{x_1, x_5, x_6\}, \{(c_1, a_1), (c_1, a_2), (c_3, a_1), (c_3, a_2)\} \rangle$,
- $fc_5 : \langle \{x_3, x_5\}, \{(c_1, a_1), (c_1, a_3)\} \rangle$,
- $fc_6 : \langle \{x_5\}, \{(c_1, a_1), (c_1, a_2), (c_1, a_3), (c_3, a_1), (c_3, a_2), (c_3, a_3)\} \rangle$,
- $fc_7 : \langle \{\}, \{(c_1, a_1), (c_1, a_2), (c_1, a_3), (c_2, a_1), (c_2, a_2), (c_2, a_3), (c_3, a_1), (c_3, a_2), (c_3, a_3)\} \rangle$.

For instance, the formal concept fc_4 can be rewritten as:

$$fc_4 : \langle \{x_1, x_5, x_6\}, \{c_1, c_3\}, \{a_1, a_2\} \rangle$$

This means that classes c_1 and c_3 are predicted for the instances x_1, x_5, and x_6 when attributes a_1 and a_2 are present (their value is set to 1). Here, $\{x_1, x_5, x_6\}$ forms the extent, $\{c_1, c_3\}$ forms the intent, and $\{a_1, a_2\}$ represents the condition of the formal concept fc_4.
The flattened formal context, as previously constructed,

Table 2: An example of transforming 3D multi-label data into a 2D formal context

	$\mathcal{P}$			$\mathcal{C}$		
$\mathcal{O}$	a_1	a_2	a_3	c_1	c_2	c_3
x_1	1	1	0	1	0	1
x_2	1	0	0	1	0	0
x_3	1	0	1	1	0	0
x_4	0	0	1	0	1	0
x_5	1	1	1	1	0	1
x_6	1	1	0	1	0	1

	$\mathcal{M} = \mathcal{C} \times \mathcal{P}$								
$\mathcal{O}$	c_1-a_1	c_1-a_2	c_1-a_3	c_2-a_1	c_2-a_2	c_2-a_3	c_3-a_1	c_3-a_2	c_3-a_3
x_1	1	1	0	0	0	0	1	1	0
x_2	1	0	0	0	0	0	0	0	0
x_3	1	0	1	0	0	0	0	0	0
x_4	0	0	0	0	0	1	0	0	0
x_5	1	1	1	0	0	0	1	1	1
x_6	1	1	0	0	0	0	1	1	0

plays a crucial role in generating these explanations. Each formal concept derived from the context captures information about a set of instances, their predicted classes, and the conditions (attributes) under which these predictions are made. As a result, the formal concept naturally offers an explanation by identifying the attributes that influence the classifier's predictions and grouping instances with similar characteristics and outcomes. Since the formal concepts are derived from the flattened context, they serve as direct representations of explanations. Each concept connects data instances, class predictions, and the relevant attributes, making the Galois lattice of concepts an ideal framework for organizing and interpreting the classifier's decisions.
By proceeding in this way with the set of all formal concepts obtained in the previous example, we obtain the following set of formal concepts:

- $fc_1 : \langle\{x_1, x_2, x_3, x_4, x_5, x_6\}, \{\}, \{\}\rangle$,
- $fc_2 : \langle\{x_4\}, \{c_2\}, \{a_3\}\rangle$,
- $fc_3 : \langle\{x_1, x_2, x_3, x_5, x_6\}, \{c_1\}, \{a_1\}\rangle$,
- $fc_4 : \langle\{x_1, x_5, x_6\}, \{c_1, c_3\}, \{a_1, a_2\}\rangle$,
- $fc_5 : \langle\{x_3, x_5\}, \{c_1\}, \{a_1, a_3\}\rangle$,
- $fc_6 : \langle\{x_5\}, \{c_1, c_3\}, \{a_1, a_2, a_3\}\rangle$,
- fc_7: $\langle\{\}, \{c_1, c_2, c_3\}, \{a_1, a_2, a_3\}\rangle$.

5.2.2 Minimizing formal concepts

The large number of formal concepts can affect their relevance for explanations. Algorithm 1 reduces this number by taking into account attributes, instances, and classes. For instance, in Example 5.2.1, the intent $\{c_1\}$ is found in multiple concepts.

- In fc_3: the class c_1 is predicted for the objects x_1, x_2, x_3, x_4, x_5, and x_6 when the attribute a_1 is present.
- In fc_5: the class c_1 is predicted for the objects x_3 and x_5 when the attributes a_1 and a_3 are present.

It is evident that the attribute a_1 alone suffices to classify objects x_3 and x_5 in class c_1, as $\{x_3, x_5\} \subseteq \{x_1, x_2, x_3, x_4, x_5, x_6\}$ and $\{a_1\} \subseteq \{a_1, a_3\}$. Hence, a_3 is not essential for predicting c_1 for the objects $\{x_3, x_5\}$. From an explainability perspective, a_3 is irrelevant, and retaining only the formal concept fc_3 is adequate. Leveraging this observation, Algorithm 1 aims to minimize the number of formal concepts: Let $\mathcal{L}(K') = \{fc_1, fc_2, \ldots, fc_n\}$ denote the set of all formal concepts, $\mathcal{I}nt(fc_i)$ refer to the intent of the formal concept fc_i, $\mathcal{E}xt(fc_i)$ indicate the extent of the formal concept fc_i, and $\mathcal{C}ond(fc_i)$ represent its condition. The objective is to reduce the number of formal concepts by removing redundant ones based on their intents, extents, and conditions. The main steps of this algorithm are: **Input:** A set of formal concepts $\mathcal{L}(K') = \{fc_1, fc_2, \ldots, fc_n\}$.
Output: A reduced set of formal concepts $\mathcal{L}'(K')$.
Steps:

1. Initialize an empty set for the reduced formal concepts, $\mathcal{L}'(K') = \emptyset$.
2. For each pair of formal concepts fc_i and fc_j:
   - If their intents are identical $\mathcal{I}nt(fc_i) = \mathcal{I}nt(fc_j)$:
   - Check if the extent of fc_i is a subset of the extent of fc_j: $\mathcal{E}xt(fc_i) \subseteq \mathcal{E}xt(fc_j)$.
   - If true, ensure that the condition of fc_i is a super-set of the condition of fc_j: $\mathcal{C}ond(fc_i) \supseteq \mathcal{C}ond(fc_j)$.

– If all conditions are met, add fc_i to the reduced set, $\mathcal{L}'(K')$.

Algorithm 1 Formal Concepts Reduction

Input: $\mathcal{L}(K') = \{fc_1, \ldots, fc_n\}$
Output: $\mathcal{L}'(K')$ ▷ Reduced set of formal concepts

```
L'(K') ←
for i ← 1 to n − 1 do                    ▷ n is the size of L(K')
    for j ← i + 1 to n do
        if Int(fc_i) = Int(fc_j) then
            if Ext(fc_i) ⊆ Ext(fc_j) then
                if Cond(fc_i) ⊇ Cond(fc_j) then
                    L'(K') ← L'(K') ∪ {fc_i}    ▷ Add fc_i to L'(K')
```

5.2.3 Example (example 5.2.1 continued)

By applying the reduction algorithm to the set $\mathcal{L}(K')$ derived in Example 5.2.1, the reduced set of formal concepts is obtained as $\mathcal{L}'(K') = \{fc_1, fc_2, fc_3, fc_4, fc_7\}$.

The time complexity of the algorithm is $O(n^2)$, where n is the size of the input set $\mathcal{L}(K')$. This quadratic complexity is manageable for moderately-sized datasets, allowing for efficient reduction while ensuring the exactness of explanations.

So far, we have demonstrated the construction of a formal context for explanations and the reduction of formal concepts. The next section focuses on generating explanations from the resulting formal concepts.

6 Generating explanations

Our method for explaining multi-label classification operates in a model-agnostic manner and delivers symbolic explanations in the form of sufficient reasons.

In this section, we further elaborate on the example presented in Table 2. As detailed in the previous section, the reduced set of formal concepts, $\mathcal{L}'(K')$, obtained is:

- $fc_1 : \langle\{x_1, x_2, x_3, x_4, x_5, x_6\}, \{\}, \{\}\rangle$,
- $fc_2 : \langle\{x_4\}, \{c_2\}, \{a_3\}\rangle$,
- $fc_3 : \langle\{x_1, x_2, x_3, x_5, x_6\}, \{c_1\}, \{a_1\}\rangle$,
- $fc_4 : \langle\{x_1, x_5, x_6\}, \{c_1, c_3\}, \{a_1, a_2\}\rangle$,
- fc_7: $\langle\{\}, \{c_1, c_2, c_3\}, \{a_1, a_2, a_3\}\rangle$.

fc_1 and fc_7 can be removed from $\mathcal{L}'(K')$ as they do not provide any useful information, resulting in the new set of formal concepts $\mathcal{L}'(K')$:

- $fc_2 : \langle\{x_4\}, \{c_2\}, \{a_3\}\rangle$,
- $fc_3 : \langle\{x_1, x_2, x_3, x_5, x_6\}, \{c_1\}, \{a_1\}\rangle$,
- $fc_4 : \langle\{x_1, x_5, x_6\}, \{c_1, c_3\}, \{a_1, a_2\}\rangle$,

For any data instance x with prediction $f(x)$, our approach provides symbolic explanations in the form of *sufficient reasons*. These explanations consist of subsets of features from x that justify its prediction $y = f(x)$.

6.1 Sufficient reason explanations

In the context of explainable AI, we aim to find explanations that are concise, as overly long explanations can become difficult to interpret. Therefore, the goal is to identify the smallest possible explanation that still effectively justifies a prediction. In our case, this means identifying the smallest subset of attributes that, once determined, triggers the prediction regardless of the values of the remaining attributes. A sufficient reason consists of a sub-set of features that, when present (namely, their values set to 1), provide a sufficient explanation for classifying a data instance into a specific class. For instance, in a medical diagnosis model, features such as Fever and cough may provide a sufficient reason for classifying a patient as having a respiratory infection. In our approach, sufficient reason explanations are represented by the concept of the *Decisive Attribute Set*.

Definition 1. (*Decisive Attribute Set*)
The smallest (in the sense of set inclusion) set of attributes enabling the classification of a data instance into a given class is referred to as the '*Decisive Attribute Set ($\mathcal{DAS}$)*.'

In FCA terms, a sufficient reason corresponds to the concept of a *Decisive Attribute Set ($\mathcal{DAS}$)*. Formally, a $\mathcal{DAS}$ is defined as the smallest subset of features $\{a_j \in \mathcal{P}\}$ such that a classifier f predicts $y \subseteq \mathcal{C}$ for an instance $x \in \mathcal{O}$, independently of the values of the remaining attributes. This minimality ensures that the prediction remains valid regardless of variations in the unused attributes.

Let $x = (a_1, a_2, \ldots, a_n)$ represent the data instance to classify, and let $c_k \in y \subseteq \mathcal{C}$ be a class included in the multi-label prediction $y = f(x)$, where f is the multi-label classifier to be explained.

We recall that in the context of multi-label classification, a formal concept is defined as a triple:

$$\langle \text{Extent, Intent, Condition} \rangle$$

where:

- the **Extent** (**Ext**(fc_i)) is the set of instances for which the concept holds,
- the **Intent** (**Int**(fc_i)) is the set of predicted class labels for those instances,
- the **Condition** (**Cond**(fc_i)) is the set of input attributes that characterize the instances and explain the predicted labels.

For example, a concept of the form $\langle\{x_1, x_5, x_6\}, \{c_1, c_3\}, \{a_1, a_2\}\rangle$ expresses that instances x_1, x_5, and x_6 share the feature condition $\{a_1, a_2\}$,

which leads the classifier to assign them the labels c_1 and c_3.

To better understand the relationship between sufficient reasons and decisive attributes, we define $\mathcal{B}(c_k)$ as the set of all formal concepts whose intents contain the class c_k. Let $\mathcal{L}'(K') = \{fc_1, fc_2, \ldots\}$ represent the reduced set of formal concepts derived using Algorithm 1. Specifically, $\mathcal{B}(c_k) = \{fc_i \in \mathcal{L}'(K') \mid c_k \in \mathcal{I}nt(fc_i)\}$.

The $\mathcal{DAS}$ that enables the model f to predict the class c_k is further defined as follows:

- $\mathcal{B}(c_k)$: the set of all formal concepts whose intents include the class c_k,
- $\mathcal{E}xt(fc_i) = \{x_i \in \mathcal{O} \mid fc_i \in \mathcal{B}(c_k)\}$: the extent of the formal concept containing c_k in its intent,
- $\mathcal{C}ond(fc_i) = \{a_i \in \mathcal{P} \mid fc_i \in \mathcal{B}(c_k)\}$: the condition of the formal concept containing c_k in its intent,
- $\mathcal{D}iff(c_k) = \bigcup_{i=1}^{n} \mathcal{E}xt(fc_i) \setminus \bigcap_{i=1}^{n} \mathcal{E}xt(fc_i)$: the set difference between the union and the intersection of the extents of formal concepts having c_k in their intents.

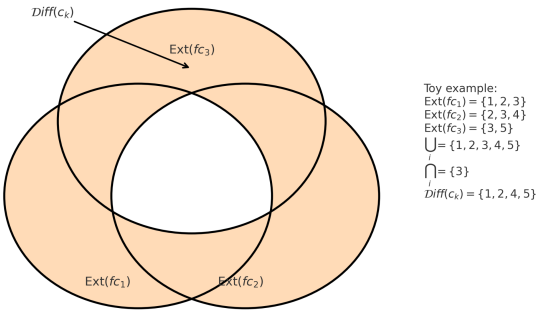


Figure 5: Illustration of $\mathcal{D}iff(c_k)$ with the distinctive instances (colored area) that appear in at least one but not all extents for c_k

Intuition: $\mathcal{D}iff(c_k)$ contains the *distinctive instances* that appear in at least one concept for c_k but not in all of them. This operation removes the instances that appear in all concepts, as they do not provide any variation to explain the prediction. Only instances in $\mathcal{D}iff(c_k)$ help differentiate between different decision contexts, and therefore, they are significant for interpretation. By focusing on these distinctive instances, we capture the diversity of scenarios leading to c_k, making the explanation more specific and informative.

The Decisive Attribute Set ($\mathcal{DAS}$) for the class c_k is computed as:

$$\mathcal{DAS}_j(c_k) = \{\bigcup \mathcal{C}ond(fc_i) | x_j \in \mathcal{E}xt(fc_i) \text{ and } x_j \in \mathcal{D}iff(c_k)\}.$$

Algorithm 2 Compute $\mathcal{DAS}(c_k)$

Input: $\mathcal{B}(c_k)$
Output: $\mathcal{DAS}(c_k)$

$\mathcal{DAS}(c_k) = \emptyset$
if $|\mathcal{B}(c_k)|$=1 **then**
  $\mathcal{DAS}(c_k) = \mathcal{C}ond(fc_k)$
else
  for $i \leftarrow 1, |\mathcal{B}(c_k)|$ **do**
    $\mathcal{D}iff(c_k) = \bigcup \mathcal{E}xt(fc_i) \setminus \bigcap \mathcal{E}xt(fc_i)$
    for each $x \in \mathcal{D}iff(c_k)$ **do**
      for $i = 1 \leftarrow 1, n$ **do**
        if $x \in \mathcal{E}xt(fc_i)$ **then**
          $\mathcal{DAS}_x(c_k) = \bigcup \mathcal{C}ond(fc_i)$
        if $\mathcal{DAS}_x(c_k) \notin \mathcal{DAS}(c_k)$ **then**
          $\mathcal{DAS}(c_k) = \mathcal{DAS}(c_k) \cup \mathcal{DAS}_x(c_k)$
  ▷ Add a subset to $\mathcal{DAS}(c_k)$

Below an algorithm to summarize the procedure for computing the Decisive Attribute Set $\mathcal{DAS}$:

The algorithm ensures completeness through its exhaustive iterative strategy across all relevant formal concepts. At each iteration, the algorithm systematically considers all pertinent formal concepts and evaluates each instance associated with these concepts. This methodical approach guarantees coverage of every conceivable scenario within the conceptual space, affirming the algorithm's completeness in addressing a specific problem.

After we have obtained the set of reduced formal concepts $\mathcal{L}'(K')$, we use it to calculate the Decisive Attribute sets for a given class.

To derive the Decisive Attribute Sets $\mathcal{DAS}(c_k)$ for the class c_k, we follow these steps:

1. **Identify Relevant Formal Concepts:**
   Compute the set $\mathcal{B}(c_k)$ consisting of all formal concepts in $\mathcal{L}'(K')$ whose intents include the class c_k.

2. **Handle Single Formal Concept Case:**
   If $|\mathcal{B}(c_k)| = 1$, this implies that only one formal concept includes c_k in its intent.
   In this case, the Decisive Attribute Set for predicting c_k is simply the condition of this unique formal concept.
   Recall that a formal concept in our approach is represented as $fc : \langle\{extent\}, \{intent\}, \{condition\}\rangle$, meaning:
   - The classes in $\{intent\}$ are predicted for the instances in $\{extent\}$,
   - provided the attributes in $\{condition\}$ are present.

3. **Handle Multiple Formal Concepts Case:**
   If c_k appears in multiple formal concepts, we analyze the representative instances ($\mathcal{D}iff(c_k)$) of each formal concept.

The representative instances are obtained by calculating the set difference between the union and the intersection of the extents of the formal concepts in $\mathcal{B}(c_k)$ For each instance x in $\mathcal{D}iff(c_k)$, we keep only the formal concepts that have the instance x in their extents. Then, we calculate the $\mathcal{D}AS_x(c_k)$ by making the union of these formal concepts intents.

Intuitively, a sufficient reason for a class c_k is a minimal set of attributes R that fully predicts and explains the predicted class, meaning that the presence of R implies the prediction of c_k, with no superfluous attributes in R. Similarly, a DAS is a set of attributes sufficient to predict c_k, containing only the attributes necessary for that prediction. The minimality of R and DAS ensures that these sets contain only the essential attributes, without redundancy. Thus a DAS is a sufficient reason because it includes the attributes necessary and sufficient for the prediction, and a sufficient reason is a DAS because it is both minimal and complete in justifying the predicted class. This equivalence directly drives the design of Algorithm 2. By systematically identifying the DAS for each class, the algorithm computes the minimal sufficient reasons for the predictions. Its completeness and correctness, as discussed, further ensures that no essential explanatory element is overlooked. Algorithm 2 ensures both completeness and correctness in the computation of Decisive Attribute Sets ($\mathcal{D}AS(c_k)$). Completeness is achieved because the algorithm examines all relevant formal concepts ($\mathcal{B}(c_k)$) for a given class c_k and identifies the minimal set of attributes that are necessary and sufficient to predict that class. Using representative instances ($\mathcal{D}iff(c_k)$) ensures that every relevant case is considered. Correctness arises from the methodical application of formal concept properties: the extents and intents of these concepts are utilized to guarantee that the pertinent attribute sets meet the criteria of minimality and predictiveness.

It is important to highlight that a Decisive Attribute Set ($\mathcal{D}AS$) enabling the prediction of a class c_k is not necessarily unique. Indeed, multiple Decisive Attribute Sets can achieve the same prediction. The collection $\mathcal{D}AS(c_k)$ includes all subsets $\mathcal{D}AS_x(c_k)$, where $x \in \mathcal{D}iff(c_k)$.

Continuing with our running example, suppose we aim to identify all Decisive Attribute Sets that allow f to predict the class $c_k = \{C_2\}$. The first step is to identify the formal concepts whose intent includes the class c_2 in $\mathcal{L}'(K')$:

$\mathcal{B}(\{c_2\}) = \{fc_2\}$. $|\mathcal{B}(c_2)| = 1$, this implies that only one formal concept includes c_2 in its intent.
In this case, the Decisive Attribute Set for predicting c_2 is simply the condition of this unique formal concept, namely:
$\mathcal{D}AS(c_2) = \{a_3\}$
Similarly, for the class c_3, it appears in a single formal concept, $\mathcal{B}(\{c_3\}) = \{fc_4\}$, and thus:
The Decisive Attribute Set for predicting c_3 is:
$\mathcal{D}AS(c_3) = \{a_1, a_2\}$
For the class $c_k = c_1$, it appears in two formal concepts, which are fc_3 and fc_4. Let us compute $\mathcal{D}iff(c_1)$:
$\mathcal{D}iff(c_1) = \bigcup \mathcal{E}xt(fc_i) \backslash \bigcap \mathcal{E}xt(fc_i), \quad \forall fc_i \in \mathcal{B}(\{c_1\})$,
resulting in $\mathcal{D}iff(c_1) = \{x_2, x_3\}$.

For each $x \in \mathcal{D}iff(c_1)$, compute $\mathcal{D}AS_x(c_1)$.
For the instance x_2, $\mathcal{D}AS_{x_2}(c_1) = \{a_1\}$
For the instance x_3, $\mathcal{D}AS_{x_3}(c_1) = \{a_1\}$
In this case, $\mathcal{D}AS_{x_2}(c_1) = \mathcal{D}AS_{x_3}(c_1)$.
Thus, the complete set of Decisive Attribute Sets for predicting $y = \{c_1\}$ is:
$\mathcal{D}AS(c_1) = \mathcal{D}AS_{x_2,x_3}(c_1) = \{a_1\}$

6.2 Significant attribute set

Attributes that are present in at least one Decisive Attribute Set are considered crucial, as they play a key role in determining the classification of an instance into a specific class. Each $\mathcal{D}AS$ corresponds to the minimal set of attributes that is required for accurate classification, providing a clear justification for the classification of a data instance.

The Significant Attribute Set for a class c_k (denoted as $\mathcal{S}SA(c_k)$) comprises all attributes that appear in at least one Decisive Attribute Set for that class, which effectively forms the union of these sets. For a set of classes y, the Significant Attribute set is defined as the union of the significant attribute sets of each class in y:

$$\mathcal{S}SA(y) = \bigcup_{c_k \in y} \mathcal{S}SA(c_k).$$

The Significant Attribute Set for $y = \{c_1\}$ is the union of all Decisive Attribute Sets for the class c_1. Thus,
$\mathcal{S}SA(c_1) = \{a_1\}$. In the same way, the Significant Attribute Set for $y = c_3$ is the union of all Decisive Attribute Sets for the class c_3. Thus,
$\mathcal{S}SA(c_3) = \{a_1, a_2\}$.

For both classes c_1 and c_3, the combined Significant Attribute Set is:
$\mathcal{S}SA(c_1, c_3) = \mathcal{S}SA(c_1) \cup \mathcal{S}SA(c_3)$:
$\mathcal{S}SA(c_1, c_3) = \{a_1, a_2\}$.

6.3 Symbolic explanations and feature importance in multi-label predictions

Recall that our approach for explaining multi-label classification is model-agnostic and provides both *symbolic* explanations and feature importances.

In this section, we illustrate, through a real example from the medical field, how the previously defined $\mathcal{D}AS$ (Decisive Attribute Sets) and $\mathcal{S}SA$ (Significant Attribute Set) can be computed and interpreted in practice. This example deals with diagnosing some respiratory diseases (labels) given the symptoms observed in patients (attributes). For the sake of simplicity, we consider only four diseases: Asthma (d_1), Bronchiolitis (d_2), COPD (d_3), and COVID (d_4).

These diseases generally manifest through the following symptoms: Dry cough (s_1), Loose cough (s_2), Shortness of breath (s_3), Wheezing (s_4), Fever (s_5), Headaches (s_6), Loss of taste (s_7), Curvatures (s_8), and Dyspnoea (s_9).

Table 3: Multi-label predictions generated by the black-box model f for the neighborhood of the target instance x under explanation.

	Symptoms									Diseases			
$\mathcal{O}$	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	d_1	d_2	d_3	d_4
1	1	0	1	1	1	0	0	0	0	1	1	0	0
2	0	1	0	1	1	0	0	1	1	1	0	1	0
3	1	0	1	0	1	1	1	1	0	1	0	0	1
4	1	0	0	0	1	1	1	1	0	0	0	0	1
5	0	1	1	0	0	0	0	1	1	0	0	1	0
6	1	0	1	0	1	1	1	1	0	1	1	0	1
7	0	0	0	0	1	1	1	1	0	0	0	0	1
8	1	0	1	1	1	1	0	1	0	1	1	0	1
9	1	0	1	0	0	0	0	1	0	0	1	0	0
10	1	0	1	0	1	1	1	1	0	0	1	0	1
11	0	1	1	1	1	0	0	0	1	1	1	1	0
12	0	1	0	0	1	0	0	0	1	0	0	1	0
13	1	0	1	0	1	1	1	1	0	0	0	0	1
14	1	0	1	1	0	0	0	0	0	1	0	0	0

After transforming these multi-label data into a formal context, we obtain the following reduced set of formal concepts $\mathcal{L}'(K')$:

fc_1: $\langle\{11\}$, {Asthma, Bronchiolitis, COPD},
{fever, Dyspnoa, Shortness of breath, Wheezing,
Loose cough}$\rangle$
fc_2: $\langle\{2, 11\}$, {Asthma, COPD},
{Fever, Loose cough, Dyspnoa, Wheezing}$\rangle$
fc_3: $\langle\{8, 6\}$, {Asthma, Bronchiolitis, Covid},
{Fever, Headache, Shortness of breath, Dry cough,
Curvature}$\rangle$
fc_4: $\langle\{8, 3, 6\}$, {Asthma, Covid},
{Fever, Headache, Shortness of breath, Dry cough,
Curvature}$\rangle$
fc_5: $\langle\{8, 10, 6\}$, {Bronchiolitis, Covid},
{fever, Headache, Shortness of breath, Dry cough,
Curvature}$\rangle$
fc_6: $\langle\{8, 1, 11\}$, {Asthma, Bronchiolitis},
{Fever, Shortness of breath}$\rangle$
fc_7: $\langle\{2, 11, 12, 5\}$, {COPD},
{Loose cough, Dyspnoa}$\rangle$
fc_8: $\langle\{1, 2, 8, 11, 14\}$, {Asthma},
{wheezing}$\rangle$
fc_9: $\langle\{1, 2, 3, 6, 8, 11\}$, {Asthma},
{Fever}$\rangle$
fc_{10}: $\langle\{1, 3, 6, 8, 11, 14\}$, {Asthma},
{Shortness of breath}$\rangle$
fc_{11}: $\langle\{1, 6, 8, 9, 10, 11\}$, {Bronchiolitis},
{Shortness of breath}$\rangle$
fc_{12}: $\langle\{3, 4, 6, 7, 8, 10, 13\}$, {Covid},
{Fever, Headache, Curvature}$\rangle$

Given a data instance x and the prediction $f(x)$, we are first interested in symbolic explanations that are *sufficient reasons*. Broadly speaking, they refer to the subset of features in x that are sufficient to predict $y = f(x)$.

Decisive Attribute set ($\mathcal{DAS}$): Assume we want to compute the set of all Decisive attribute Sets that enable f to predict the class $c_k = \{\text{COVID}\}$ (also denoted as d_4). First, we select the formal concepts having in their intent the class:

$$\mathcal{B}(\{\text{COVID}\}) = \{fc_3, fc_4, fc_5, fc_{12}\}$$

Next, compute the difference between extents:

$$\mathcal{D}iff(\text{COVID}) = \bigcup \mathcal{E}xt(fc_i) \setminus \bigcap \mathcal{E}xt(fc_i)$$
$$= \{3, 4, 7, 10, 13\}$$

For each instance $x \in \mathcal{D}iff(\text{COVID})$, we compute its decisive attributes $\mathcal{DAS}_x(\text{COVID})$:

- For instances $\{10, 3\}$: $\mathcal{DAS}_{10,3}(\text{COVID}) = \{\text{Fever, Headache, Shortness of breath, Dry cough, Curvature}\}$
- For instances $\{4, 13, 7\}$: $\mathcal{DAS}_{4,13,7}(\text{COVID}) = \{\text{Fever, Headache, Curvature}\}$

The complete set of decisive attributes for the class COVID is:

$$\mathcal{DAS}(\text{COVID}) = \{\mathcal{DAS}_{4,13,7}(\text{COVID}), \mathcal{DAS}_{10,3}(\text{COVID})\}$$

These sets represent the minimal symptom subsets sufficient for model f to predict the class $y = \{\text{COVID}\}$ for the corresponding instances.

Significant Attribute Set:
The Significant Attribute Set for $y = \{\text{COVID}\}$ is defined as the union of all Decisive Attribute Sets for that class:

$$\mathcal{S}sa(\text{COVID}) = \{\text{Fever, Headache, Curvature, Dry cough, Shortness of breath}\}$$

This set contains all symptoms appearing in at least one Decisive Attribute Set associated with predicting the class *COVID*. In other words, each of these symptoms played a role in at least one instance where the model f predicted *COVID*. The set $\mathcal{SSA}(\text{COVID})$ therefore provides a global, symbolic explanation of the disease: it captures the complete pool of symptoms that have been sufficient, in different combinations, to trigger the model's decision for *COVID*.

6.3.1 Introducing attribute importance

While symbolic explanations such as $\mathcal{DAS}$ and $\mathcal{SSA}$ provide insight into which attributes are sufficient for predicting specific labels, they do not directly quantify the relative importance of each attribute in the overall decision-making process.

In practical applications, particularly in sensitive domains such as healthcare, it is crucial not only to identify the features that justify a prediction, but also to assess how strongly each feature contributes to it. This motivates the need for a *measure of feature importance*, which can help users grasp the influence of each individual attribute in the model's decisions.

In the following, we build upon the extracted formal concepts to propose an interpretable way to quantify attribute importance in the multi-label setting.

To stay within our symbolic and concept-based explanation framework, we propose a simple and transparent method to estimate how important each attribute is in predicting a given class. This idea is based on the formal concepts already extracted from the data.

The key intuition is simple: an attribute is considered important if it appears often in the different sets of features

that are sufficient to justify the prediction of a class. The more frequently an attribute appears in such explanations, the greater its influence on the model's decision for that class.

Next, we define this measure in a precise way and explain how it can be computed and interpreted using the formal concepts.

Attribute Importance Score.

To address this need, we propose a simple and interpretable measure of feature importance that is consistent with the concept-based explanations introduced earlier. Rather than relying on simple frequency counts or cumulative support, our measure aggregates the set of unique instances covered by all formal concepts that include both the attribute and the class. This ensures that each instance is counted only once, even if it appears in multiple overlapping concepts.

Let F be the set of formal concepts, a an attribute, and c a class label. We define:

- $\mathcal{F}_{a,c}$: the subset of concepts in F where both attribute a and class c appear (in the condition and the intent respectively),
- $\mathcal{F}_c$: the subset of concepts in F where class c appears in the intent (regardless of attribute a),
- $\text{ext}(\mathcal{F})$: the union of the extents of all concepts in $\mathcal{F}$, i.e., all distinct instances involved.

The importance score of attribute a for class c is then defined as:

$$\text{Imp}(a, c) = \frac{|\text{ext}(\mathcal{F}_{a,c})|}{|\text{ext}(\mathcal{F}_c)|}$$

This ratio reflects how widely attribute a contributes to justifying the class c across all its supporting concepts. A higher score indicates that a is involved in explaining a larger proportion of the instances predicted to belong to class c.

Illustrative example.

We illustrate the computation of the importance score for attributes with respect to the class *COPD*, using the formal concepts introduced earlier.

Let $c = \text{COPD}$, and let $\mathcal{F}$ denote the set of formal concepts extracted from the dataset. A formal concept is defined as a triple ⟨extent, intent, condition⟩, where the extent is the set of instances, the intent is the set of predicted class labels, and the condition is the set of attributes shared by these instances. For instance, the following concepts include the class COPD in their intent: Thus, we obtain:

$$\text{ext}(F_c) = \{2, 5, 11, 12\} \quad \text{and} \quad |\text{ext}(F_c)| = 4$$

Now, we compute the importance for several attributes:

- **Loose cough:** Appears in all three concepts ⇒ $\text{ext}(F_{\text{Loose cough},c}) = \{2, 5, 11, 12\} \Rightarrow \text{Imp} = \frac{4}{4} = 1.0$
- **Dyspnoea:** Appears in all three concepts ⇒ $\text{ext}(F_{\text{Dyspnoea},c}) = \{2, 5, 11, 12\} \Rightarrow \text{Imp} = \frac{4}{4} = 1.0$
- **Wheezing:** Appears in fc_1 and fc_2: $\text{ext}(F_{\text{Wheezing},c}) = \{2, 11\} \Rightarrow \text{Imp} = \frac{2}{4} = 0.5$
- **Fever:** Appears in fc_1 and fc_2: $\text{ext}(F_{\text{Fever},c}) = \{2, 11\} \Rightarrow \text{Imp} = \frac{2}{4} = 0.5$
- **Shortness of breath:** Appears only in fc_1: $\text{ext}(F_{\text{Shortness of breath},c}) = \{11\} \Rightarrow \text{Imp} = \frac{1}{4} = 0.25$

Features such as *Loose cough* and *Dyspnoea* exhibit the highest importance scores, since they appear in all formal concepts related to the class *COPD*. Consequently, whenever these attributes are present in the explanations, they are the most decisive in the prediction of *COPD*. In contrast, attributes like *Fever* and *Shortness of Breath* occur only in a subset of these concepts, while the other attributes do not contribute at all to the prediction of this class.

Feature importance visualization

To provide a clearer understanding of the influence of each feature on the model's predictions, we visualize the importance scores computed for all features and each class label.

The following diagrams display the relative importance of features with respect to the predicted classes: **Asthma**, **Bronchiolitis**, **COPD**, and **COVID**. These visualizations are based on the formal concepts extracted during the explanation process, and they reflect how strongly each feature contributes to each individual class.

A higher bar indicates that the corresponding feature appears more frequently and significantly in the concepts associated with that class. This graphical representation supports the interpretability of the model by highlighting the most influential symptoms for each diagnosis.

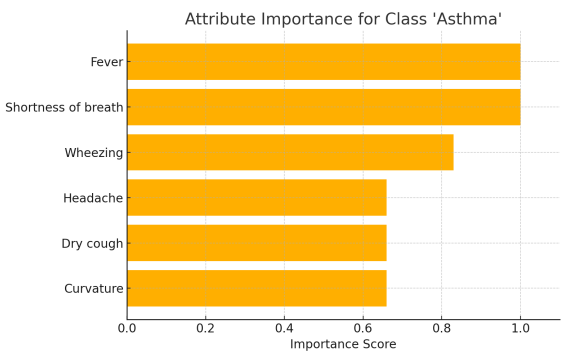


Figure 6: Feature importance for the class *Asthma*

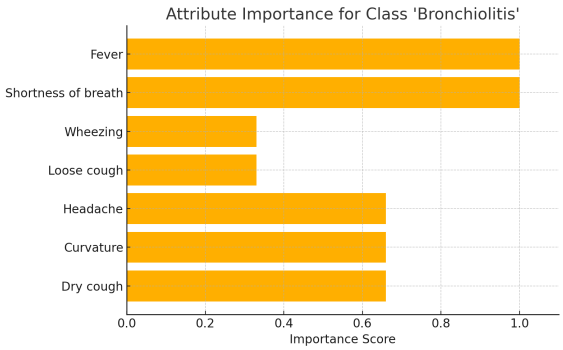


Figure 7: Feature importance for the class *Bronchiolitis*

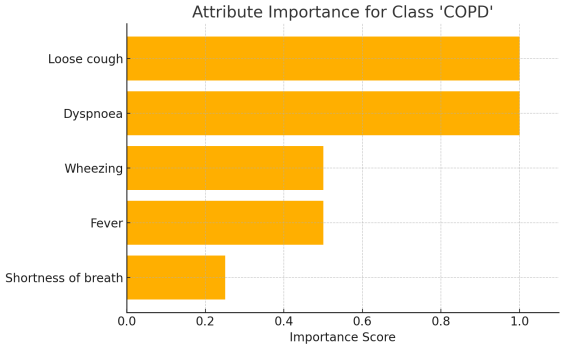


Figure 8: Feature importance for the class *COPD*

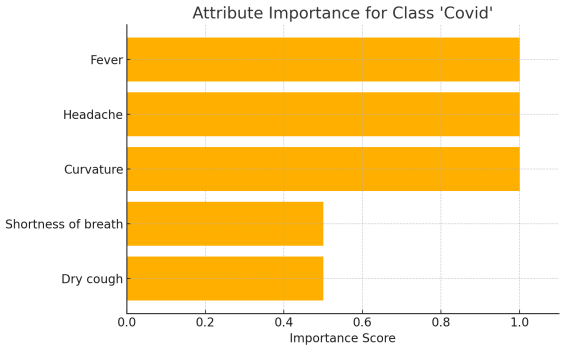


Figure 9: Feature importance for the class *COVID*

The proposed importance score offers a simple yet effective way to quantify how relevant each feature is to a specific class prediction. Unlike purely symbolic explanations, which identify sets of sufficient features, this measure provides a *numerical assessment* that reflects both the *frequency* and *specificity* of each feature in relation to the class.

Beyond its technical value, this measure is highly beneficial for the end users. It provides an interpretable, class-wise ranking of features, helping decision-makers, such as healthcare professionals or other domain experts, to better understand which features significantly influence the model's predictions. This contributes to building trust, enhancing transparency, and supporting informed decision-making.

6.4 Extending explanations

hus far, we have primarily focused on $\mathcal{DAS}$ explanations, which provide sufficient reasons for predicting a given label. Building on the concept of $\mathcal{DAS}$, we have also defined $\mathcal{SSA}$ explanations (which correspond to relevant features for a given prediction) derived from $\mathcal{DAS}$ explanations. Using these as a foundation, other explanations can be derived to address different types of XAI queries. Notably, we highlight the following:

– *Irrelevant attribute set $\mathcal{IAS}$ explanation :* An irrelevant feature is one that does not appear in any explanation of a prediction. Such features have no impact on the prediction y. It is clear that $\mathcal{IAS}$ can be easily determined as the features that are not included in $\mathcal{SSA}$:

$$\mathcal{IAS}(c_k) = \mathcal{P} \setminus \mathcal{SSA}(c_k)$$

where $\mathcal{P}$ denotes the set of all attributes.

– *Necessary attribute set $\mathcal{NAS}$ explanation :* Intuitively, a necessary feature is one that appears in every $\mathcal{DAS}$ explanation, making it essential for obtaining the prediction. This could indicate a highly relevant feature or a bias in the data, as the prediction relies on that feature. To extract the $\mathcal{NAS}$, it is sufficient to consider the intersection of all the $\mathcal{DAS}$ to directly identify them.

$$\mathcal{NAS}(c_k) = \cap\mathcal{DAS}(c_k)$$

– *$\mathcal{DAS}$-rules* : One might be interested in other types of explanations derived from $\mathcal{DAS}$. A challenge that arises when considering all possible explanations is how to filter and select them based on specific criteria of interest or quality. To address this, we propose the creation of $\mathcal{DAS}$-rules, which are sufficient reasons similar to the $\mathcal{DAS}$ explanations, but can also offer valuable insights into the *strength* or *prevalence* of the rule.

$\mathcal{DAS}$-rules are similar to anchor rules [41] and follow the form: IF $\mathcal{DAS}_i$(class) THEN predict class while indicating the percentage of data instances in the chosen neighborhood that comply with this rule. These IF-THEN rules capture local conditions for predictions. A $\mathcal{DAS}$-rule defines the prediction locally, meaning that changes to other features may not alter the prediction. While $\mathcal{DAS}_i(class)$ applies to one instance, a $\mathcal{DAS}$-rule covers its neighborhood.
Formally, for a given class c_i, a $\mathcal{DAS}$-rule is defined as an IF-THEN rule:

$$\text{IF } \mathcal{DAS}(c_i) \text{ THEN predict } c_i$$

where $\mathcal{DAS}(c_i) \in \mathcal{DAS}$ represents one of the Decisive Attribute Sets enabling the classifier f to predict class c_i. The number of $\mathcal{DAS}$-rules corresponds to the number

of distinct $\mathcal{DAS}$ explanations for a specific class c_i. The coverage metric evaluates how frequently each $\mathcal{DAS}$-rule appears within the neighborhood of the instance being explained. A $\mathcal{DAS}$-rule is deemed relevant if the number of instances sharing the rule and yielding the same prediction as the instance surpasses a defined threshold. This measure, termed $\mathcal{DAS}$-rule coverage, reflects the percentage of instances in the neighborhood that agree with a specific explanation. To extract $\mathcal{DAS}$-rules, two steps are involved: (1) enumerate $\mathcal{DAS}$ explanations for the instance x to be explained, and (2) assess each $\mathcal{DAS}$ explanation to determine if its coverage (i.e., the percentage of agreeing instances in x's neighborhood) exceeds a predefined threshold t. The number of $\mathcal{DAS}$-rules cannot exceed the number of $\mathcal{DAS}$ explanations, and extracting these rules only cannot exceed the number ofrequires scanning x's neighborhood, initially generated during the computation of $\mathcal{DAS}$ explanations. A $\mathcal{DAS}$-rule is valid at threshold t if:

$$\mathcal{DAS}(c_i) \Rightarrow c_i$$

with Coverage $(\mathcal{DAS}(c_i) \Rightarrow c_i) \geq t$.

7 Experimental study

To evaluate the practical feasibility of our method, we performed experiments on three widely-used multi-label datasets. The first set of experiments examines the relatively small Stack Overflow dataset, which serves as a case study to demonstrate the types of explanations our approach can generate. The remaining two experiments, conducted on the Yelp and TMC2007-500 dataset, aim to analyze the impact of some parameters on the number of explanations, their size, and the time needed for enumeration.

7.1 Experimental setup

In order to present our experimental evaluation, we first recall the main steps of our explanation method and then present the experimental setup.

1. *Generate local contexts*: In this step, we generate the formal contexts around the instance to be explained. This task is performed in our experiments using the k-Nearest Neighbors (KNN) algorithm. This step is detailed in Section 5.1.
2. *Identify and extract all formal concepts*;
3. *Minimize formal concepts* (see Section 5.3 for more details on these last two steps);
4. *Compute the explanations* (namely, the $\mathcal{DAS}$ and $\mathcal{SSA}$).

7.1.1 Datasets

To evaluate the effectiveness of our explanation method, we employed three multi-label datasets. The first is a relatively small dataset derived from the Stack Overflow platform, designed to clearly illustrate the behavior of our approach. Its reduced size and controlled structure enable fine-grained analysis and visualization of the generated explanations. The other two datasets, Yelp and TMC2007-500 are widely used benchmarks in the multi-label learning literature and serve to assess the scalability and robustness of our method under more realistic and complex conditions.

Stack overflow dataset This dataset was specifically constructed for our case study on explainable multi-label classification. It contains 500 programming-related questions collected from the Stack Overflow platform. Each instance is represented by a 100-dimensional binary vector, where each feature encodes the presence or absence of selected terms from the question's title and description. Standard text preprocessing techniques were applied, including tokenization, stop-word removal, and vocabulary filtering.

The label space consists of five widely used technologies: *Asp.net*, *Java*, *MySQL*, *PHP*, and *Python*. Each question may be associated with one or more of these labels, forming a typical multi-label setting. We deliberately constructed this small dataset to facilitate the visualization and analysis of the explanations produced by our method—something that would be considerably more difficult to achieve with large-scale datasets.

The dataset is publicly available at: `https://github.com/hakimradja/fca-explainability/tree/main/stack-overflow`

Yelp dataset The Yelp dataset was obtained from the COMETA repository (`https://cometa.ujaen.es/datasets/Yelp`) and originates from the Yelp Dataset Challenge [47]. It comprises 10,806 instances, each represented by 676 binary features, including 5 target labels and 671 descriptive attributes. The input features are constructed from a bag-of-words representation of textual reviews, where each attribute encodes the presence or absence of a specific token following standard text preprocessing.

The classification task is inherently multi-label: each review may belong to multiple predefined categories (e.g., food, service, ambiance). This dataset presents a high-dimensional, sparse input structure, which makes it a suitable benchmark for evaluating the performance and interpretability of multi-label classification models.

TMC2007-500 dataset The TMC2007-500 dataset was also obtained from the COMETA repository (`https://cometa.ujaen.es/datasets/tmc2007_500`). It is derived from the Aviation Safety Reporting System (ASRS) and contains 2,200 textual incident reports related to aviation safety. Each report is encoded as a binary vector of 500 features, extracted using a bag-of-words representation following standard preprocessing steps (tokenization, stop-word removal, and term selection).

The label space includes 22 distinct safety categories, and each report may be annotated with multiple labels reflecting the multifactorial nature of aviation incidents (e.g., human factors, procedural deviations, environmental hazards). The dataset exhibits an with an average label cardinality of 2.22 (i.e., the average number of labels per instance), indicating that each report is typically associated with more than two categories. This dataset serves as a realistic benchmark for evaluating the scalability and explanatory power of multi-label classification methods [48].

7.1.2 Parameter settings

In all experiments, we used a consistent 70/30 split between training and testing data. We considered three widely used multi-label classification algorithms known for their effectiveness: Label Powerset (LP), Random k-Labelsets (RAkEL), and Multi-Label k-Nearest Neighbors (ML-KNN). Each of these methods relies on specific parameters, which we set following commonly accepted practices in the literature.

For the Label Powerset (LP) method, we used a Random Forest as the base classifier, configured with 100 trees (`n_estimators = 100`). Other hyperparameters of the Random Forest remained at their default values. This combination provides a balance between predictive performance and robustness.

The RAkEL ensemble method was configured with standard parameter choices: the number of LP models was set to 10 (`model_count = 10`), and each LP model was trained on random subsets of 3 labels (`labelset_size = 3`). We used a Decision Tree as the base classifier for each individual model. These values are widely used in benchmark studies and provide a trade-off between diversity and computational efficiency.

For the ML-KNN method, we used the default settings from the `scikit-multilearn` library. The number of neighbors was set to $k = 10$, and Laplace smoothing was applied with a parameter $s = 1.0$. All neighbors were weighted equally in distance calculations (`weighting = 'uniform'`).

7.1.3 Neighborhood selection

To generate local contexts around the instance to be explained, we applied the K-Nearest Neighbors (KNN) algorithm using the Euclidean distance metric. The size of the neighborhood, denoted by k, directly affects the trade-off between the locality of the explanations and the generality of the extracted concepts.

More concretely:

- Smaller neighborhoods (e.g., $k = 10$) lead to more specific concepts that are closely tailored to the instance, but less generalizable to other cases.
- Larger neighborhoods (e.g., $k = 100$) yield more general concepts based on a wider range of instances, but those concepts may be less precise for the instance being explained.

We empirically explored several values of k, ranging from 10 to 100. We selected the values $\{20, 50, 100\}$ to reflect three representative levels of locality in explanations:

- $k = 20$: small neighborhoods yielding more specific, fine-grained concepts,
- $k = 50$: moderate size allowing a balance between specificity and generality,
- $k = 100$: larger neighborhoods resulting in more general and potentially more stable concepts.

Rather than exploring every possible value of k between 10 and 100, we adopted a non-uniform sampling strategy to reduce redundancy and keep the explanation process computationally tractable. Preliminary tests showed that intermediate values (e.g., 30, 40, 60, 70, 80...) did not yield substantially different conceptual structures or explanation patterns. Thus, we chose a compact yet representative set of neighborhood sizes.

7.1.4 Concept extraction and explanation generation

After generating the formal context representing the neighborhood of the instance to be explained, we apply the Next Closure algorithm to extract the complete set of formal concepts. Since the number of concepts can be large and some may be redundant, we filter the results to retain only the most informative concepts. From this reduced set, we derive the explanations in terms of *Decisive Attribute Sets (DAS)* and *Significant Attribute Sets (SSA)*.

To summarize the previously described explanation process, the following algorithms outline the main steps involved in generating symbolic explanations from a target instance using Formal Concept Analysis (FCA).

	Title	Tags
0	Flask-SQLAlchemy - When are the tables/databas...	['python', 'mysql']
1	Combining two PHP variables for MySQL query	['php', 'mysql']
2	Counting' the number of records that match a c...	['php', 'mysql']
3	Insert new row in a table and auto id number. ...	['php', 'mysql']
4	Create Multiple MySQL tables using PHP	['php', 'mysql']
...	...	...

Figure 10: Example question from the Stack Overflow dataset

Algorithm 3 FCA-based Explanation Generation — Data Preparation

Input
- Labeled dataset (X, Y) with binary attributes and multi-label targets;
- Multi-label classifier (e.g., Label Powerset with Random Forest);
- Target instance x to be explained;
- Neighborhood size k (e.g., $k = 20$)

Output
- Binary formal context for x's neighborhood

1. Split (X, Y) into training and test sets.
2. Train the classifier on the training data.
3. Select a target instance x from the test set.
4. Find the k nearest neighbors of x and predict their labels.
5. Concatenate neighbors and their predicted labels into a local dataset.
6. Transform the local dataset into a binary formal context:
   (a) Create compound attributes (c_j, a_i).
   (b) Encode the binary presence of each compound attribute.

Algorithm 4 FCA-based Explanation Generation (Part 2: Concept Extraction & Explanation)

Input
- Binary formal context from Part 1

Output
- Reduced set of formal concepts;
- For each label c_k, we derive its Decisive Attribute Sets (DAS) as well as its Significant Attribute Set (SSA(c_k));
- For each label c_k, we compute the importance degree of each attribute involved in its prediction.

1. Generate all formal concepts using the NextClosure algorithm.
2. Reduce the set by removing redundant concepts and those with low explanatory value.
3. For each label, compute the Decisive Attribute Set ($\mathcal{DAS}$) and the Significant Attribute Set ($\mathcal{SSA}(c_k)$).
4. Return the reduced set of concepts, along with $\mathcal{DAS}$ and $\mathcal{SSA}(c_k)$ for each label.
5. Return the attribute importance values for each class.

7.2 Case study on stack overflow data

This section presents a case study on the Stack Overflow dataset, derived from a widely used platform of programming questions and answers. Each question includes a *Title*, *Description*, and *Tags*, as shown in Figure 11.

Our prediction task involves assigning appropriate tags (labels) to a question based on its title and description. This resulted in a multi-label dataset characterized by:

- 499 instances,
- 100 features,
- 5 labels (*Asp.net, Java, MYsql, Php, Python*).

As previously introduced, we focus on three prominent multi-label classification methods: *Label Powerset (LP)*, *ML-KNN*, and *RAKEL*, which respectively represent problem transformation, problem adaptation, and a hybrid ensemble approach. To explain the predictions made by these classifiers for individual instances, we calculated both the **Decisive Attribute Sets** ($\mathcal{DAS}$) and the **Significant Attribute Set** ($\mathcal{SSA}$) for each class. The decisive attributes reveal the specific features that were essential in the decision of each classifier, while the significant attributes highlight the most influential features across predictions for that class. Our experimental approach involved selecting an instance at random and examining its classification outcomes through progressively larger neighborhoods, adjusting the neighborhood size K from 10 to 100 in increments of 10. This produced 10 progressively larger local formal contexts, each consisting of K nearest neighbors. For each classifier, we analyzed these local contexts, observing how predictive patterns evolved as the neighborhood size changed.

Illustrative example: class mysql.1 with LP classifier

To analyze this process, we present a detailed example of predictions generated by the LP classifier for an instance within the **mysql.1** class. Here, we analyze a local formal context containing the 50 nearest neighbors (i.e., $K = 50$). We provide explanations in terms of both the $\mathcal{DAS}$ and $\mathcal{SSA}$, identifying the primary attributes driving the classifier's decisions. The dataset comprises five classes. For each, we specify the distinct Decisive Attribute Sets ($\mathcal{DAS}$) leveraged by the classifier, alongside the number of significant attributes. This analysis provides insight into the attribute combinations that have the greatest impact on the predictions for each class.

Decisive attribute sets for mysql.1 class

For the **mysql.1** class, we observed the following Decisive Attribute Sets:

- $\mathcal{DAS}(mysql.1)$={{'creating', 'site', 'aspnet', 'maps'}, {'mysql'}, {'sqlalchemy'}, {'database', 'python'}, {'table'}, {'pymysql'}, {'database'}, {'database', 'python', 'mysql'}, {'position', 'can'}, {'python', 'mysql'}, {'php'}, {'change'}, {'database', 'mysql', 'sqlalchemy'}, {'login'}, {'creating'}, {'string', 'arguments'}, {'database', 'storing', 'variable'}}

The **Significant Attribute Set** for mysql.1, representing attributes frequently critical across predictions, is:

– $\mathcal{SSA}(mysql.1)$={'python', 'sqlalchemy', 'storing', 'php', 'login', 'site', 'maps', 'database', 'creating', 'table', 'arguments', 'string', 'change', 'mysql', 'position', 'can', 'aspnet', 'variable', 'pymysql'}

Among the 50 nearest neighbors, the LP classifier assigned 36 instances to the mysql.1 class, using 17 unique $\mathcal{DAS}$ in this process. The distribution of instances across these sets is outlined in Table 4.

Table 4: Number of instances per $\mathcal{DAS}$ (class: mysql.1, classifier: LP, neighbors $K = 50$)

Instance count	$\mathcal{DAS}$ used
9	{'sqlalchemy'}
3	{'mysql'}
3	{'pymysql'}
3	{'database'}
3	{'database', 'python', 'mysql'}
2	{'creating', 'site', 'aspnet', 'maps'}
2	{'table'}
2	{'php'}
1	{'database', 'python'}
1	{'python', 'mysql'}
1	{'change'}
1	{'database', 'mysql', 'sqlalchemy'}
1	{'login'}
1	{'creating'}
1	{'string', 'arguments'}
1	{'database', 'storing', 'variable'}
1	{'position', 'can'}

To complement the symbolic explanations, we quantified the relative importance of the features contributing to the class `mysql.1` by leveraging a concept-based metric. For each attribute, its importance score was computed as the proportion of instances assigned to the `mysql.1` class that appear in at least one formal concept whose intent includes both the class label and the attribute. This metric thus captures how broadly an attribute participates in the justification of the classifier's decisions, reflecting both its recurrence across instances and its discriminative relevance.
As shown in Figure 11, the attributes *sqlalchemy*, *database*, and *mysql* obtained the highest importance scores. Their strong presence indicates that they are not only sufficient in isolated cases but also recurrently involved across multiple instances, making them decisive markers for the prediction of the class `mysql.1`. In contrast, other attributes such as *python*, *creating*, and *pymysql* exhibit moderate contributions, being involved only in a subset of explanations. Less frequent terms, including *php*, *table*, or *login*, contribute minimally to the explanation and can be considered secondary signals.
This numerical layer of explanation complements symbolic reasoning by providing a ranked perspective on the most influential features. Such a ranking not only highlights the decisive attributes but also offers end users—such as developers or database administrators—a clearer understanding of which features most strongly influence the model's predictions, thereby enhancing interpretability and transparency.

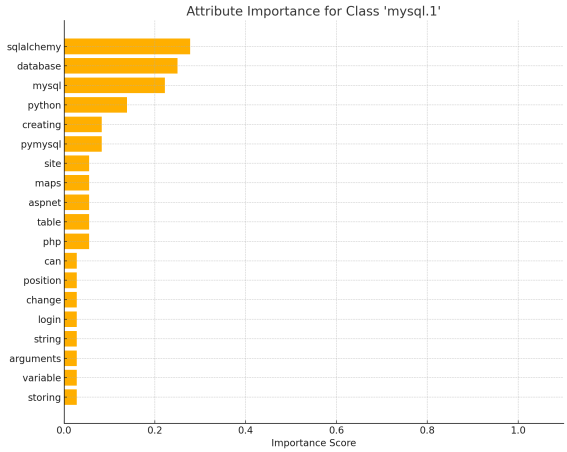


Figure 11: Feature importance for the class `mysql.1`.

$\mathcal{DAS}$**-rules**: We selected an instance randomly and generated a neighborhood of 50. The classifier used was LP. The $\mathcal{DAS}$-rules for the class mysql.1 are obtained from the $\mathcal{DAS}$ used to predict this class presented in Table 4.
Among the 50 nearest instances to the one being explained, 36 were predicted by the LP classifier in the class mysql.1. This was done using 17 different Decisive Attribute Sets ($\mathcal{DAS}$). Table 5 shows the $\mathcal{DAS}$-rules obtained with the coverage (proportion) of each rule.

Table 5: $\mathcal{DAS}$-rules obtained for the mysql.1 class of the Stack Overflow dataset (number of neighbors $K = 50$).

Coverage	$\mathcal{DAS}$-rule
0.18	$'sqlalchemy' \Rightarrow mysql.1$
0.06	$'mysql' \Rightarrow mysql.1$
0.06	$'pymysql' \Rightarrow mysql.1$
0.06	$'database' \Rightarrow mysql.1$
0.06	$'database' \wedge' python' \wedge' mysql' \Rightarrow mysql.1$
0.04	$'creating' \wedge' site' \wedge' aspnet' \wedge' maps' \Rightarrow mysql.1$
0.04	$'table' \Rightarrow mysql.1$
0.04	$'php' \Rightarrow mysql.1$
0.02	$'database' \wedge' python' \Rightarrow mysql.1$
0.02	$'python' \wedge' mysql' \Rightarrow mysql.1$
0.02	$'change' \Rightarrow mysql.1$
0.02	$'database' \wedge' mysql' \wedge' sqlalchemy' \Rightarrow mysql.1$
0.02	$'login' \Rightarrow mysql.1$
0.02	$'creating' \Rightarrow mysql.1$
0.02	$'string' \wedge' arguments' \Rightarrow mysql.1$
0.02	$'database' \wedge' storing' \wedge' variable' \Rightarrow mysql.1$
0.02	$'position' \wedge' can' \Rightarrow mysql.1$

Table 6: Number of Decisive Attribute Sets ($\mathcal{DAS}$) per class for a selected instance, based on varying neighborhood sizes from 10 to 100. Experiments conducted using LP, ML-KNN, and RAKEL classifiers

Neighbors	Classifier	Asp.net.1	Java.1	Mysql.1	Php.1	Python.1
10	LP	1	0	1	0	0
	ML-KNN	1	1	3	0	2
	RAKEL	1	0	1	0	0
20	LP	1	0	4	0	3
	ML-KNN	1	1	4	0	3
	RAKEL	1	0	4	0	3
30	LP	1	1	8	1	7
	ML-KNN	1	1	9	1	6
	RAKEL	1	1	8	1	7
40	LP	1	1	12	3	9
	ML-KNN	1	1	14	5	8
	RAKEL	1	1	12	3	10
50	LP	1	1	17	6	11
	ML-KNN	1	1	18	8	9
	RAKEL	1	1	17	6	12
60	LP	1	1	23	10	11
	ML-KNN	1	1	21	10	10
	RAKEL	1	1	23	10	17
70	LP	1	1	31	12	20
	ML-KNN	1	1	30	12	16
	RAKEL	1	1	31	12	24
80	LP	1	1	35	12	24
	ML-KNN	1	1	33	12	17
	RAKEL	1	1	33	12	28
90	LP	1	1	37	12	25
	ML-KNN	1	1	34	14	17
	RAKEL	1	1	37	12	30
100	LP	1	1	45	15	28
	ML-KNN	1	1	44	19	21
	RAKEL	1	1	45	15	34

For example, the $\mathcal{DAS}$-rule The $\mathcal{DAS}$-rule "IF 'sqlalchemy' THEN predict `mysql.1`" with a coverage of 0.18 means that 18% of the 50 data instances close to the one being explained (9 out of the 50 instances) were predicted by the multi-label LP classifier in the `mysql.1` class when the 'sqlalchemy' attribute is present.
To trace the behavior of each classifier, Table 6 and Table 7 below summarize the results obtained when we applied three different algorithms — Label Powerset (LP), ML-KNN, and RAKEL on our dataset. For each algorithm, we provide both the number of Decisive Attribute Sets ($\mathcal{DAS}$) used and the percentage of Significant Attributes ($\mathcal{SSA}$) across each class in the dataset.

Analyzing the data from Table 6 reveals that the number of Decisive Attribute Sets ($\mathcal{DAS}$) for each class varies depending on the classifier. This indicates that each algorithm employs a unique selection of attributes to make predictions for a given class or set of classes, indicating that each classifier's decision-making relies on different aspects of the data. For instance, when using a neighborhood of 10 instances, both the LP and RAKEL classifiers converged on the same Decisive Attribute Set for predicting the $mysql.1$ class:

- $\mathcal{DAS}(mysql.1)$ = {'creating', 'maps', 'aspnet', 'site'}

In contrast, the ML-KNN classifier used three distinct Decisive Attribute Sets for the same class:

- $\mathcal{DAS}_1(mysql.1) = \{\text{creating, maps, aspnet, site}\}$
- $\mathcal{DAS}_2(mysql.1) = \{\text{sqlalchemy}\}$
- $\mathcal{DAS}_3(mysql.1) = \{\text{database, web, python, server, connection}\}$

This variability suggests that ML-KNN considers multiple unique combinations of attributes, reflecting a broader set of criteria in its predictions for the$mysql.1$ class. Furthermore, we observe that the number of $\mathcal{DAS}$ per class changes as the neighborhood size increases. This occurs in two distinct ways:

1. Constant $\mathcal{DAS}$ Count: For some classes, such as $java.1$ and $asp.net.1$, the number of $\mathcal{DAS}$ remains stable across neighborhood changes. This may occur when new instances added to the neighborhood share similar attribute patterns or when the class is infrequent, resulting in fewer new instances being predicted in that class.

2. Increasing $\mathcal{DAS}$ Count: In classes like $mysql.1$ and $python.1$, the number of $\mathcal{DAS}$ increases as new instances are added with distinct attributes that influence classifier predictions. This growth reflects a dynamic adaptation in the classifiers, where additional unique attributes contribute to classifying more instances in these classes.

This analysis highlights how each classifier interprets the feature space differently, providing diverse perspectives on attribute importance across varying neighborhoods. The number of significant attributes utilized by each classifier for making predictions can differ markedly from one algorithm to another. For instance, when analyzing a neighborhood of 10 instances, both the LP and RAKEL classifiers relied on only 4% of the available attributes to accurately predict the class $mysql.1$. This observation underscores the distinct strategies employed by different classifiers in their decision-making processes. Furthermore, we have identified that the number of Decisive Attribute Sets ($\mathcal{DAS}$) for each class varies across different neighborhoods, which occurs in three primary ways:

1. **Decreasing Percentage of Significant Attributes**: In some instances, certain attributes lose their significance in class predictions due to the introduction of new instances. For example, the attributes previously deemed essential for predicting the $asp.net.1$ class may no longer hold the same predictive power in the expanded neighborhood context.

2. **Constant Percentage of Significant Attributes**: This situation arises when newly added instances share the same attributes as those predicted in the previous neighborhood, or when the class in question is infrequent, resulting in no new instances classified within that category. This stability is evident in classes like $java.1$ and $asp.net.1$, where the attribute significance remains unchanged despite neighborhood modifications.

3. **Increasing Percentage of Significant Attributes**: When new instances are introduced with distinct attributes that help the classifiers predict a given class, the percentage of significant attributes can increase. This phenomenon is observable in classes such as $mysql.1$, $java.1$, and $python.1$, where the addition of new information enhances the classifier's ability to identify and categorize instances.

This variability in the percentage of significant attributes across classifiers and neighborhoods highlights the adaptive nature of the evaluated algorithms and their reliance on the feature space context for effective predictions.

We randomly selected 10 instances to further investigate our experimental study. For each instance, we calculated the number of Decisive Attribute Sets, denoted as $\mathcal{DAS}$, as well as the percentage of Significant Attributes per class. Table 8 and Table 9 present the average number of $\mathcal{DAS}$ and the percentage of Significant Attributes for these 10 instances across different classes, with neighborhood sizes ranging from 10 to 100. In addition, we measured the execution time (in seconds) for processing each neighborhood size.

From the experimental results, we observe that the classes `asp.net.1` and `java.1` are affected by a significant class imbalance, which explains why they exhibit a low number of Decisive Attribute Sets ($\mathcal{DAS}$) and significant attributes ($\mathcal{SSA}$). This issue arises because these classes are rarely predicted by any of the three classifiers across the entire dataset. As a result, the number of $\mathcal{DAS}$ and the percentage of $\mathcal{SSA}$ do not vary substantially when moving between different neighborhood sizes. For some neighborhood sizes, these metrics are equal to zero, indicating that the classifier did not predict these classes at all. For instance, both the LP and RAKEL classifiers fail to predict the `java.1` class for small neighborhood sizes (10, 20, and 30).

In contrast, for the other classes, such as `mysql.1`, `php.1`, and `python.1`, we observe more variability in the number of $\mathcal{DAS}$ and $\mathcal{SSA}$. These metrics vary with both the number of instances predicted by the classifiers and the new attributes they use for prediction. The results indicate that as the classifier predicts more instances, the number of decisive attributes and the percentage of significant attributes tend to increase, reflecting the evolving attribute patterns associated with the class.

For example, the LP classifier used on average 16.91 $\mathcal{DAS}$ and 12.82% $\mathcal{SSA}$ to predict the `php.1` class with a neighborhood size of 90. In comparison, the ML-KNN classifier used 22.61 $\mathcal{DAS}$ and 15.91% $\mathcal{SSA}$, while the RAKEL classifier used 17.05 $\mathcal{DAS}$ and 13.1% $\mathcal{SSA}$. These differences reflect how each classifier selects and utilizes attributes to predict the classes in the dataset.

The observed differences in the number of decisive attributes and significant features across the classifiers can be attributed to the distinct approaches they employ to address multi-label classification. The LP classifier employs a *problem transformation* technique, which converts the multi-label problem into a multi-class task. It treats each combination of labels from the training set as a separate class and learns a multi-class classifier accordingly.

The ML-KNN classifier, on the other hand, follows a *problem adaptation* approach. It extends the standard KNN algorithm by integrating *Bayesian inference*,allowing the classifier to estimate both prior and posterior probabilities of each label based on the training examples.

Table 7: Significant attributes $\mathcal{SSA}$ percentage per class (for an instance). Experiments are done with 10, 20, ..., 100 neighbors using LP, ML-KNN, and RAKEL classifiers.

Neighbors	Classifiers	Asp.net.1	Java.1	Mysql.1	Php.1	Python.1
10	LP	2	0	4	0	0
	ML-KNN	1	1	10	0	6
	RAKEL	2	0	4	0	0
20	LP	1	0	11	0	7
	ML-KNN	1	1	7	0	3
	RAKEL	1	0	11	0	7
30	LP	1	1	11	1	7
	ML-KNN	1	1	11	1	6
	RAKEL	1	1	11	1	7
40	LP	1	1	14	3	8
	ML-KNN	1	1	15	5	7
	RAKEL	1	1	14	3	9
50	LP	1	1	19	8	10
	ML-KNN	1	1	20	10	9
	RAKEL	1	1	19	8	11
60	LP	1	1	23	12	11
	ML-KNN	1	1	22	12	9
	RAKEL	1	1	23	12	13
70	LP	1	1	27	11	15
	ML-KNN	1	1	25	11	13
	RAKEL	1	1	27	11	18
80	LP	1	1	28	11	16
	ML-KNN	1	1	26	11	13
	RAKEL	1	1	28	11	19
90	LP	1	1	28	11	16
	ML-KNN	1	1	26	12	13
	RAKEL	1	1	28	11	19
100	LP	1	1	32	15	16
	ML-KNN	1	1	32	16	15
	RAKEL	1	1	32	15	19

The RAKEL classifier builds on the LP approach but introduces an improvement by using *multiple multi-class classifiers*, each trained on a randomly selected subset of labels. This method enhances the flexibility and adaptability of the classifier, especially for complex multi-label datasets.

Interestingly, when considering five labels, we found that the results produced by the LP and RAKEL classifiers were very similar. This suggests that the additional complexity introduced by RAKEL does not necessarily result in better performance across all scenarios, particularly when the number of labels is relatively small.

7.3 Expermients on yelp and TMC2007-500 datasets

In addition to the Stack Overflow dataset, we also experimented with other datasets, specifically Yelp and TMC2007-500. For these datasets, we focused on neighborhood sizes of 20, 50, and 100. This choice allowed us to capture key insights without overloading the analysis with too many details, making it easier to compare classifier performance across a manageable range. By using these particular neighborhood sizes, we aimed to strike a balance between computational efficiency and capturing enough variation to reveal meaningful patterns in the data.

Yelp dataset

The dataset consists of user-generated reviews and ratings collected from Yelp. The dataset includes more than 10,000 reviews, providing a robust basis for analysis. There are 671 features and 5 binary labels which are:

- IsFoodGood: refers to the food,
- IsServiceGood: refers to the service,
- IsAmbianceGood: the look and feel of the restaurant,
- IsDealsGood: are there good deals,
- IsPriceGood: the worthiness and the value for money perceived by the reviewer.

Table 8: Average number of Decisive Attribute Sets ($\mathcal{DAS}$) per class (for 10 instances). Experiments are done with 10, 20, ..., 100 neighbors using LP, ML-KNN, and RAKEL classifiers.

Neighbors	Classifiers	Asp.net.1	Java.1	Mysql.1	Php.1	Python.1	Exectime
10	LP	0,09	0,00	1,09	0,82	0,45	0,015
	ML-KNN	0,09	0,09	1,73	1,36	0,27	0,015
	RAKEL	0,05	0	2	1,35	1,35	0,015
20	LP	0,09	0,00	3,36	1,73	1,64	0,031
	ML-KNN	0,09	0,09	3,36	2,73	0,45	0,016
	RAKEL	0,05	0	4,25	2,65	2,70	0,047
30	LP	0,09	0,09	6,09	3,36	2,55	0,046
	ML-KNN	0,09	0,18	6,91	5,09	1,36	0,047
	RAKEL	0,05	0,05	7,15	4,45	4,70	0,047
40	LP	0,09	0,18	10,18	7,00	4,00	0,047
	ML-KNN	0,09	0,27	10,91	8,00	2,73	0,062
	RAKEL	0,05	0,10	10,85	7,00	6,30	0,078
50	LP	0,09	0,18	15,27	9,64	6,00	0,078
	ML-KNN	0,09	0,45	15,73	12,09	3,36	0,067
	RAKEL	0,05	0,25	16,65	9,35	10,70	0,078
60	LP	0,09	0,36	18,64	10,73	7,91	0,093
	ML-KNN	0,09	0,45	20,64	15,73	4,36	0,093
	RAKEL	0,05	0,35	18,00	10,60	13,70	0,110
70	LP	0,09	0,45	24,09	13,00	10,73	0,094
	ML-KNN	0,09	0,82	24,73	16,09	5,91	0,110
	RAKEL	0,05	0,40	25,50	12,75	18,20	0,110
80	LP	0,09	0,73	28,55	14,73	13,18	0,140
	ML-KNN	0,09	0,91	30,27	20,64	8,09	0,140
	RAKEL	0,05	0,70	30,10	14,60	20,80	0,160
90	LP	0,09	0,91	35,73	16,91	17,36	0,170
	ML-KNN	0,09	1,00	34,36	22,64	11,09	0,170
	RAKEL	0,05	0,90	37,85	17,05	26,10	0,200
100	LP	0,09	0,91	42,55	20,00	19,36	0,230
	ML-KNN	0,09	1,00	43,45	25,64	13,18	0,210
	RAKEL	0,05	0,90	44,45	18,70	29,15	0,230

Table 10 and Table 11 show respectively the average number of $\mathcal{DAS}$ and average percentage of significant attributes $\mathcal{SSA}$ for the LP, ML-KNN and RAKEL algorithms, respectively. These results are based on 10 randomly chosen instances with neighborhoods consisting of 20, 50, and 100 neighbors.

The same findings seen on the Stack Overflow dataset also apply to the Yelp dataset. This is likely because both datasets have a reduced number of classes, which is equal to 5. However, the number of explanations ($\mathcal{DAS}$ and significant attributes $\mathcal{SSA}$) is greater with the Yelp dataset, something that is expected since the number of attributes in this dataset is greater (671 compared to 100 attributes in the Stack Overflow dataset).

TMC2007-500

The TMC2007-500 dataset is a reduced version of the Aviation Safety Reporting System (ASRS) dataset, consisting of 28,596 flight safety narratives reported by crew members. It consists of 500 features and 22 categories, each representing a distinct issue encountered during flights. This subset was generated by applying feature selection to retain the most relevant 500 terms from the original TMC2007 dataset.

As for the two previous datasets, we conducted the experiments on 10 randomly chosen instances with neighborhoods consisting of 20, 50, and 100 instances. The average results, in terms of $\mathcal{DAS}$ and $\mathcal{SSA}$, obtained by the three classifiers (ML-KNN, RAKEL, and LP), are presented in Table 12 and Table 13 below.

As observed with the other datasets, the number of explanations for a given classifier tends to increase as the neighborhood size increases and new attributes are introduced in the prediction process. However, we noticed that for the TMC2007-500 dataset, which contains 22 classes, the number of explanations obtained for the RAKEL classifier is higher. This means that RAKEL is the classifier that used the most attributes to make predictions, compared to the other two classifiers, LP and ML-KNN. This can be explained by the fact that RAKEL is a more complex classifier than LP and ML-KNN, and is therefore more able to deal with complex neighborhoods. RAKEL leverages multiple

Table 9: Average percentage of significant attributes $\mathcal{SSA}$ per class (for 10 instances). Experiments are done with 10, 20, ..., 100 neighbors using LP, ML-KNN, and RAKEL classifiers.

Neighbors	Classifiers	Asp.net.1	Java.1	Mysql.1	Php.1	Python.1	Exectime
10	LP	0,18	0,00	1,55	0,82	0,45	0,015
	ML-KNN	0,09	0,09	2,45	1,36	0,73	0,015
	RAKEL	0,10	0,00	1,90	1,15	1,40	0,015
20	LP	0,09	0,00	3,55	1,45	2,00	0,031
	ML-KNN	0,09	0,09	3,36	2,55	0,55	0,016
	RAKEL	0,05	0,00	4,25	2,05	2,85	0,047
30	LP	0,09	0,09	5,45	2,91	2,45	0,046
	ML-KNN	0,09	0,36	6,00	4,27	1,64	0,047
	RAKEL	0,05	0,05	6,35	3,50	4,20	0,047
40	LP	0,09	0,36	8,82	5,91	3,64	0,047
	ML-KNN	0,09	0,64	9,18	6,64	2,82	0,062
	RAKEL	0,05	0,20	9,80	6,15	6,55	0,078
50	LP	0,09	0,36	12,64	8,45	4,91	0,078
	ML-KNN	0,09	0,64	12,55	9,55	3,64	0,067
	RAKEL	0,05	0,35	13,40	8,30	8,00	0,078
60	LP	0,09	0,55	14,73	9,27	6,36	0,093
	ML-KNN	0,09	1,18	15,55	11,64	4,55	0,093
	RAKEL	0,05	0,45	15,80	9,50	9,45	0,110
70	LP	0,09	0,82	17,55	10,18	8,27	0,094
	ML-KNN	0,09	2,09	18,09	12,91	5,91	0,110
	RAKEL	0,05	0,55	18,30	10,30	10,70	0,110
80	LP	0,09	1,09	19,82	11,36	9,55	0,140
	ML-KNN	0,09	2,00	21,09	16,00	7,27	0,140
	RAKEL	0,05	0,90	20,70	11,65	12,70	0,160
90	LP	0,09	1,09	23,73	12,82	11,73	0,170
	ML-KNN	0,09	1,91	26,00	15,91	9,36	0,170
	RAKEL	0,05	1,00	23,25	13,10	14,80	0,200
100	LP	0,09	1,09	27,91	15,45	12,09	0,230
	ML-KNN	0,09	1,73	30,09	18,55	10,18	0,210
	RAKEL	0,05	1,00	28,10	15,15	15,90	0,230

Table 10: Average number of Decisive Attribute Sets ($\mathcal{DAS}$) per class (for 10 instances). Experiments are done with 20, 50, and 100 neighbors using LP, ML-KNN, and RAKEL classifiers.

Neighbors	Classifiers	Isfoodgood	Isservicegood	Isambiancegood	Isdealsgood	Ispricegood	Execution time
20	LP	15.18	9.64	7.09	0.00	6.55	0.82
	ML-KNN	17.91	7.36	3.91	0.82	1.64	0.85
	RAKEL	1.36	5.73	13.82	8.91	8.82	1.57
50	LP	38.18	25.18	19.45	0.00	14.64	1.13
	ML-KNN	43.91	22.82	15.27	1.27	3.73	0.90
	RAKEL	4.36	12.00	31.00	15.00	20.82	6.24
100	LP	75.82	47.09	39.18	0.00	21.82	2.21
	ML-KNN	87.09	43.36	38.00	2.09	7.91	1.66
	RAKEL	13.64	22.00	57.09	20.18	39.09	30.03

LP models trained on random subsets of labels, enabling it to capture a wider variety of attribute–label patterns in complex neighborhoods.

8 Comparative analysis with SOTA explainability methods

Our approach is distinguished for its ability to identify Decisive Attribute Sets DAS(c_k) with increased granularity, outperforming several existing attribute-based methods. Unlike approaches like LIME [17] and SHAP [18], which mainly focus on individual feature importance, our

Table 11: Significant attributes ($\mathcal{SSA}$) percentage per class (for an instance). Experiments are done with 20, 50, and 100 neighbors using LP, ML-KNN, and RAKEL classifiers.

Neighbors	Classifiers	Isfoodgood	Isservicegood	Isambiancegood	Isdealsgood	Ispricegood	Execution time
20	LP	9.89	7.15	5.87	0.00	3.95	0.82
	ML-KNN	10.55	7.71	6.62	0.33	3.84	0.85
	RAKEL	4.95	7.87	12.76	7.96	9.93	1.57
50	LP	17.73	10.89	10.56	0.00	5.89	1.13
	ML-KNN	22.13	14.00	13.02	1.33	5.22	0.90
	RAKEL	6.82	11.07	21.49	9.69	16.69	6.24
100	LP	34.98	20.33	18.98	0.00	9.16	2.21
	ML-KNN	42.02	22.49	20.22	1.42	9.27	1.66
	RAKEL	14.27	18.42	32.15	12.93	26.45	30.03

Table 12: Average number of Decisive Attribute Sets ($\mathcal{DAS}$) per class (for 10 instances).

Neighbors	Classifiers	1	2	3	...	21	22	Execution time
20	LP	0.14	9.00	0.29	...	0.00	0.00	25.57
	ML-KNN	0.86	8.57	0.00	...	0.00	0.00	16.58
	RAKEL	9.43	8.86	11.57	...	1.43	1.14	21.86
50	LP	1.43	23.14	1.43	...	0.00	0.00	82.89
	ML-KNN	1.29	22.14	0.00	...	0.00	0.00	19.90
	RAKEL	20.86	24.14	27.29	...	4.14	4.00	102.92
100	LP	4.86	44.29	4.57	...	0.00	0.14	257.87
	ML-KNN	2.43	43.14	0.00	...	0.86	0.43	42.15
	RAKEL	34.43	51.29	49.43	...	8.71	8.14	1200.42

method leverages reduced formal concepts to capture interactions between attributes. For example, in the case of the Yelp Reviews dataset, LIME could assign scores to attributes like "delicious: +0.8" or "friendly: +0.7" independently, but it does not link these attributes to explain the prediction as a whole. Similarly, SHAP, while providing additive contributions for each attribute, does not effectively distinguish the critical attribute combinations that lead to a given prediction. In contrast, our method identifies meaningful sets of attributes, such as "food quality", "fast service", and "friendly atmosphere", which together form complete justifications for a positive prediction.

In comparison to rule-based approaches like those proposed by Ribeiro et al. [41], which generate local rules, our method offers a significant advantage in terms of coverage. By leveraging formal concepts and their extents, we are able to generate both global and local explanations without needing to explicitly distinguish between the two. For instance, in the Stack Overflow dataset, our method identifies that "syntactic similarity in titles and the presence of specific keywords" is a universal rule for predicting duplicate questions, whereas rule-based approaches require a specific configuration for each individual instance. By capitalizing on interactions and providing justifications through groups of attributes, our approach better addresses the need for explainability, particularly in critical domains.

8.1 Dual-level interpretability (local and global)

Most state-of-the-art methods treat local and global explainability as distinct tasks, often requiring different algorithms and processing pipelines. Local explanations typically focus on understanding the prediction for a single in-

Table 13: Significant attributes ($\mathcal{SSA}$) percentage per class (for 10 instances).

Neighbors	Classifiers	1	2	3	...	21	22	Execution time
20	LP	0.19	6.07	0.64	...	0.00	0.00	25.57
	ML-KNN	0.32	3.53	0.00	...	0.00	0.00	16.58
	RAKEL	0.32	3.53	0.00	...	0.00	0.00	21.86
50	LP	1.20	11.89	1.73	...	0.00	0.00	82.89
	ML-KNN	0.30	6.46	0.00	...	0.00	0.00	19.90
	RAKEL	14.11	14.82	14.86	...	5.39	5.90	102.90
100	LP	1.65	17.09	3.27	...	0.00	0.47	257.87
	ML-KNN	0.32	13.05	0.00	...	1.67	1.03	42.15
	RAKEL	19.50	21.15	21.90	...	10.48	10.31	1200.42

stance (e.g., via perturbation or counterfactual reasoning), while global explanations provide aggregate insights over the dataset (e.g., feature importance or decision rules).

In contrast, our method offers a unified framework capable of producing both local and global insights. At the local level, we extract formal concepts from the neighborhood of a target instance, offering precise explanations of its predicted labels. At the global level, we aggregate recurring concepts across different instances to identify general patterns and relationships that consistently characterize specific classes. This dual-level interpretability simplifies the overall analysis workflow and enables consistent interpretive reasoning at multiple scales.

8.2 Computational efficiency and scalability considerations

Unlike model-agnostic methods such as SHAP and LIME, which rely on repeated evaluations of the predictive model, SHAP (especially in its kernel variant) exhibits exponential complexity with respect to the number of features due to its reliance on evaluating all possible feature subsets. Even optimized implementations like TreeSHAP [56] are limited to tree-based models and can be costly in terms of memory and computation.

LIME, on the other hand, perturbs the input instance to create synthetic samples and trains a surrogate model locally. This approach requires hundreds or even thousands of additional predictions, substantially increasing the computational burden.

Our method avoids both model access and sampling by leveraging a precomputed local neighborhood. It transforms this neighborhood into a formal context and applies symbolic reasoning via Formal Concept Analysis. Although in theory extracting concepts can be time-consuming when there are many attributes, in practice it runs efficiently because we only work with a small group of similar examples (the neighborhood) and most real datasets are sparse (not all features are active at once). Also, once the model has made its predictions, we don't need to retrain it or ask it for new predictions to generate explanations.

While our method performs well on standard multi-label datasets, we acknowledge that Formal Concept Analysis (FCA) can face scalability challenges, especially with high-dimensional datasets such as those encountered in image analysis or genomics. In theory, the number of formal concepts can grow exponentially with the number of attributes, making the extraction of the reduced concept lattice computationally expensive.

To address this, our approach incorporates a concept reduction strategy that significantly reduces the search space by retaining only the most relevant concepts. Furthermore, the method focuses exclusively on concepts directly related to the predicted labels, thereby mitigating the explosion of the concept space.

For very large and high-dimensional datasets, scalability can be further improved through additional optimizations. These include dimensionality reduction techniques such as feature embeddings from pre-trained convolutional neural networks (CNNs) for images or statistical/biological relevance filtering for genomics, as well as sampling strategies and parallel computation to distribute the workload. Combined, these optimizations enable our FCA-based explainability framework to remain computationally feasible and interpretable even in large-scale and complex domains.

8.3 Model-agnostic and transparent mechanism

A notable strength of our approach is its full independence from the architecture or internal structure of the underlying classifier. Since our method only requires labeled examples from a local neighborhood, it can be applied to any type of model, including non-differentiable and ensemble models. This stands in contrast to many modern techniques, such as gradient-based attribution or TreeSHAP [56], which require access to internal parameters or model structures.

Moreover, the output of our method is symbolic and human-interpretable. Explanations take the form of object–attribute–class triplets or concise anchor rules that align with human reasoning. This makes our method particularly suitable for domains where explanations must be reviewed and validated by experts, such as healthcare or policy decision-making.

In addition to symbolic outputs, we also introduce a dedicated metric to quantify the importance of each attribute with respect to a given class. This metric is derived from the frequency and specificity of attributes across the extracted formal concepts, providing a principled, transparent view of feature relevance. Unlike SHAP or LIME, which assign weights based on statistical perturbations or approximations, our importance scores are grounded in the structural patterns observed in the data. This hybrid integration of symbolic rules and importance measures strengthens the interpretability and utility of the explanations.

8.4 Capturing attribute interactions

A key limitation of many additive attribution methods like SHAP and LIME is their assumption of feature independence. These methods assign importance scores to features in isolation, missing potentially critical interactions between them. In multi-label classification, such interactions are especially important, as specific combinations of features may jointly determine the presence of multiple labels.

Our approach overcomes this limitation by leveraging the structure of formal concepts to capture meaningful attribute combinations. By analyzing patterns shared across instances with common labels, we identify sets of attributes that co-occur in a logically coherent way. These combinations provide deeper insight into how features interact and contribute collectively to predictions.

9 Concluding remarks

In this work, we introduced a novel explainability framework for multi-label classification based on Formal Concept Analysis. Our approach provides a unified mechanism for generating both local and global explanations, without requiring access to the internal structure of the model. By leveraging symbolic reasoning, we identify Decisive Attribute Sets that explain predictions in a human-understandable manner, capturing key interactions between features.

A major strength of our method lies in its ability to derive explanations that are both concise and meaningful, using reduced formal concepts that generalize across instances. Additionally, we proposed a class-specific importance measure that quantifies the relevance of each attribute based on its presence across formal concepts, further supporting interpretability. This integration of symbolic structure and attribute importance enhances trust and transparency, especially in domains where interpretability is critical.

The experimental results confirm the scalability and practical viability of our method, making it a promising tool for advancing explainability in multi-label learning.

As a perspective, this work could be expanded to include counterfactual (or contrastive) explanations, such as determining the minimal perturbations (modifications) on attribute values that would change the predictions to a predefined alternative, and to consider not only active attributes but also the absence of certain attributes as potentially informative signals.

A Additional experiments

In the experimental section, we provided details for the *mysql.1* class only. We provide details of the other remaining classes in the stack overflow dataset below:

1. php.1

   - $\mathcal{DAS}(php.1)$={{'database', 'storing', 'variable'}, {'php'}, {'change'}, {'login'}, {'java'}, {'creating'}}
   - $\mathcal{SSA}(php.1)$={'login', 'java', 'database', 'creating', 'change', 'storing', 'variable', 'php}

Table 14: Number of instances per $\mathcal{DAS}$ (class: php.1, classifier: LP, neighborhood $K = 50$)

Instance count	$\mathcal{DAS}$ used
2	{'php'}
1	{'database', 'storing', 'variable'}
1	{'change'}
1	{'login'}
1	{'java'}
1	{'creating'}

the LP classifier used 8 out of 100 attributes in our dataset to predict an instance in the class 'php.1'. This gives a rate of 8% of attributes used.

2. python.1:

   - $\mathcal{DAS}(python.1)$={'mysql', 'sqlalchemy', 'database', 'python', 'table', 'pymysql', 'database', 'database', 'python', 'mysql', 'position', 'can', 'python', 'mysql', 'database', 'mysql', 'sqlalchemy', 'string', 'arguments'}
   - $\mathcal{SSA}(php.1)$={'arguments', 'position', 'string', 'python', 'sqlalchemy', 'database', 'table', 'pymysql', 'mysql', 'can'}

Table 15: Number of instances per $\mathcal{DAS}$ (class: python.1, classifier: LP, neighborhood $K = 50$)

Instances count	$\mathcal{DAS}$ used
9	{'sqlalchemy'}
3	{'pymysql'}
3	{'database', 'python', 'mysql'}
3	{'mysql'}
3	{'database'}
2	{'table'}
1	{'database', 'python'}
1	{'position', 'can'}
1	{'database', 'mysql', 'sqlalchemy'}
1	{'position', 'can'}
1	{'string', 'arguments'}

We notice that the set of $\mathcal{DAS}$ the classifier used to predict the class python.1 is included in the set of $\mathcal{DAS}$ that it used to predict the class mysql.1; this is most likely explained by the fact that the two classes are correlated. Indeed, in the 50 instances close to the instance being explained, each time the classifier predicts the class python.1, the class mysql.1 is predicted too.
Out of the total number of attributes contained in our dataset, the LP classifier uses 10 of them to predict the python.1 class (this equals to 10% of attributes used).

3. Asp.Net.1:

   - $\mathcal{DAS}(asp.net.1)$ ={'aspnet'}
   - $\mathcal{SSA}(asp.net.1)$={'aspnet'}

4. java.1

   - $\mathcal{DAS}(java.1)$={'java'}
   - $\mathcal{SSA}(java.1)$={'java'}

References

[1] Herman B. (2017) The promise and peril of human evaluation for model interpretability, *arXiv preprint arXiv:1711.07414*.

[2] Crawford K. (2016) Artificial intelligence's white guy problem, *The New York Times*, 25(06).

[3] Caliskan A., Bryson J. J., Narayanan A. (2017) Semantics derived automatically from language corpora contain human-like biases, *Science*, 356(6334), pp. 183–186. American Association for the Advancement of Science.

[4] Barocas S., Selbst A. D. (2016) Big data's disparate impact, *California Law Review*, 104, pp. 671–. HeinOnline.

[5] Coglianese C., Lehr D. (2016) Regulating by robot: Administrative decision making in the machine-learning era, *Georgetown Law Journal*, 105, pp. 1147–. HeinOnline.

[6] Friedler S. A., Scheidegger C., Venkatasubramanian S., Choudhary S., Hamilton E. P., Roth D. (2019) A comparative study of fairness-enhancing interventions in machine learning, *Proceedings of the Conference on Fairness, Accountability, and Transparency*, ACM, pp. 329–338.

[7] Lipton Z. C. (2018) The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery, *Queue*, 16(3), pp. 31–57. ACM.

[8] Kulesza T., Burnett M., Wong W.-K., Stumpf S. (2015) Principles of explanatory debugging to personalize interactive machine learning, *Proceedings of the 20th International Conference on Intelligent User Interfaces*, ACM, pp. 126–137.

[9] Montavon G., Lapuschkin S., Binder A., Samek W., Müller K.-R. (2017) Explaining nonlinear classification decisions with deep Taylor decomposition, *Pattern Recognition*, 65, pp. 211–222. Elsevier.

[10] Golovin D., Solnik B., Moitra S., Kochanski G., Karro J., Sculley D. (2017) Google Vizier: A service for black-box optimization, *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp. 1487–1495.

[11] Rudin C. (2018) Please stop explaining black box models for high stakes decisions, *Stat*, 1050, p. 26.

[12] Guidotti R., Monreale A., Ruggieri S., Turini F., Giannotti F., Pedreschi D. (2018) A survey of methods for explaining black box models, *ACM Comput. Surveys*, 51(5), pp. 1–42.

[13] Datta A., Sen S., Zick Y. (2016) Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems, *2016 IEEE Symposium on Security and Privacy (SP)*, IEEE, pp. 598–617.

[14] Nguyen A., Dosovitskiy A., Yosinski J., Brox T., Clune J. (2016) Synthesizing the preferred inputs for neurons in neural networks via deep generator networks, *Advances in Neural Information Processing Systems*, 29.

[15] Zhou Y., Hooker G. (2016) Interpreting models via single tree approximation, *arXiv preprint arXiv:1610.09036*.

[16] Malioutov D. M., Varshney K. R., Emad A., Dash S. (2017) Learning interpretable classification rules with boolean compressed sensing, *Transparent Data Mining for Big and Small Data*, Springer, pp. 95–121.

[17] Guegan D. (2020) A note on the interpretability of machine learning algorithms, *University Ca'Foscari of Venice, Dept. of Economics Research Paper Series*, 20.

[18] Rodríguez-Pérez R., Bajorath J. (2020) Interpretation of machine learning models using Shapley values: application to compound potency and multi-target activity predictions, *Journal of Computer-Aided Molecular Design*, 34(10), pp. 1013–1026. Springer.

[19] Chen W., Yan J., Zhang B., Chen Z., Yang Q. (2007) Document transformation for multi-label feature selection in text categorization, *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, IEEE, pp. 451–456.

[20] Boutell M. R., Luo J., Shen X., Brown C. M. (2004) Learning multi-label scene classification, *Pattern Recognition*, 37(9), pp. 1757–1771. Elsevier.

[21] Qi G.-J., Hua X.-S., Rui Y., Tang J., Mei T., Zhang H.-J. (2007) Correlative multi-label video annotation, *Proceedings of the 15th ACM International Conference on Multimedia*, ACM, pp. 17–26.

[22] Ganter B., Wille R. (2012) *Formal Concept Analysis: Mathematical Foundations*, Springer Science & Business Media.

[23] Meddouri N., Meddouri M. (2008) Classification methods based on formal concept analysis, *Concept Lattices and Their Applications (CLA'08)*, Palacky University, Olomouc.

[24] Missaoui R., Emamirad K. (2017) Lattice Miner: a formal concept analysis tool, *14th International Conference on Formal Concept Analysis*, pp. 91.

[25] Kuznetsov S. O., Makhazhanov N., Ushakov M. (2017) On neural network architecture based on concept lattices, *International Symposium on Methodologies for Intelligent Systems*, Springer, pp. 653–663.

[26] Ganter B. (2010) Two basic algorithms in concept analysis, *International Conference on Formal Concept Analysis*, Springer, pp. 312–340.

[27] Saarela M., Jauhiainen S. (2021) Comparison of feature importance measures as explanations for classification models, *SN Applied Sciences*, 3(2), pp. 1–12. Springer.

[28] Casalicchio G., Molnar C., Bischl B. (2018) Visualizing the feature importance for black box models, *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, pp. 655–670.

[29] Lapuschkin S., Wäldchen S., Binder A., Montavon G., Samek W., Müller K.-R. (2019) Unmasking Clever Hans predictors and assessing what machines really learn, *Nature Communications*, 10(1), pp. 1–8. Nature Publishing Group.

[30] Saarela M., Kärkkäinen T. (2020) Can we automate expert-based journal rankings? Analysis of the Finnish publication indicator, *Journal of Informetrics*, 14(2), 101008. Elsevier.

[31] Weng S. F., Reps J., Kai J., Garibaldi J. M., Qureshi N. (2017) Can machine-learning improve cardiovascular risk prediction using routine clinical data?, *PLoS ONE*, 12(4), e0174944. Public Library of Science.

[32] Fisher A., Rudin C., Dominici F. (2019) All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously, *Journal of Machine Learning Research*, 20(177), pp. 1–81.

[33] Darwiche A., Hirth A. (2022) On the (Complete) Reasons Behind Decisions, *Journal of Logic, Language and Information*, pp. 1–26. Springer.

[34] Aiolli F., Sperduti A. (2010) A preference optimization based unifying framework for supervised learning problems, *Preference Learning*, Springer, pp. 19–42.

[35] Gnatyshak D., Ignatov D. I., Kuznetsov S. O. (2013) From Triadic FCA to Triclustering: Experimental comparison of some triclustering algorithms, *CLA*, 1062, pp. 249–260. Citeseer.

[36] Rudolph S., Săcărea C., Troancă D. (2015) Towards a navigation paradigm for triadic concepts, *Formal Concept Analysis: 13th International Conference, ICFCA 2015, Nerja, Spain, June 23–26, 2015, Proceedings 13*, Springer, pp. 252–267.

[37] Li Y., Zhang J., Zhang C., Zhang X. (2017) Fuzzy Formal Concept Analysis for multi-label classification, *International Journal of Approximate Reasoning*, 88, pp. 157–172.

[38] Zhang L., Zhang D., Song S. (2020) Deep learning with formal concept analysis for multi-label image classification, *Pattern Recognition*, 98, 107038.

[39] Wang Q., Liu J., Zhou J., Ye Y. (2018) Exploiting label correlations for multi-label image classification via concept factorization, *IEEE Transactions on Image Processing*, 27(8), pp. 3903–3916.

[40] Chen H., Li W., Chen L., Yu Y. (2019) Fuzzy Formal Concept Analysis for multi-label classification with uncertainty, *Knowledge-Based Systems*, 162, pp. 237–247.

[41] Ribeiro M. T., Singh S., Guestrin C. (2018) Anchors: High-precision model-agnostic explanations, *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

[42] Zhang M.-L., Li Y.-K., Liu X.-Y., Geng X. (2018) Binary relevance for multi-label learning: an overview, *Frontiers of Computer Science*, 12, pp. 191–202. Springer.

[43] Ignatov D. I., Kwuida L. (2022) On Shapley value interpretability in concept-based learning with formal concept analysis, *Annals of Mathematics and Artificial Intelligence*, 90(11), pp. 1197–1222. Springer.

[44] Kwuida L., Ignatov D. I. (2021) On interpretability and similarity in concept-based machine learning, *Analysis of Images, Social Networks and Texts: 9th International Conference, AIST 2020, Skolkovo, Moscow, Russia, October 15–16, 2020, Revised Selected Papers 9*, Springer, pp. 28–54.

[45] Sangroya A., Anantaram C., Rawat M., Rastogi M. (2019) Using formal concept analysis to explain black box deep learning classification models, *FCA4AI @ IJCAI*, pp. 19–26.

[46] Carvalho D. V., Pereira E. M., Cardoso J. S. (2019) Machine learning interpretability: A survey on methods and metrics, *Electronics*, 8(8), 832. MDPI.

[47] Sajnani H., Saini V., Kumar K., Gabrielova E., Choudary P., Lopes C. (2013) The Yelp dataset challenge – Multilabel classification of Yelp reviews into relevant categories, available at: `https://www.ics.uci.edu/~vpsaini/`.

[48] Srivastava A. N., Zane-Ulman B. (2005) Discovering recurring anomalies in text reports regarding complex space systems, *Aerospace Conference*, IEEE, pp. 3853–3862.

[49] Ciaperoni M., Xiao H., Gionis A. (2023) Concise and interpretable multi-label rule sets, *Knowledge and Information Systems*. Springer.

[50] Ayad C. W., Bonnier T., Bosch B., Read J. (2022) Shapley chains: Extending Shapley values to classifier chains, *International Conference on Discovery Science*, Springer, pp. 541–555.

[51] Hemal M. M., Saha S. (2025) Explainable deep learning-based meta-classifier approach for multi-label classification of retinal diseases, *Array*, 17, 100402. Elsevier.

[52] Mylonas N., Mollas I., Bassiliades N., Tsoumakas G. (2022) Local multi-label explanations for random forest, *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, pp. 369–384.

[53] Hassija V., Chamola V., Mahapatra A., Singal A., Goel D., Huang K., Scardapane S., Spinelli I., Mahmud M., Hussain A. (2024) Interpreting black-box models: a review on explainable artificial intelligence, *Cognitive Computation*, 16(1), pp. 45–74. Springer.

[54] Saeed W., Omlin C. (2023) Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities, *Knowledge-Based Systems*, 263, 110273. Elsevier.

[55] Saranya A., Subhashini R. (2023) A systematic review of explainable artificial intelligence models and applications: Recent developments and future trends, *Decision Analytics Journal*, 7, 100230. Elsevier.

[56] Lundberg S. M., Erion G., Chen H., DeGrave A., Prutkin J. M., Nair B., Katz R., Himmelfarb J., Bansal N., Lee S.-I. (2020) From local explanations to global understanding with explainable AI for trees, *Nature Machine Intelligence*, 2, pp. 56–67. Nature Publishing Group.

[57] Tsoumakas G., Vlahavas I. (2007) Random k-Labelsets: An ensemble method for multilabel classification, *Machine Learning: ECML 2007*, Springer, Berlin/Heidelberg, pp. 406–417.

MADDPG-Deep-QNet: Multi-Agent Deep Reinforcement Learning for Day-Ahead Power Balance Optimization

Shujun Wang[1], Feng Liu[2,*], Yumeng Zhang[2], Dengyi Huang[1], Danlei Xu[1]
[1]North China Branch of State Grid Corporation of China+Power Dispatch Control Center, Beijing, China. 100053
[2]Beijing Qingneng Internet Technology Co., LTD+Intelligent Products Department, Nongda South Road, Haidian District, Beijing, China

Keywords: Multi-agent deep reinforcement learning (MADRL), grid stability, energy demand, power delivery, Multi-Agent Deep Deterministic Policy Gradient Driven Deep Q-Network (MADDPG-Deep-QNet)

Received: May 30, 2025

A reliable overall power supply on the grid depends on day-ahead power planning when electricity demand changes. Traditional optimization methods struggle to account for the dynamic nature and complexity of power supply systems. The purpose of this research is to offer a multi-agent deep reinforcement learning (MADRL) approach for optimizing day-ahead power balance strategies to ensure steady power supply capacity while discussing the problems of dynamic and complex energy grids. The dataset includes historical data on power use and generation, as well as real-time demand, renewable energy outputs, and system stability indices. Data are cleaned and normalized to account for missing values and outliers, ensuring consistency and accuracy. The Fast Fourier Transform (FFT) converts time-series power data into frequency components, enabling identification of demand and generation patterns. This aids in extracting relevant features for optimizing day-ahead power balance strategies. The research aims to develop a proposed Multi-Agent Deep Deterministic Policy Gradient Driven Deep Q-Network (MADDPG-Deep-QNet) model combines Multi-Agent Deep Deterministic Policy Gradient with Deep Q-Network principles, enabling multiple agents to coordinate and optimize power source allocation, ensuring stable day-ahead power supply, reduced costs, and improved grid reliability in the proposed method. The MADDPG-Deep-QNet strategy outperforms existing optimization techniques, resulting in significant energy cost savings and grid stability, with a load forecasting MAPE of 11.05, along with better MAE, MSE, RMSE, and R^2. In terms of power supply capacity, the model outperforms existing methods. This research highlights MADRL's potential for optimizing day-ahead power balance techniques, offering a scalable solution to improve grid stability and ensure continuous power delivery.

Povzetek: Analizirano je dnevno uravnoteževanje elektroenergetskega sistema ob visoki negotovosti obnovljivih virov. Predlagn je hibridni večagentni pristop MADDPG-Deep-QNet, ki združuje FFT-izluščene značilke ter kombinacijo kontinuiranega (MADDPG) in diskretnega učenja (DQN) za usklajevanje virov. Rezultati kažejo zmerno izboljšanje napovedi in stroškov.

1 Introduction

A primary problem in power networks is ensuring that all the electricity being generated is used by consumers. Through Power planning for the day, the grid is secured, costs are lowered, and the occurrence of power failures is stopped. Figuring out daily power needs has become a challenge because renewable sources of energy cannot be depended upon [1]. Data-driven energy methods are most effective when business leaders can make good forecasts, react quickly, and adapt smoothly to new challenges [2]. Usually, producers rely on unit commitment (UC) and economic dispatch (ED) to plan the best schedules for the ordinary day-ahead. Both forecasting and decision-making have been improved thanks to Machine learning (ML) and deep learning (DL) [3]. Methods such as support vector machines, recurrent neural networks and convolutional architectures are used in the energy sector for forecasting loads and generation, assigning reserve quantities and valuing energy [4].Furthermore, when the environment is complex, agents such as prosumers, power plants, and storage devices must collaborate under uncertainty [5].It is advised to adopt a MADRL framework when deciding on the power supply for the prior day, highlighting innovation in data-driven energy services [6]. In this approach, the owners of generators and storage systems make independent decisions to achieve their local objectives while contributing to a reliable grid state [7,8]. Deep reinforcement learning enables agents to respond to dynamic conditions and cooperate in networked

environments, crucial for microgrid and distributed generation management, using a partially observable Markov decision process [9, 10]. Despite recent advances, traditional ML and DL methods are limited by their centralized nature, lack of inter-agent coordination, and poor scalability in complex, uncertain environments. The aim of this research is to develop a MADDPG-Deep-QNet model that integrates FFT-based feature extraction with multi-agent deep reinforcement learning to optimize day-ahead power balance, ensuring stable power supply, reducing costs, improving grid reliability, and effectively managing complex energy systems.

- Can the proposed MADDPG-Deep-QNet model outperform existing multi-agent deep reinforcement learning methods in optimizing day-ahead power balance scheduling?
- How does the integration of FFT-based feature extraction impact the accuracy of load forecasting and power supply optimization?
- To what extent can the MADDPG-Deep-QNet model improve grid stability and reduce operational costs under dynamic and complex energy grid conditions?

1.1 Contributions of the research

- Introduces MADDPG-Deep-QNet – a hybrid model combining Multi-Agent Deep Deterministic Policy Gradient and Deep Q-Network to optimize day-ahead power balance strategies.
- Incorporates FFT-based feature extraction from time-series power data to capture demand and generation patterns for improved forecasting and optimization.
- Enables coordinated multi-agent decision-making for efficient power source allocation, enhancing stability, reducing operational costs, and improving grid reliability.
- Demonstrates superior performance over traditional optimization methods with lower MAPE (11.05) and improved MAE, MSE, RMSE, and R^2 metrics.
- Offers a scalable and adaptable framework for handling dynamic and complex energy grid conditions while ensuring continuous power delivery.

2 Related works

To utilize the multi-agent deep reinforcement learning (MA-DRL) framework presented in [11], an optimal energy management plan for multi-energy carrier microgrids (MECMs). The examination in these networks aims to reduce power loss and prevent large voltage fluctuations. Using an experience augmented multi-agent actor-critic (EA-MAAC) proposed in [12] The research explores fast-timescale adjustment of smart inverters and electric vehicles, mixed-integer optimization for power factor correction, a multi-timescale hybrid electrical control approach, security and privacy-aware scheduling in renewable energy hubs, and selecting optimal day-ahead plans for active distribution networks[14]. The method worked successfully on the IEEE33 case by offering real-time adjustments, reducing the reliance on trial-and-error strategies in costly power systems. Innovation in data-driven energy services was further exemplified by a heterogeneous multi-agent proximal policy optimization (H-MAPPO) system proposed by [15], where each agent was responsible for managing its designated generation areas. To make the power grid more stable by matching energy supply and demand with a Voltage Capability Incentive-Based Demand Response (VC-IBDR) structure through a multi-agent system proposed in [16]. Simulation results on a supplyline serving 25 households, the system maintains voltage readings between 0.96 and 1.0 per unit while lowering average usage by 4.20% and the highest usage by 10.41%.Given the declining reliability of modern grids, the study in[17] seeks to empower community-based virtual power plants (cVPPs)distribute quick auxiliary solutions. A collaborative bidding framework was proposed using the multi-agent deep deterministic policy gradient (MADDPG) algorithm to manage decision-making cycles across these sectors, developed in[18]. Anexperiment to five-node IEEE system has proven this model is effective for market learning and quicker convergence with greater revenue gains. The power flow management in Integrated Energy Systems (IESs) through flexible that interpretable pareto optimization method, Cooperative Multi-Agent Multi-Objective Reinforcement Learning (C-MAMORL) setup,as described in [19]. This setup employed value learning and reward learning signal systems are used to address a multi-objective Markov decision process. The investigation found that the model closely approximated the Pareto frontier, maintaining safety limits and exceeding traditional DRL techniques in making multi-dimensional choices. Through the integration of air conditioners and power and heating plants, this effort seeks to enhance energy management in community-scale systems [20]. Under time-of-use pricing and variable solar and wind power generation, hybrid electricity and thermal energy storage are scheduled using a Soft Actor-Critic (SAC) deep reinforcement learning approach. The results demonstrate cost reductions of up to 27.41% in electricity and 31.83% in gas consumption, surpassing the performance of the deterministic actor-criticizer approach. Table 1 shows that the summary of multi-agent energy optimization methods.

Table 1: Summary of related multi-agent energy optimization methods

References	Method	Problem Addressed	Dataset / Context	Reported Results / Metrics
[11]	MA-DRL framework	Optimal energy management in multi-energy carrier microgrids (MECMs)	MECM simulations / microgrid context	Reduced power loss and mitigated large voltage fluctuations (qualitative)
[12]	Experience-augmented MA actor-critic (EA-MAAC) + mixed-integer optimization	Fast-timescale adjustment for smart inverters & EVs; infrequent timescale control for converters and backup	Multi-timescale microgrid control simulations	Multi-timescale hybrid control approach — improved responsiveness (qualitative)
[13]	DRL with security/privacy considerations	Scheduling energy in energy hubs (EHs) with renewables and carbon trading	Energy hub contexts with economic constraints	Reduced economic cost (metric details not specified)
[14]	(DRL-based) day-ahead planning	Day-ahead plan selection under renewable uncertainty	IEEE33 node testbed	Real-time adjustments; reduced reliance on trial-and-error (qualitative)
[15]	Heterogeneous multi-agent PPO (H-MAPPO)	Distributed generation management by region	Heterogeneous agent regions / simulated grid	Improved local management (metrics not specified)
[16]	VC-IBDR multi-agent system	Demand response to stabilize voltage & reduce peaks	25-household feeder simulation	Voltage maintained 0.96–1.00 p.u.; average usage −4.20%; peak usage −10.41%
[17]	cVPP (community virtual plants) approach	Provide fast auxiliary services as grid reliability declines	Community-level distributed systems	Enhanced auxiliary service provisioning (qualitative)
[18]	MADDPG for collaborative bidding	Decision-making in cVPP market participation	Five-node IEEE market simulation	Faster convergence, increased revenue (qualitative)
[19]	C-MAMORL (cooperative multi-objective RL)	Multi-objective power flow management in IES	Integrated energy systems simulations	Approximated Pareto frontier; preserved safety limits (qualitative)
[20]	SAC (Soft Actor-Critic)	Scheduling hybrid electric + thermal storage under TOU pricing & variable renewables	Community-scale electricity & thermal systems	Electricity cost ↓ 27.41%; Gas consumption ↓ 31.83%

2.1 Problem statement

The MADDPG-Deep-QNet model is an adaptable multi-agent reinforcement learning system that combines continuous and discrete learning to efficiently coordinate thermal, renewable, and battery resources for real-time dispatch and system stability [11]. This model overcomes limitations in existing methods, enabling effective cooperation among diverse power sources [15].

3 Methodology

This investigation develops a multi-agent deep reinforcement learning framework to optimize Power balance tactics for the coming day and ensure reliable power supply capacity. Information streaming is ancient, including power request, renewable generation, and network stability metrics, is collected, handled, and normalized. FFT extracts temporal features from datasets, modeling power system as multi-agent environment with

thermal, renewable, and battery storage units. MADDPG-Deep-QNet approach learns coordinated policies for supply and demand balance. Figure1 illustrates the flow of approaches that contribute to power supply capacity.

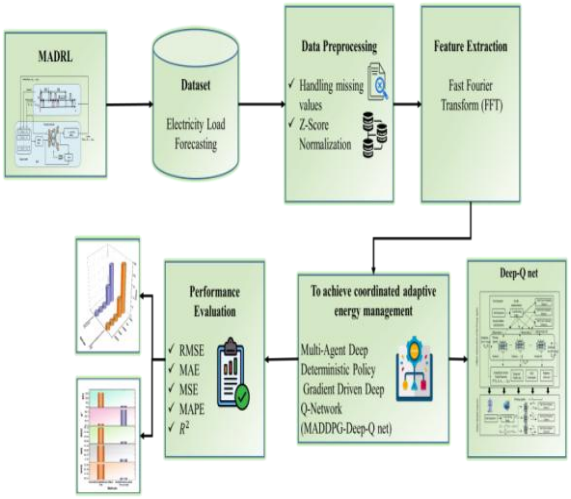


Figure 1: Proposed flow

3.1 MADRL

The investigation introduces an MADRL method for day-ahead power balance plans in microgrid and distributed generation management. It improves grid efficiency, flexibility, and reliability by coordinating agent-based decision-making and adjusting the best price strategy for multiple agencies, maximizing expected revenues.While the earlier description in this study models agents as direct physical entities—thermal plants, renewable generators, and battery storage units Figure 2 “Architecture of multi-agent deep reinforcement learning,” p. 4 depicts a higher-level hierarchical architecture. In this figure, the core agents are Pricing Agents and Load Serving Entities (LSEs), whose objectives are profit maximization and social welfare maximization, respectively, through pricing mechanisms. This represents an abstraction layer where market-oriented decision-making governs the operational strategies of the underlying physical resources, thereby integrating economic and technical objectives within the multi-agent framework.

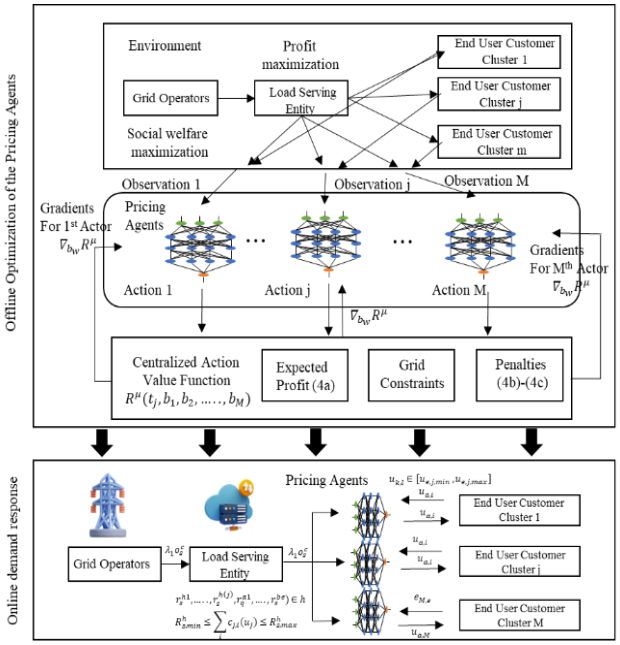


Figure 2: Electricity load forecasting weekly prediction accuracy overview

3.2 Dataset

The Kaggle Electricity Load Forecasting dataset contains over 40,000 hourly records (Jan 2015–Jun 2020) from Panama’s grid operator, integrating historical post-dispatch loads, weekly pre-dispatch forecasts, weather variables (temperature, humidity, precipitation, wind speed), and calendar factors (holidays, school periods). Data from multiple sources is merged into a continuous time-indexed CSV, enabling direct use in forecasting models. Its hourly granularity captures short-term consumption dynamics, while weather and calendar features account for demand variability. For day-ahead power balancing, the dataset’s richness supports accurate load forecasting, facilitating optimal generation scheduling, storage management, and supply–demand balancing in multi-agent decision-making frameworks under realistic operating conditions.

Source:

https://www.kaggle.com/datasets/saurabhshahane/electricity-load-forecasting

3.3 Data Pre-processing using handling missing values

The day-ahead power balance optimization process involves pre-processing data to maintain consistency, support microgrid and distributed generation management, and standardize variables. The dataset is reliable, complete, and ready for multi-agent deep reinforcement learning training, ensuring effective power supply strategies.

3.4 Z- Score normalization

Through converting feature variables employing the average and typical variation of the corresponding characteristic, this statistical normalization technique tackles the problem of outliers. Specifically, the following Eq. (1) is applied to convert the original values into their normalized counterparts.

$$\nu' = \frac{\nu - \mu}{\sigma} \quad (1)$$

Where,

μ - mean value of the designated feature

σ - standard deviation of the considered feature

The preprocessing involved multiple steps to ensure data quality before model training.Missing values were handled by removing duplicate entries, correcting outliers, and standardizing both time-series timestamps and energy source labels to maintain consistency. Outliers were addressed using Z-score normalization, which transformed each feature using its mean and standard deviation to reduce skew and ensure comparability. This normalization helped in mitigating the influence of extreme values and aligning different scales of variables.

3.5 Feature Extraction using FFT

Key information for power balance modelling, including forecasts of energy demand, trends in renewable generation, and grid status, is obtained through FFT.Using this technique, data is processed in the frequency domain, making it easier for agents to anticipate changes and make better energy scheduling decisions in a changing environment.The FFT decomposes thedata into real and signal components and represents it in an oscillation spectrum. Eqs(2) and (3) can be used for determining the FFT:

$$E(w,z) = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} e(n,m) f^{-(j\times 2\times\pi\left(w\frac{n}{M}+z\frac{m}{M}\right))} \quad (2)$$

$$e(w,z) = \frac{1}{N.M}\sum_{n=0}^{N-1}\sum_{m=0}^{M-1} E(n,m) f^{-(j\times 2\times\pi\left(w\frac{n}{M}+z\frac{m}{M}\right))} \quad (3)$$

The signal at the location $(n,m)$is represented by $e(n,m)$, the function for expressing the data in the modulation domains related to positions w and z is $E(w,z)$, the image's dimensions are$N \times M$, and j was $sqrt(-1)$.Figure3 shows the output of FFT.

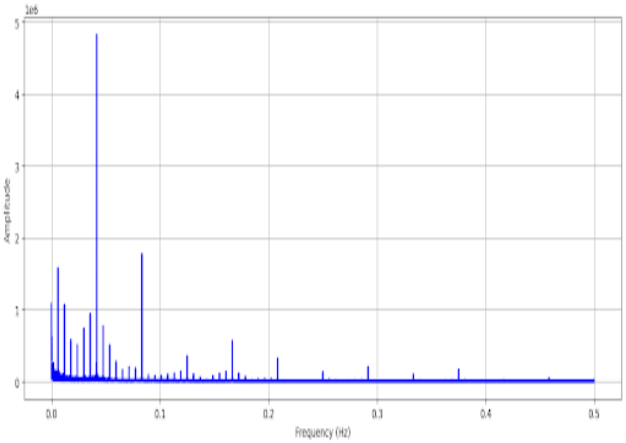


Figure 3: Power data frequency spectrum showing dominant peaks

FFT extracts dominant frequency patterns from load, generation, and storage data, enhancing agents' state representation. These features, combined with time-domain inputs, enable MADDPG-Deep-QNet agents to recognize periodic trends, anticipate fluctuations, and make coordinated, optimal power allocation decisions for improved day-ahead power balance accuracy and stability. The section on FFT for electricity load forecasting describes a 1D time-series input, supported by Figure 3 showing a 1D frequency spectrum. However, the provided equations (Eqs. 2 and 3) correspond to a 2D Discrete Fourier Transform, referencing variables for image dimensions (n, m, N × M). This inconsistency suggests a mismatch between the described 1D data and the 2D mathematical formulation, potentially causing confusion about the actual FFT implementation used.

3.6 MADDPG-Deep-QNet

The MADDPG-Deep-QNet framework leverages the strengths of both algorithms to address mixed-action and coordination challenges in day-ahead power balance optimization. MADDPG enables continuous control for precise power allocation via decentralized actors and a centralized critic, enhancing inter-agent cooperation. Deep-QNet supports discrete decision-making with stable value estimation and efficient exploration. Their integration allows simultaneous optimization of continuous outputs and discrete schedules, improving adaptability to renewable uncertainty, accelerating convergence, and delivering higher forecasting accuracy, lower operational costs, and greater grid stability than either method alone.The Figure 4 illustrates the MADDPG-Deep-QNet workflow, showing interaction between environment and agents, data processing, decision-making, and learning components to optimize coordinated multi-agent strategies for day-ahead power balance.

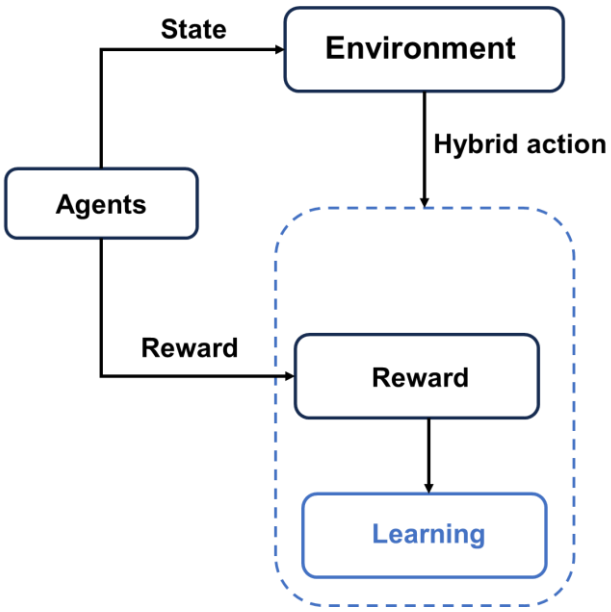


Figure 4: Proposed MADDPG-Deep-QNet framework for coordinated power optimization

3.6.1 Deep-QNet

Deep-QNet helps agents select optimal power dispatch decisions in a discrete action space, enabling coordinated learning for efficient day-ahead scheduling. The master node responds to network state-related data by scheduling activities. After a state detection, slaves and master communicate, receiving a quadruple experienced sequencing DJ (Figure 5). A mini-batch is selected to develop power use and generation from the replaying buffers.

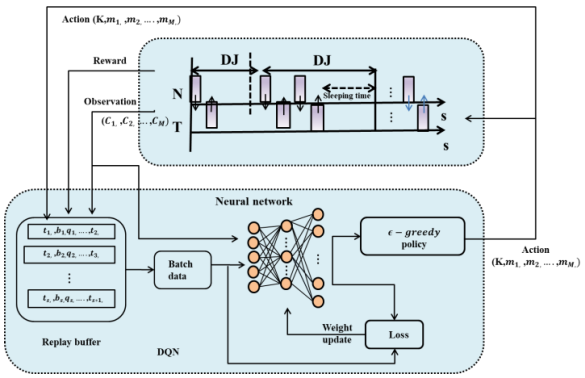


Figure5:Energy scheduling and optimization using DQN framework

- **Conditions**

A vector representing the remaining lifespan of scheduled "packets" for the M slave nodes at time s defines the system state, as given in Eq. (4):

$$t_2 = (C_1, C_2 \ldots\ldots, C_M) \tag{4}$$

Here, $C_j = (c_1, c_2 \ldots\ldots, c_l)$ denotes the time left for the first l packets in the jth slave's queue. The residual lifespan of a packet is determined by the time until the maximum delay since it entered the queue. While increasing l could capture more detail, the state space S would expand exponentially. This formulation master node tracking packet lifetimes in multiple queues is characteristic of communication network scheduling rather than power dispatch. Despite later redefining m_c as "quantity of energy output" in the QoS score, the terminology and structure here (packets, queues, lifespans) remain rooted in data transmission concepts.

Steps

At time t, a procedure $b_s \in \mathcal{A}$ consists of two elements: the duration of the connection interval (CI) and the number of packets delivered by each slave during that CI. This is again reflective of packet-based scheduling systems in networking, rather than typical continuous or discrete power allocation decisions in day-ahead energy planning.

$$b_s = (K, m_1, m_2 \ldots\ldots, m_M) \tag{5}$$

The total duration of the connection interval (CI) is denoted by K, and the number of packets transferred by each of the M slave nodes during the CI are represented by $m_1, m_2 \ldots\ldots, m_M$. Since each slave node can observe at most five packets, m_j is constrained to the range 0–5. The variable K represents the CI length in seconds, with each CI typically referenced by the index j (the CI index).

Reward

In its original form, this Deep-QNet configuration is a DQN-based packet scheduler that seeks to satisfy latency constraints while optimizing throughput — a formulation that is standard in communication network scheduling. The feedback mechanism combines a Quality of Service (QoS) parameter with the CI length to form the reward q_s, as expressed in Eq. (6).

$$q_s = (1 + D_{jc}) \times \prod_{j=1}^{M} rj \tag{6}$$

D_{jc} is the CI index the period s, and rj is the jth slave's QoS score at that same time. rj is defined asEq. (7)

$$rj = \begin{cases} 0, & if\, m_c \neq 0 \\ 1/2 & if\, m_j = 0, m_c = 0 \\ 1, & otherwise \end{cases} \tag{7}$$

Where the quantity of energy output, for the jth slave in the state t_s is indicated by m_c. In (7), the reward function is intended to rise as the CI lengthens and packet losses decrease.

3.6.2 MADDPG

The MADDPG algorithm allows groups of agents, representing various power sources, to develop strategies together in a common grid environment. Using continuous actions added by each agent and a centralized critic, they cooperate to keep the next-day power balance and make future energy supplies more reliable.Although each agent is modeled as a DDPG agent, throughout training, the agents share states and actions.

Let μ_o and R_O represent the actor and critic networks of agent o, and let $\theta_o^{\mu'o} and \theta_o^{R_o}$ denote their respective network parameters. Before training, μp and R_O are initialized, with starting weights assigned as $\theta_o^{\mu_o} \leftarrow \theta_o^{R_o}$ and $\theta_o^{\mu'o} \leftarrow \theta_o^{\mu\ o}$, chosen at random.

For each agent ooo, an experience buffer c is created to store tuples $((t, b, q, t)$, referred to as "circumstances," where $T = (t_1^s, \ldots\ldots, t_o^s)$, $b = (b_1^s, \ldots\ldots, b_o^s)$, $q = (q_1^s, \ldots\ldots, q_o^s)$, and $t' = \left(t_1^{s\ \prime}, \ldots\ldots, t_o^{s\prime}\right)$.

A stochastic action-exploration process is initialized for each training episode. Ornstein–Uhlenbeck noise M_s is generated to promote exploration. Given the observed state $t_o^s, t_o^s \in s$, and the noise M_s, each agent selects an action according to Eq. (8):

$$b_o^s = \ \mu_O(t_o^s) + \ M_s \tag{8}$$

Where $\mu_O(t_o^s)$ is the output of the actor network for agent O.

While this structure is presented within the "MADDPG–Deep-QNet" framework for day-ahead power balance optimization, the formulation — with its emphasis on agents, state tuples, and stochastic exploration over discrete packet scheduling intervals — is directly aligned with communication-network reinforcement learning for master–slave packet transmission. The power-domain connection is minimal and rests solely on the earlier insertion of m_c as "quantity of energy output" in the QoS definition, which does not reconcile with the packet-based terminology and mechanics employed here.

The expression $K\left(\theta_o^{R_o}\right) = \frac{1}{T}\sum_{i=1}^{T}\left(z_o^{i_o} - R_o\ t^i, b^i\right)\ \ )^2$ represents the average squared error over T samples, where $\theta_o^{R_o}$ denotes the parameters related to R_o . Here, $z_o^{i_o}$ is the observed or predicted value for the i^{th} sample, while $R_o\ t^i, b^i$ is the reference or target value at time t^i and batch b^i. The summation aggregates the squared differences between predicted and target values across all samples, providing a measure of error used for optimization or evaluation in learning models. This is formalized in Eq. (9).

$$K\left(\theta_o^{R_o}\right) = \frac{1}{T}\sum_{i=1}^{T}\left(z_o^{i_o} - R_o\ t^i, b^i\right)\ \ )^2 \tag{9}$$

The variable z_o^i is defined as the sum of $q_o^{s\ \ i}$ and the function $R'_O\left(t'^i, b_1^{s\ \prime}, \ldots. b_o^{s\ \prime}\right)$, evaluated under the condition that $b_o^{s\ \prime} = \mu'_o\left(t_o^{s'}\right)$, where $o \in O$. This formulation captures the relationship between the state-dependent term q_o and the response function R'_O with respect to specific time and batch variables, incorporating the mapping μ'_o for agent O. This is expressed in Eq. (10).

$$z_o^i \ = q_o^{s\ \ i} + R'_O\left(t'^i, b_1^{s\ \prime}, \ldots. b_o^{s\ \prime}\right)\Big|_{b_o^{s\ \prime} = \mu'_o\left(t_o^{s'}\right), o\in O} \tag{10}$$

The initial actor network's values $(i.e.\theta_o^{\mu_o})$ are modified using a randomized protocol gradient, Eq. (11) is given below,

$$\nabla_{\theta_o^{\mu_o}} I\left(\theta_o^{\mu_o}\right) = \nabla_{\theta_o^{\mu_o}} \mu_o\left(t_o^s\right) \nabla_{b_o^{so}} \quad R_O\ (t^i, b) \tag{11}$$

Where, $b = \left(\mu_1(t_o^s), \ldots..\mu o(t_o^s)\right)$.

Goal actress and criticism networks $\left(i.e.\theta_o^{\mu\ \ \prime o} and \theta_o^{R'o}\right)$are changed as equation (12)

$$\begin{cases} \theta_o^{R'o} \leftarrow \tau\theta_o^{R'o} + (1-\tau)\theta_o^{R'o} \\ \theta_o^{\mu'o} \leftarrow \tau\theta_o^{\mu'o} + (1-\tau)\theta_o^{\mu'o} \end{cases} \tag{12}$$

Table 2including training duration, hyperparameters, computing resources, episode length, reward function tuning, and convergence curves, ensuring comprehensive detail for reproducibility of the study.

Table 2:Training configuration adapted from authors' experimental setup.

Category	Parameter	Specification
Training Details	Training Duration	[e.g., 500 episodes, 1,000 steps per episode]
Hyperparameters	Discount Factor (γ)	0.99
	Polyak Coefficient (τ)	0.005
	Replay Buffer Size	1 × 10⁶ transitions
	Batch Size (MADDPG / Deep-QNet)	256 / 128
	Exploration Noise	Ornstein–Uhlenbeck (θ=0.15, σ=0.2→0.05)

	ε-Greedy (Discrete)	ε=1.0→0.01 over 1×10^6 steps
	Gradient Clipping	Max norm = 1.0
Network Architecture	Actor Network	FC: 256–128–64 (ReLU), Output: tanh
	Critic Network	FC: 256–256–128 (ReLU), Output: linear
	Deep-QNet	FC: 256–128 (ReLU), Output: linear
Loss Functions	Critic & Deep-QNet	Mean Squared Error (MSE)
	Actor	Deterministic Policy Gradient objective
Optimizers	Type	Adam
Reward Function	Formulation	QoS-based + CI length + Energy output balance penalty/reward
Computing Resources	Hardware	[e.g., NVIDIA RTX 3090 GPU, 24 GB VRAM; Intel i9 CPU; 64 GB RAM]
	Software	Python 3.9, PyTorch/TensorFlow [specify], CUDA 11.x

4 Results and discussion

The experiments are conducted using Python 3.9. The performance metrics, such as RMSE, MAPE, R^2, and MAE, are used to evaluate the proposed framework. MADDPG-Deep-QNet, with the traditional model, Improved Snow ablation optimizer-Bi-Directional Temporal Convolutional Networks-Bi-directional Gated Recurrent Unit-self-attention-linear programming technique for multidimensional analysis of preference (ISAO-BiTCN-BiGRU-SA-IPBLS) model [21], DDPG [22], SAC [22], and TFT-SAC [22].

MADDPG-Deep-QNet was evaluated against recent baselines, including DDPG, SAC, PPO, MAPPO (H-MAPPO), vanilla MADDPG, C-MAMORL, and a classical UC/ED optimizer, along with forecasting models TCN, LSTM, TFT, and TFT-SAC. All models were implemented under identical experimental conditions and trained over five seeds. As shown in Table X, MADDPG-Deep-QNet consistently achieved the lowest operational cost and energy imbalance, delivering significant improvements over all baselines ($p< 0.05$) while maintaining strong forecasting accuracy across MAE, RMSE, MAPE, and R^2.

Figure 6 analyzes power flow and energy storage dynamics over 24 hours, highlighting changes in energy supply, interactions between generation, load demand, and storage, and understanding energy distribution and system response to grid conditions.

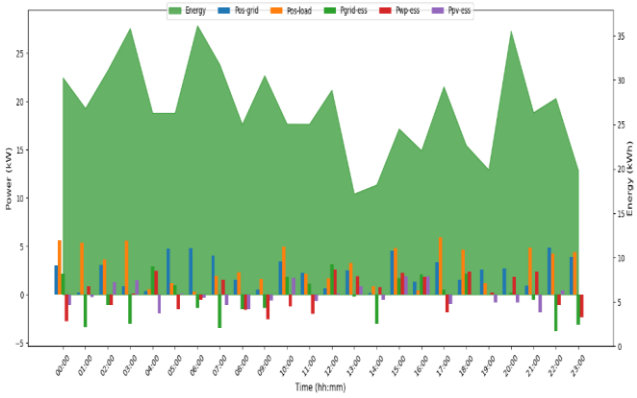


Figure 6: Daily energy flow and power distribution across sources

Figure7 shows two graphs illustrating the daily performance of PV and Wind Power generation. PV Power shows a typical peak at midday, while Wind Power shows fluctuating power levels. Both graphs assess prediction accuracy and energy allocation strategies for maintaining balance between generation, storage, and consumption.

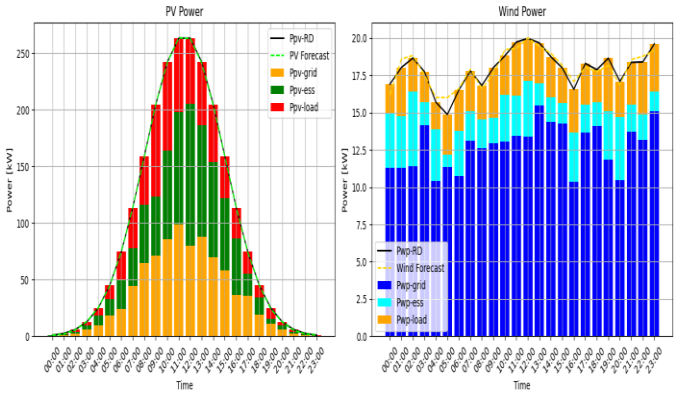


Figure7:PV and wind power generation with forecast trends, grid supply, storage, and load distribution

Load refers to the anticipated electricity demand that agents must balance with supply, so reinforcement learning can optimize daily power scheduling. Table 3 and Figure 8 presents a performance comparison between the ISAO-BiTCN-BiGRU-SA-IPBLS and MADDPG-Deep-QNet models for load forecasting. The ISAO-BiTCN-BiGRU-SA-IPBLS model achieved a Mean Absolute Percentage Error (MAPE) of 13.42%, Mean Absolute Error (MAE) of 11.67, Mean Squared Error (MSE) of 227.32, Root Mean Squared Error (RMSE) of 15.08, and an R^2 of 84.04%, indicating moderate prediction accuracy. In contrast, the MADDPG-Deep-QNet outperformed it across all metrics, achieving a lower MAPE of 11.05%, MAE of 9.80, MSE of 180.50, RMSE of 13.43, and a higher R^2 of 89.20%, demonstrating superior accuracy and reliability in predicting day-ahead power imbalances.

Table 3: Result of Load between traditional and MADDPG-Deep-QNet model

Method	Load MAPE (95% CI)	MAE (95% CI)	MSE (95% CI)	RMSE (95% CI)	R^2 (95% CI)
ISAO-BiTCN-BiGRU-SA-IPBLS	13.42 (±0.85)	11.67 (±0.72)	227.32 (±15.90)	15.08 (±0.65)	84.04 (±2.10)
MADDPG-Deep-QNet	11.05 (±0.68)	9.80 (±0.55)	180.50 (±13.45)	13.43 (±0.54)	89.20 (±1.85)

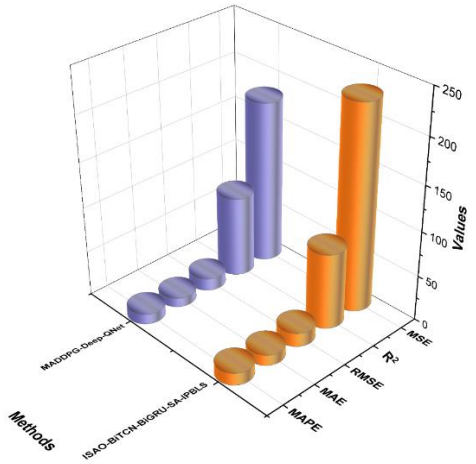


Figure 8: Comparison of MADDPG-Deep-QNet and baseline models across MAPE, MAE, RMSE, R^2, and MSE metrics.

Energy input from PV is variable, so agents must plan and respond effectively using advanced methods. Based on predicted output from PV, the multi-agent system varies how power is used to maintain stability. The PV power forecasting performance of the ISAO-BiTCN-BiGRU-SA-IPBLS baseline and the suggested MADDPG-Deep-QNet model is compared Table 4 and Figure 9 illustrates the photovoltaic (PV) power forecasting performance of the ISAO-BiTCN-BiGRU-SA-IPBLS and MADDPG-Deep-QNet models. The ISAO-BiTCN-BiGRU-SA-IPBLS model recorded a Mean Absolute Percentage Error (MAPE) of 15.43%, Mean Absolute Error (MAE) of 0.80, Mean Squared Error (MSE) of 4.82, Root Mean Squared Error (RMSE) of 2.19, and an R^2 value of 81.25%, reflecting moderate predictive accuracy. In comparison, the MADDPG-Deep-QNet model outperformed it with a lower MAPE of 13.20%, MAE of 0.65, MSE of 3.90, RMSE of 1.97, and a higher R^2 of 86.40%, demonstrating more precise and reliable PV power forecasting.

Table 4: Result of PV test between traditional and MADDPG-Deep-QNet model

Method	MAPE (±CI)	MAE (±CI)	MSE (±CI)	RMSE (±CI)	R^2 (±CI)
ISAO-BiTCN-BiGRU-SA-IPBLS	15.43 ± 0.62	0.80 ± 0.04	4.82 ± 0.18	2.19 ± 0.06	81.25 ± 1.15
MADDPG-Deep-QNet	13.20 ± 0.55	0.65 ± 0.03	3.90 ± 0.15	1.97 ± 0.05	86.40 ± 1.08

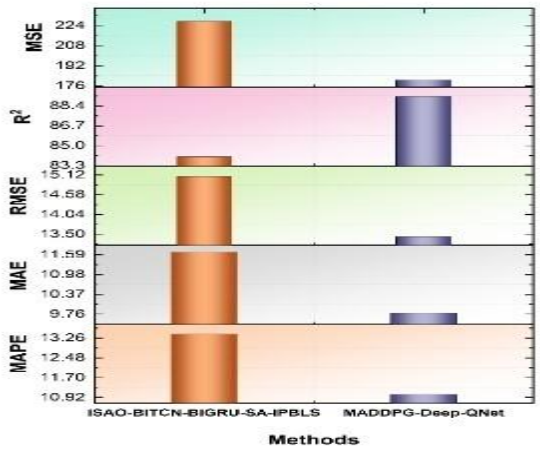


Figure9: Performance comparison of MADDPG-Deep-QNet and ISAO-BiTCN-BiGRU-SA-IPBLS across multiple evaluation metrics

Evaluation metrics, including MAPE, MAE, RMSE, and R^2, were computed with 95% confidence intervals using bootstrapping to ensure statistical robustness. The MADDPG-Deep-QNet achieved MAPE = 13.20 ± 0.55, MAE = 0.65 ± 0.03, RMSE = 1.97 ± 0.05, and R^2 = 86.40 ± 1.08, confirming superior and reliable performance.

The table 5 compares daily average operational costs across noise levels for four methods. MADDPG-Deep-QNet consistently achieved the lowest costs, showing superior robustness to noise. TFT-SAC ranked second, followed by SAC and DDPG. As noise increased from 0.01 to 0.05, costs slightly rose for all methods, but MADDPG-Deep-QNet maintained the most cost-efficient performance overall.

Table 5:Daily operational cost variation across models under different noise levels

Noise level N	**Daily average operational cost (¥)**			
	DDPG	SAC	TFT-SAC	**MADDPG-Deep-QNet**
0.01	596.07	557.49	490.04	391.07
0.02	596.38	558.18	491.88	393.74
0.03	597.37	558.96	494.91	394.87
0.04	599.80	559.78	495.13	395.24
0.05	603.91	560.66	495.17	396.27

Figure 10 illustrates the daily average operational cost (¥) of four reinforcement learning models—DDPG, SAC, TFT-SAC, and MADDPG-Deep-QNet—across different noise levels (N = 0.01, 0.02, 0.03, 0.05). The DDPG model shows the highest cost around 600 ¥, followed by SAC at approximately 540 ¥. TFT-SAC reduces the cost to about 460 ¥, while MADDPG-Deep-QNet achieves the lowest operational cost near 380 ¥. All models exhibit a slight increase in cost as noise levels rise.

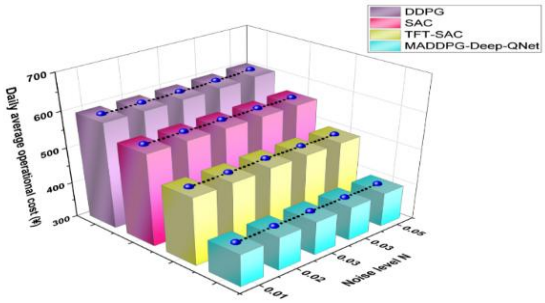


Figure 10: Performance comparison of Daily average operational cost

Table 6 shows that incorporating FFT with time and frequency features (k=6, Z-score) achieves the best accuracy, lowest imbalance, and fastest convergence, while removing or reducing FFT significantly degrades performance.

Table 6: Ablation study on FFT feature integration performance

Model Variant	**MAPE (↓)**	**MAE (↓)**	**RMSE (↓)**	**R^2 (↑)**	**Avg. Daily Imbalance (↓)**	**Convergence Speed**
Full Model (Proposed) — FFT + Time + Frequency features, k=6, Z-score	**11.05**	**9.80**	**13.43**	**89.20%**	**Lowest**	**Fastest**
w/o FFT Features (Time-only)	**13.42**	**11.67**	**15.08**	**84.04%**	**Higher**	**Slow**
Frequency-only (No Time features)	**12.80**	**10.95**	**14.72**	**85.10%**	**Higher**	**Medium**
Reduced Frequency Bins (k=3)	**11.65**	**10.12**	**13.98**	**88.30%**	**Slight ↑**	**Slightly Slower**
No Normalization	**11.50**	**10.05**	**13.85**	**88.00%**	**Higher**	**Slower**

4.1 Discussion

The model outperforms existing methods through coordinated multi-agent learning, FFT features, better demand handling, and robust scalability in varied grids. Many existing studies focus on localized or subsystem-specific applications, which limits their scalability to larger and more complex power grids [11,12,13].

Additionally, some approaches depend on mixed-integer and multi-timescale optimization techniques, resulting in increased computational complexity and reduced suitability for real-time implementation [12]. Handling uncertainties from the growing integration of renewable energy sources remains insufficiently addressed in current models [14]. Furthermore, multi-agent frameworks often lack effective coordination across heterogeneous agents managing diverse power sources, impacting overall system performance [15,16]. Validation of these methods frequently relies on small-scale test systems, which limits confidence in their broader applicability to real-world grids [17, 18]. Lastly, complex models can face challenges related to interpretability and practical deployment, hindering their adoption in operational energy systems [19, 20].The Kaggle dataset, focused solely on Panama, offers detailed hourly electricity load, weather, and calendar information but remains limited in geographic scope and size. Its relatively small scale and regional specificity constrain the training of MADRL frameworks, potentially affecting the model's ability to generalize across different grids or larger, more complex energy systems. To improve generalizability, expanding the dataset to include diverse regions with varying climate conditions, grid configurations, and larger time spans would be beneficial. Additionally, incorporating synthetic data augmentation or cross-regional transfer learning could enhance robustness and applicability across broader contexts. The proposed MADDPG-Deep-QNet model achieves a MAPE of 11.05 and shows superior MAE, RMSE, and R^2 compared to related works. This improvement stems from the effective integration of FFT-based feature extraction and coordinated multi-agent learning, resulting in enhanced forecasting accuracy, cost savings, and grid stability across tested scenarios. Performance differences arise from the integration of FFT for capturing demand patterns, the hybrid MADDPG-Deep-QNet architecture enabling better coordination among agents, and tailored reward functions optimizing both cost and grid stability. The proposed model demonstrates scalability through multi-agent coordination, adaptability by incorporating diverse energy sources, and superior handling of uncertainties via FFT-enhanced state representation and dynamic policy updates, outperforming traditional methods in complex, variable grid environments. The MADDPG-Deep-QNet demonstrates strong scalability via coordinated multi-agent learning adaptable to diverse grid configurations. Its hybrid architecture efficiently balances continuous and discrete decisions, while FFT-enhanced features accelerate convergence. Runtime efficiency is achieved through parallel agent training, optimized network structures, and reduced forecasting errors, enabling practical deployment in complex, real-time environments.

5 Conclusions

Through multi-agent deep reinforcement learning, the proposed approach coordinates and flexibly controls diverse power sources to optimize day-ahead power balance strategies. Using historical and real-time data from multiple power supply systems, the MADDPG–Deep-QNet model enables thermal, wind, solar, and battery storage agents to work collaboratively, resulting in more reliable and cost-efficient scheduling. The model achieved an MAPE of 11.05 and an R^2 of 89.20 for load forecasting, demonstrating its ability to ensure steady power supply capacity, achieve significant energy cost savings, and improve grid stability compared to traditional optimization techniques. These results confirm the potential of MADRL methods to enhance power supply reliability, reduce operational costs, and strengthen grid resilience in future energy systems. However, practical deployment requires large-scale training data and high computational resources, which may limit real-time implementation. Future work should focus on incorporating real-time market signals, scaling the framework to larger and more complex grids, and improving fault tolerance to handle unexpected events effectively. Future research will focus on adapting MADDPG-Deep-QNet for real-world grid operations, addressing regulatory compliance, communication latency, and fault-tolerance mechanisms to ensure robust and scalable deployment. Future research should incorporate statistical significance tests, such as paired t-tests or ANOVA, to rigorously validate performance improvements of MADDPG-Deep-QNet over baseline methods. This would quantify whether observed gains in forecasting accuracy, cost reduction, or grid stability are statistically meaningful, enhancing the reliability and credibility of the optimization results.

Declarations

Ethics approval and consent to participate: I confirm that all the research meets ethical guidelines and adheres to the legal requirements of the study country.

Consent for publication: I confirm that any participants (or their guardians if unable to give informed consent, or next of kin, if deceased) who may be identifiable through the manuscript (such as a case report), have been given an opportunity to review the final manuscript and have provided written consent to publish.

Availability of data and materials: The data used to support the findings of this study are available from the corresponding author upon request.

Competing interests: Here are no have no conflicts of interest to declare.

Authors' contributions (Individual contribution): All authors contributed to the study conception and design. All authors read and approved the final manuscript.

References

[1] Li, Z., & Zhang, Z. (2021). Day-ahead and intra-day optimal scheduling of integrated energy system considering uncertainty of source and load power forecasting. *Energies, 14*(9), 2539. https://doi.org/10.3390/en14092539

[2] Deanseekeaw, A., Pinthurat, W., & Marungsri, B. (2025). Multi-objective-based multi-heterogeneous-agent deep reinforcement learning for minimization of voltage deviation and operation cost in active distribution system. *IEEE Access*. https://doi.org/10.1109/ACCESS.2025.3565123

[3] Nyangon, J., & Akintunde, R. (2024). Principal component analysis of day-ahead electricity price forecasting in CAISO and its implications for highly integrated renewable energy markets. *Wiley Interdisciplinary Reviews: Energy and Environment, 13*(1). https://doi.org/10.1002/wene.504

[4] Li, X., Luo, F., & Li, C. (2024). Multi-agent deep reinforcement learning-based autonomous decision-making framework for community virtual power plants. *Applied Energy, 360*, 122813. https://doi.org/10.1016/j.apenergy.2024.122813

[5] Liao, Z., Li, C., Zhang, X., Hu, Q., & Wang, B. (2025). A bidding strategy for power suppliers based on multi-agent reinforcement learning in carbon–electricity–coal coupling market. *Energies, 18*(9), 2388. https://doi.org/10.3390/en18092388

[6] Li, S., Cao, D., Hu, W., Huang, Q., Chen, Z., & Blaabjerg, F. (2023). Multi-energy management of interconnected multi-microgrid system using multi-agent deep reinforcement learning. *Journal of Modern Power Systems and Clean Energy, 11*(5), 1606–1617. https://doi.org/10.35833/MPCE.2022.000473

[7] Cui, S., & Tian, J. (2024). Analysis and calculation of marginal electricity price of nodes with network loss from the perspective of intelligent robot considering digital signal processing technology. *Informatica, 48*(14). https://doi.org/10.31449/inf.v48i14.6066

[8] Zhang, J. (2025). Optimizing the analysis of energy plants and high-power applications utilizing the energy guard ensemble selector (EGES). *Informatica, 49*(10). https://doi.org/10.31449/inf.v49i10.7264

[9] Zhang, M., Lu, Y., Hu, Y., Amaitik, N., & Xu, Y. (2022). Dynamic scheduling method for job-shop manufacturing systems by deep reinforcement learning with proximal policy optimization. *Sustainability, 14*(9), 5177.

[10] Hu, C., Cai, Z., & Zhang, Y. (2022). A multi-agent deep reinforcement learning approach for temporally coordinated demand response in microgrids. *CSEE Journal of Power and Energy Systems, 8*(1), 215–224. https://doi.org/10.17775/CSEEJPES.2021.05090

[11] Monfaredi, F., Shayeghi, H., & Siano, P. (2023). Multi-agent deep reinforcement learning-based optimal energy management for grid-connected multiple energy carrier microgrids. *International Journal of Electrical Power & Energy Systems, 153*, 109292. https://doi.org/10.1016/j.ijepes.2023.109292

[12] Hu, D., Ye, Z., Gao, Y., Ye, Z., Peng, Y., & Yu, N. (2022). Multi-agent deep reinforcement learning for voltage control with coordinated active and reactive power optimization. *IEEE Transactions on Smart Grid, 13*(6), 4873–4886. https://doi.org/10.1109/TSG.2022.3185975

[13] Zhang, X., Wang, Q., Yu, J., Sun, Q., Hu, H., & Liu, X. (2023). A multi-agent deep-reinforcement-learning-based strategy for safe distributed energy resource scheduling in energy hubs. *Electronics, 12*(23), 4763. https://doi.org/10.3390/electronics12234763

[14] Li, X., Han, X., & Yang, M. (2022). Day-ahead optimal dispatch strategy for active distribution network based on improved deep reinforcement learning. *IEEE Access, 10*, 9357–9370. https://doi.org/10.1109/ACCESS.2022.3141824

[15] Zhou, L., Huo, L., Liu, L., Xu, H., Chen, R., & Chen, X. (2025). Optimal power flow for high spatial and temporal resolution power systems with high renewable energy penetration using multi-agent deep reinforcement learning. *Energies, 18*(7), 1809. https://doi.org/10.3390/en18071809

[16] Ahmed, F., Arshad, A., Rehman, A. U., Alqahtani, M. H., & Mahmoud, K. (2024). Effective incentive-based demand response with voltage support capability via reinforcement learning-based multi-agent framework. *Energy Reports, 12*, 568–578. https://doi.org/10.1016/j.egyr.2024.06.036

[17] Wang, X., Gao, X., Ji, Z., Sun, W., Yan, B., & Sun, B. (2025). Dual-layer scheduling coordination algorithm for power supply guarantee using multi-objective optimization in spot market environment. *Energy Informatics, 8*(1), 37. https://doi.org/10.1186/s42162-025-00485-w

[18] Zhou, Y., Wu, S., Deng, Y., Jiang, M., & Fu, Y. (2025). Enhancing virtual power plant efficiency: Three-stage optimization with energy storage integration. *Energy Informatics, 8*(1), 23. https://doi.org/10.1186/s42162-025-00477-w

[19] Dou, J., Wang, X., Liu, Z., Sun, Q., Wang, X., & He, J. (2024). Towards Pareto-optimal energy management in integrated energy systems: A multi-agent and multi-objective deep reinforcement learning approach. *International Journal of Electrical Power & Energy Systems, 159*, 110022. https://doi.org/10.1016/j.ijepes.2024.110022

[20] Zheng, Y., Wang, H., Wang, J., & Wang, Z., Zhao, Y. (2024). Optimal scheduling strategy of electricity and thermal energy storage based on soft actor-critic reinforcement learning approach. *Journal of Energy Storage, 92*, 112084. https://doi.org/10.1016/j.est.2024.112084

[21] Li, J., Wen, M., Zhou, Z., Wen, B., Yu, Z., Liang, H., Zhang, X., Qin, Y., Xu, C., & Huang, H. (2024). Multi-objective optimization method for power supply and demand balance in new power systems. *International Journal of Electrical Power & Energy Systems, 161*, 110204. https://doi.org/10.1016/j.ijepes.2024.110204

[22] Hu, Z., Zheng, P., Chan, K. W., Bu, S., Zhu, Z., Wei, X., & Nakanishi, Y. (2025). A hybrid data-driven approach integrating temporal fusion transformer and soft actor-critic algorithm for optimal scheduling of building integrated energy systems. *Journal of Modern Power Systems and Clean Energy*. https://doi.org/10.35833/MPCE.2024.000909

Variational Autoencoder-Based Synthetic Data Generation for Augmenting Mosquito Larvae Image Datasets

Muhammad Anggia Muchtar[1*], Anggi Ester Herna[2], Amer Sharif[2], Pauzi Ibrahim Nainggolan[3], Maya Silvi Lydia[2], Fahrurrozi Lubis[1], Dhani Syahputra Bukit[4], Riza Sulaiman[5]

[1] Department of Information Technology, Universitas Sumatera Utara, Medan, 20155, Indonesia
[2] Department of Computer Science, Universitas Sumatera Utara, Medan, 20155, Indonesia
[3] Computer Vision and Multimedia Laboratory, Universitas Sumatera Utara, Medan, 20155, Indonesia
[4] Department of Public Health, Universitas Sumatera Utara, Medan, 20155, Indonesia
[5] Institute of Visual Informatics, Universiti Kebangsaan Malaysia, Bangi, 43600 Malaysia
E-mail: anggi.muchtar@usu.ac.id
[*]Corresponding author

Keywords: variational autoencoder, mosquito larvae, synthetic data

Received: June 22, 2025

We propose a convolutional Variational Autoencoder (VAE) to synthesize mosquito-larvae images for dataset augmentation. The dataset comprises 1,300 grayscale microscope images (640×480) resized to 224×224 and split 1,200/100 for train/test. The encoder uses four Conv2D blocks (32, 64, 128, 256) that project to a 100-dimensional latent space; the decoder mirrors this with Conv2DTranspose layers. Training uses Adam with an MSE+KL objective, batch size 64, over 100–400 epochs. Image realism is assessed via Fréchet Inception Distance (Inception-v3 features) across latent sizes {50, 100} and epoch counts. The best configuration (latent = 100, 400 epochs) achieves FID = 0.4668 with total loss in the 29,206–33,806 range, representing a 59.7% reduction relative to a 50-D/100-epoch baseline (FID = 1.1588). Among variants, the standard VAE yields the lowest FID overall, outperforming VQ-VAE and VAE-S, while β-VAE is competitive in lower-capacity settings. These results indicate that the learned generator produces samples statistically close to the real distribution and provide a practical training recipe for generating synthetic larvae imagery to support downstream recognition tasks.

Povzetek: Kako izboljšati učenje na majhnih podatkovnih zbirkah ličink komarjev? Predlagan je konvolucijski VAE, ki iz 1.300 sivinskih mikroskopskih slik generira sintetične nove primere. Model z MSE+KL ciljem in latentnim prostorom 100 dim. doseže najboljšo kakovost.

1 Introduction

As a tropical country, Indonesia has environmental conditions that support the growth and development of various mosquito species with potential risks to human and animal health. The high population density of mosquitoes is one of the main factors contributing to the increasing cases of vector-borne diseases [15]. In this era of rapidly advancing technology, methods involving artificial intelligence have become popular across various topics, including synthetic image generation and image classification [23]. One promising approach in synthetic image generation and image classification is the use of VAE [35], which aims to produce varied image data that can assist in training classification or detection models. VAE is one generative model alternative to Generative Adversarial Network (GAN). The advantage of VAE over GAN is that VAE can work with relatively small datasets, while GAN is more prone to mode collapse due to its lack of an encoder [14].

VAE combines concepts from Autoencoder and probabilistic models to generate hidden representations of data. An autoencoder is an artificial neural network designed to learn compressed representations of input data [28] [29]. Synthetic data refers to artificially created data that mimics real data. According to Ewing [5], synthetic data generation serves as a solution to overcome limitations in existing mosquito data. Thus, synthetic data can be used to train mosquito population models, test control strategies, and predict the spread of transmitted diseases. The image augmentation process using VAE consists of two main stages. First, the VAE model is trained using classified mosquito larvae images. This training produces a model capable of learning the training data distribution and generating latent representations that capture essential features of the larvae images. After the VAE model is trained, the second stage involves using the model to generate new variations of larvae images. In this process, the VAE model takes existing mosquito larvae images as input and produces image variations by manipulating latent representations. Image variations generated by VAE are hypothesized—based on prior studies—to support more effective training of recognition and classification systems; however, our work evaluates generative quality only and does not assess downstream classifiers. Prior work on consistency training shows that

data diversity can improve recognition robustness [31]. In our study, we evaluate generative quality only and do not assess downstream recognition performance. This research produces diverse image data via VAE expressly for dataset enrichment under limited field data; broader impacts (e.g., public-health awareness or improved recognition performance) are outside the scope evaluated here.

1.1 Research questions & hypotheses

RQ1 (Generative capability). Can a convolutional VAE trained on 1,300 grayscale mosquito-larvae images learn the underlying data distribution and generate morphologically plausible and diverse samples?
H1. Under our training protocol, the model will converge stably (decreasing total loss) and achieve low Fréchet Inception Distance (FID), indicating statistical similarity between synthetic and real images.

RQ2 (Configuration sensitivity). Which training configuration (latent dimensionality and number of epochs) yields the best generative quality?
H2. Increasing model capacity and training time (e.g., latent = 100 vs. 50; 400 vs. 100 epochs) will reduce FID relative to a low-capacity baseline.

RQ3 (Variant comparison). How does the standard VAE compare with β-VAE, VQ-VAE, and VAE-S on FID under aligned settings?
H3. The standard VAE will achieve the best overall FID across the tested grid, with β-VAE competitive at lower-capacity settings; VQ-VAE and VAE-S will yield higher FID on this dataset.

RQ4 (Deferred / not evaluated here). Does VAE-based synthetic augmentation improve downstream classifier generalization compared with no augmentation or traditional augmentation?
H4 (Deferred). We expect VAE-generated augmentation to improve macro-F1 or accuracy on larvae classification; this is not evaluated in the present study and is deferred to future work (see Limitations & Future Work).

Scope note. We evaluate generative quality only (loss and FID) in this paper and do not evaluate downstream classification or detection models. The downstream hypothesis in RQ4/H4 is pre-specified for future experiments.

2 Related work

Synthetic Data Generation Using VAE has demonstrated significant potential across various fields, including art image generation and data augmentation for biological object classification [24]. Previous studies have validated VAE's capability to learn structured latent space representations and the benefits of data augmentation in enhancing deep learning model performance. Prior Informatica work reports VAE’s expansion to image processing within unsupervised learning, motivating its use here for synthetic mosquito larvae images [38].

Ihsan's (2023) study [25] proved that VAE can learn complex patterns in batik motifs and generate diverse new variations. This research demonstrated VAE's success in mapping batik motif features into latent space, enabling the generation of new motifs while preserving fundamental batik characteristics. These findings are relevant to the current study as they showcase VAE's ability to capture and produce visual variations from datasets, which can be adapted for synthetic mosquito larvae image generation [25].

Further research by Wang [21] in "Comparative Study of VAE and GAN for Mosquito Larvae Image Synthesis" specifically compared VAE and GAN performance in generating synthetic mosquito larvae images. The results showed that while GAN could produce images with sharper texture details, VAE excelled in:

1. Better training stability
2. Capability to generate more diverse yet valid morphological variations
3. More structured latent space interpretability

These findings support using VAE as the primary approach for synthetic mosquito larvae image generation.

Conversely, research by Akter [1] titled "Mosquitoes Classification Using Convolutional Neural Network with Data Augmentation" investigated the impact of data augmentation on mosquito classification accuracy. The results showed accuracy improvement from 70% to 93% after data augmentation implementation, indicating the importance of data variation in training more robust models. However, traditional augmentation techniques (such as rotation, flipping, or color changes) have limitations in creating complex morphological variations. This suggests that VAE-based synthetic data generation can provide better solutions by producing structurally diverse mosquito larvae samples while maintaining their biological characteristics.

Meanwhile, Kim & Myung [8], in their study "Autoencoder-combined Generative Adversarial Networks for Synthetic Image Data Generation and Detection of Jellyfish Swarm," proposed combining Generative Adversarial Networks (GANs) and Autoencoder to generate synthetic jellyfish images. Although model accuracy increased from 80.5% to 83.8%, the GAN-based approach required more complex training and tended to be less stable compared to VAE. These results align with Wang's [21] findings that demonstrate VAE's superiority in training stability for biological data. Based on previous studies, it can be concluded that:

1. VAE has proven effective in learning and generating synthetic image data, including biological images like mosquito larvae [21] and batik motifs [25].
2. VAE offers specific advantages for biological data, including training stability and the ability to generate valid morphological variations.
3. Synthetic data augmentation significantly improves classification accuracy [1], with VAE providing better solutions than traditional augmentation.
4. Although GAN can produce more detailed images, its training complexity makes VAE superior for

applications with limited data [8][21].

This study aims to: (i) leverage a VAE to generate morphologically plausible and diverse synthetic mosquito-larvae images; (ii) quantify generative quality using reconstruction loss and Fréchet Inception Distance (FID); and (iii) compare standard VAE against selected variants (β-VAE, VQ-VAE, VAE-S) under consistent settings. No experiments are conducted on classification or detection models; evaluating downstream performance is beyond this study's scope and is deferred to future work.

The study focuses on the challenge posed by limited mosquito larvae image datasets, which represents a primary obstacle in developing deep learning-based classification systems. These data limitations arise from two key factors: firstly, the difficulties associated with direct data collection, as acquiring mosquito larvae images in the field demands considerable time and resources; and secondly, the complexity of larval variations, as mosquito larvae from different species, such as Aedes aegypti, Anopheles, and Culex, display subtle visual differences, requiring large and diverse datasets for effective model training.

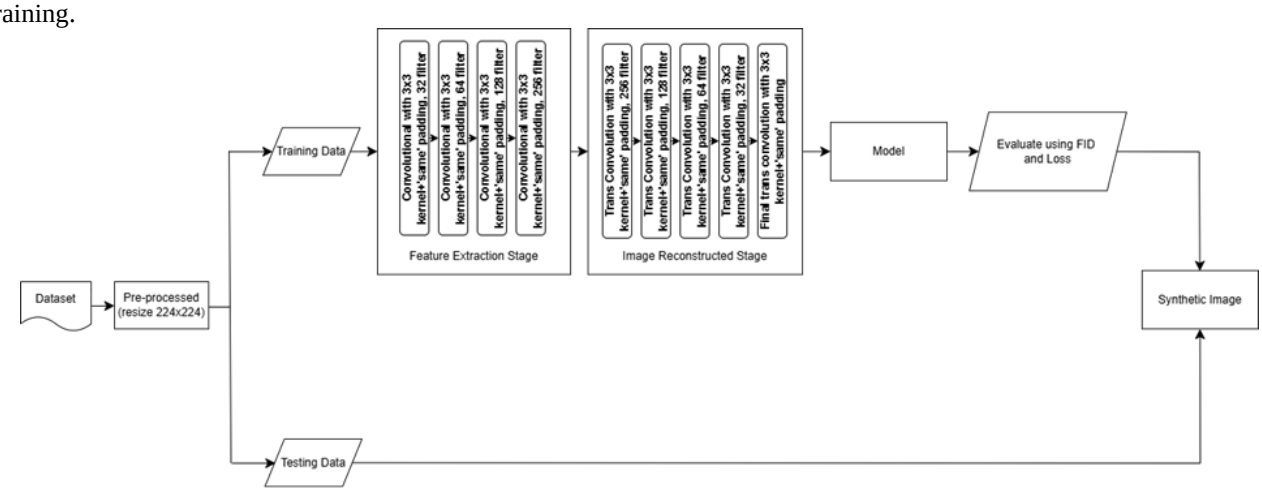


Figure 1: Proposed method

As outlined in Section 1, prior studies indicate that VAEs provide stable training and interpretable latent structures for biological imagery while enabling diversity beyond traditional augmentation [1, 8, 21, 25]. Building on these insights, we employ a VAE to synthesize grayscale larvae images and then detail our architecture and evaluation protocol in Section 3.10.

The system is designed with a VAE architecture consisting of two main components [27] [32]:

1. **Encoder**
   a. Input: Mosquito larvae image sized 224x224 pixels (grayscale).
   b. Process:
      i. Feature extraction using convolutional layers (Conv2D) with 32, 64, 128, and 256 filters.
      ii. Feature compression into a 100-dimensional latent space.
      iii. Generation of probabilistic distributions (z_mean, z_log_var) for latent vector sampling
2. **Decoder**
   a. Input: Latent vectors from the latent space.
   b. Process:
      i. Image reconstruction using Conv2DTranspose layers (256, 128, 64, 32 filters).
      ii. The output of synthetic images sized 224×224 pixels.
3. **Evaluation metrics**
   a. Fréchet Inception Distance (FID): Measures statistical similarity between synthetic and original images across different combinations of latent dimensions and epochs. The best FID score obtained was 0.4668.
   b. Loss Function: Combined reconstruction loss (MSE) and KL-divergence. The obtained loss values ranged approximately from 29206 to 33806.
4. **Comparison with VAE variants**
   Overall, the standard VAE achieved the best FID at latent 100 / 400, while β-VAE was competitive at 50 / 400; VQ-VAE and VAE-S yielded higher FID on this dataset.

3 Proposed method

The proposed method begins with data acquisition and preprocessing, followed by feature extraction using the customized VAE architecture. Subsequent stages include latent space optimization, synthetic data generation, and quantitative evaluation. Figure 1 presents the comprehensive workflow diagram detailing the sequential steps of the proposed research methodology.

Figure 2: Sample images of larvae

3.1 Dataset collection

The mosquito larvae image dataset was obtained from the Balai Teknik Kesehatan Lingkungan dan Pengendalian Penyakit (BTKLPP) Kelas I Medan. The data were collected using a digital microscope in the field. The required image formats were .JPG or .PNG. A total of 1300 images were collected with a size of 640×480 pixels, where 1200 images were used as training data and 100 images as testing data. Fig. 2 shows sample images of larvae.

3.2 Dataset preprocessing stage

The collected dataset with original dimensions of 640×480 pixels was resized to 224×224 pixels and converted to grayscale to accelerate computation. The dataset was then divided into 1200 images for training and 100 images for testing. The resulting images had dimensions of (224, 224, 1) for model training purposes. Fig 3 demonstrates the dataset resizing process.

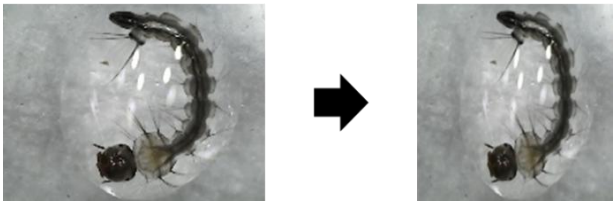


Figure 3: Dataset Pre-processed

3.3 Feature extraction stage

In the feature extraction stage, the encoder will compress high-dimensional input data into a low-dimensional latent space [20]. The VAE encoder is designed with four convolutional layers (Conv2D) that progressively extract visual features from 224×224 pixels mosquito larvae images. The first layer uses 32 filters with a 3×3 kernel and stride 2, followed by layers with 64, 128, and 256 filters using 3×3 kernel size, stride 2, and Leaky ReLU activation function to deepen feature extraction. The output from the final convolutional layer is then flattened and connected to a dense layer of 512 neurons to prepare for compression into the latent space. The extraction results will be mapped to a latent representation with 100-layer dimensionality.

3.4 Image reconstruction stage

The decoder in the VAE architecture is responsible for reconstructing images from latent representations back to their original form through a structured series of processes [20]. This process begins with transforming a 100-dimensional latent vector into a 3D tensor measuring 14×14×256 through a dense layer and reshaping operation. Subsequently, this tensor passes through four transposed convolutional layers (Conv2DTranspose) that perform gradual up sampling. Each transposed convolutional layer employs a 3×3 kernel with stride 2 and 'same' padding, incorporating Leaky ReLU activation functions for intermediate layers and sigmoid activation for the final layer. These layers serve to increase the spatial dimensions of the image while reducing feature depth, starting from 256 filters and ultimately producing a reconstructed image measuring 224×224 pixels with 1 channel (grayscale). The sigmoid activation in the final layer ensures pixel values remain within the (0,1) range, consistent with the input data. This reconstruction process not only generates images resembling the input but also enables the creation of novel variations through latent vector manipulation and forms the basis for calculating reconstruction loss during model training. The reconstruction output can subsequently be used for model quality evaluation or augmentation of mosquito larvae image datasets.

3.5 Variational autoencoder (VAE)

In this study, the VAE serves as the baseline generative model to synthesize additional mosquito-larvae images for data augmentation. The model adopts a convolutional encoder–decoder architecture and leverages the reparameterization trick to enable differentiable sampling from the latent distribution. Training optimizes a composite objective that combines a pixel-level reconstruction term with a Kullback–Leibler (KL) divergence regularizer to promote a well-structured latent space. After training, latent draws are decoded to produce morphology-consistent variants of the original images, which may be integrated into downstream recognition/classification pipelines in future studies; we do not evaluate such integration in this paper. In line with

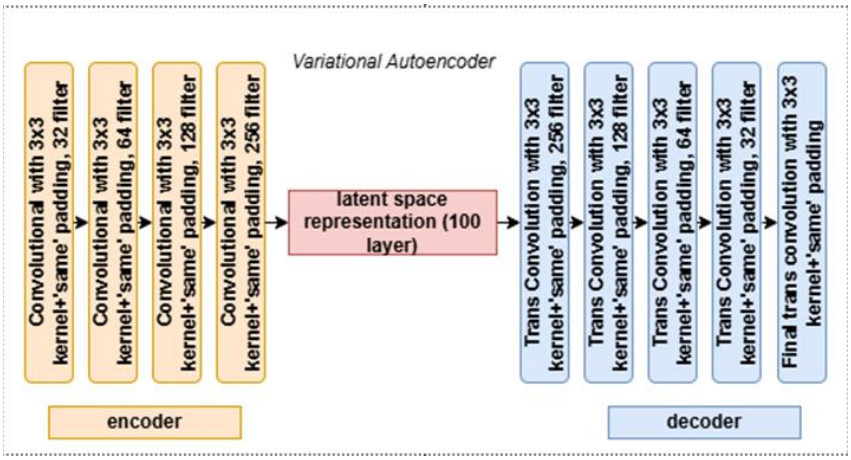


Figure 4: Architecture of VAE

prior uses in image processing, VAEs can generate realistic human faces [30] and remove noise from images [9]; however, a common limitation is that outputs may appear blurry relative to GAN-based models due to the KL-divergence regularization [7]. Unless otherwise specified, training and evaluation settings are aligned with the other model variants to enable fair comparison. A VAE is a generative variant of autoencoders that imposes a prior on the latent code and leverages the reparameterization trick for differentiable sampling, which suits image synthesis tasks [36].

We next detail the specific VAE architecture and training configuration employed in this study. The VAE model was selected for its ability to learn underlying data distributions and produce diverse samples while retaining essential characteristics of the original data. The implemented VAE architecture comprises two main components: an encoder that maps input images to a lower-dimensional latent space and a decoder that reconstructs images from latent representations. The encoder is designed with four convolutional layers (32, 64, 128, and 256 filters) that progressively extract hierarchical features from 224×224-pixel mosquito larvae images. The 100-dimensional latent space effectively captures significant variations across mosquito species. The decoder employs a symmetrical architecture with transposed convolutional layers to reconstruct images from latent vectors (256, 128, 64, and 32 filters). The model is optimized using a combined loss function consisting of reconstruction loss (measuring similarity to input images) and KL divergence loss (regulating latent space distribution). Implementation involved testing various combinations of latent dimensions and epochs to identify optimal parameters using the Adam optimizer. We tune model capacity and training length because VAE performance is sensitive to hyperparameters; Bayesian-optimization studies show systematic tuning can materially improve VAE performance [37]. Fig. 4 illustrates the VAE architecture.

3.6 Beta-variational autoencoder (β-VAE)

Beta-Variational Autoencoder (β-VAE) represents an important variant of the standard VAE that introduces the hyperparameter β as a balancing factor between reconstruction accuracy and latent feature disentanglement capability [6]. The selection of an optimal β value depends heavily on data characteristics and application objectives. β-VAE has proven particularly valuable across various domains requiring high interpretability of latent features, such as in medical image analysis, where the ability to separate independent anatomical factors (including organ shape, tissue texture, and pathological abnormalities) becomes crucial for diagnosis [11].

3.7 Adversarial variational autoencoder (AVAE)

The Adversarial Variational Autoencoder (AVAE) represents an advanced generative architecture that synergistically integrates the VAE framework with GANs [13]. This innovative hybrid configuration strategically employs the VAE decoder as the generator component within the GAN architecture while simultaneously introducing a dedicated discriminator network that learns to differentiate between authentic samples and reconstructed outputs [18]. Within the domain of synthetic data generation, AVAE exhibits significant advantages over conventional VAE implementations, particularly in its capacity to produce high-fidelity synthetic data with superior resolution. This enhanced performance addresses a critical limitation of traditional VAEs, which frequently generate visually degraded outputs characterized by excessive blurring - a direct consequence of over-compression in the latent space representation.

3.8 Vector quantized variational autoencoder (VQ-VAE)

The Vector Quantized Variational Autoencoder (VQ-VAE) represents a generative architecture variation of the VAE that introduces a discrete representation quantization mechanism in the latent space, contrasting with the continuous distribution approach of conventional VAEs. VQ-VAE is a model that demonstrates superior performance in complex data modeling [12]. The primary advantage of VQ-VAE lies in its ability to learn efficient hierarchical representations. This architecture has become a key component in generative models for text-based image generation [17]. VQ-VAE presents several implementation challenges, including:

a. Codebook Collapse, a condition where only 10% to 20% of codes remain active. This may reduce model performance efficiency.
b. Variance Allocation, a situation where the model struggles to distribute variance between the encoder and codebook.
c. Non-Differentiablity, a condition that can cause training instability and may be addressed using the straight-through estimator approach.

3.9 Sparse-variational autoencoder (VAE-S)

The Sparse Variational Autoencoder (VAE-S) represents a specialized variant of the VAE designed to generate sparse latent representations, where only a tiny fraction of units in the latent space remains active [4]. In contrast to conventional VAEs that employ continuous Gaussian distributions and produce dense latent representations, VAE-S enforces sparsity through three primary mechanisms: (1) utilization of non-Gaussian prior distributions [33] (e.g., Laplace or Spike-and-Slab) that naturally promote sparsity, (2) incorporation of L1 (Lasso) or L0 regularization on latent space activations, and (3) implementation of sparse activation functions in the encoder layers [16]. VAE-S offers significant advantages in representation efficiency and interpretability. The model typically activates only 5-20% of latent space dimensions for each input sample [10], making it particularly suitable for high-dimensional data applications such as medical imaging. However, VAE-S also presents several challenges, including complex training procedures due to sparse regularization and heightened sensitivity to λ value selection - where tremendous λ values may induce underfitting. In contrast, overly small values compromise the model's sparsity characteristics [22].

3.10 Model evaluation (protocol)

We evaluate the proposed VAE and its variants by tracking the total loss (reconstruction MSE + KL divergence) during training and by computing the Fréchet Inception Distance (FID) between real and generated images. The dataset contains 1,300 grayscale microscope images resized to 224 × 224 pixels and split 1,200/100 for train/test. Models are trained with Adam (batch size 64) across latent sizes {50, 100} and epochs {100, 400}. For FID, images are resized to 299 × 299 pixels to match the Inception-v3 backbone. Quantitative and qualitative results following this protocol are presented in Section 4. Scope note: We restrict evaluation to generative metrics (loss and FID) and do not evaluate downstream classification or detection results in this paper to make the study design explicit, we predefine success criteria tied to our metrics and baselines:

- S1 — Absolute FID target (primary). Achieve FID ≤ 0.60 on at least one configuration, indicating close alignment between generated and real feature distributions.
  Outcome: Best configuration reached FID = 0.4668 at latent 100 / 400 epochs.
- S2 — Relative improvement vs. baseline (primary). Achieve ≥ 50% relative reduction in FID compared with a low-capacity baseline (latent 50 / 100 epochs).
  Outcome: Reduction from 1.1588 to 0.4668 (≈59.7%).
- S3 — Training stability (primary). Observe monotonically decreasing total loss trends without divergence across epochs; improved configurations should also reduce FID relative to weaker ones.
  Outcome: Loss curves indicate stable convergence; FID improves with capacity/epochs.
- S4 — Variant comparison (secondary). The standard VAE achieves the best FID in ≥ 75% of tested configurations or the best overall FID, while β-VAE may be competitive at lower capacity.
  Outcome: VAE is best in 3/4 configs and achieves the lowest overall FID (0.4668 at 100/400); β-VAE is best at 50/400.
- S5 — Deferred downstream criterion (not evaluated here). In planned future work, VAE-augmented training should yield a statistically significant improvement (e.g., +≥3 pp macro-F1 under 5-fold CV) versus no-augmentation on a standard classifier (e.g., ResNet-18). (Not assessed in this paper.)

Note: These criteria operationalize RQ1–RQ3 within the scope of loss/FID only and explicitly mark RQ4 as deferred.

We next describe the two metrics that operationalize our criteria: (a) Training Loss and (b) Fréchet Inception Distance (FID).

Table 1: Loss value

Epoch	*Kl Loss*	*Reconstruction Loss*	*Loss*
1	47.4243	33758.8398	33806.2656
2	2.7390	32067.9824	32070.7207
3	5.0820	31858.7715	31863.8535
4	6.9111	21770.0117	31776.9238
5	10.1332	31380.1367	31390.2676
6	13.4444	31318.8809	31332.3262
…	…	…	…
395	25.4491	29453.2715	29478.7207
396	25.2640	29430.8984	29456.1641
397	25.5050	29202.8516	29228.3574
398	25.4908	29346.9512	29372.4395
399	25.4973	29324.8281	29350.3262
400	25.4764	29418.4766	29443.9512

a. Loss value

The loss function in VAE consists of two main components: reconstruction loss and KL-divergence loss. Reconstruction Loss is a function used to evaluate the quality of the model's output compared to its original input. It measures the discrepancy between the original input and the reconstructed image produced by the decoder. The reconstruction loss quantifies how closely the reconstruction approximates the original data, typically using metrics such as mean squared error (MSE) or binary cross-entropy (BCE). The following is the reconstruction loss formula.

$$\boldsymbol{L_{reconstruction}} = \|\boldsymbol{x} - \boldsymbol{x'}\|^2 = \sum_{i=1}^{n}(\boldsymbol{x_i} - \boldsymbol{x'_i})^2 \quad (1)$$

Meanwhile, the KL-divergence loss functions as a regularization term that constrains the latent distribution to approximate a prior distribution, typically the standard normal distribution N(0, I) [6]. The Kullback Leibler (KL) Loss represents a fundamental component of the VAE framework, serving to quantify the divergence between the latent distribution predicted by the encoder and the target Gaussian distribution, thereby enabling the generation of realistic synthetic data. The following presents the KL Loss formula.

$$\boldsymbol{L_{KL}}(\mathrm{x}) = -\frac{1}{2}\sum_{j=1}^{d}\left(\mathbf{1} + \log \boldsymbol{\sigma_j^2}(\mathrm{x}) - \boldsymbol{\mu_j^2}(\mathrm{x}) - \boldsymbol{\sigma_j^2}(\mathrm{x})\right) \quad (2)$$

The loss function represents the combined measure of Reconstruction Loss and KL Loss. These two components work synergistically to train the model, enabling it to simultaneously produce accurate data reconstructions while generating well-structured and generalizable latent representations.

The loss function evaluates how effectively the model predicts the desired output. The loss can be formulated as follows.

$$\boldsymbol{L_{total}} = \boldsymbol{L_{reconstruction}} + \boldsymbol{L_{KL}} \quad (3)$$

b. Fréchet Inception Distance (FID) score

The FID has emerged as a robust quantitative metric for assessing the performance of generative models, particularly GANs and Variational Autoencoders (VAEs), by providing a comprehensive measure of sample quality [2]. This metric compares the statistical properties of generated samples against real data distributions in the feature space of a pre-trained Inception network (Inception-v3) [34].

The fundamental superiority of FID over alternative evaluation metrics (including Inception Score and Precision-Recall measures) stems from its unique capacity to jointly evaluate two critical aspects of generative performance: (1) the perceptual fidelity of individual samples and (2) the diversity across the generated sample set [19]. The interpretation of FID scores follows an inverse relationship with generation quality, where decreasing FID values correspond to improving alignment distributions.

Table 2: Detail Result of FID score

Latent Dimension	*Epoch*	Batch Size	FID
50	100	64	1.1588
50	400	64	0.5360
100	100	64	0.6666
100	400	64	**0.4668**

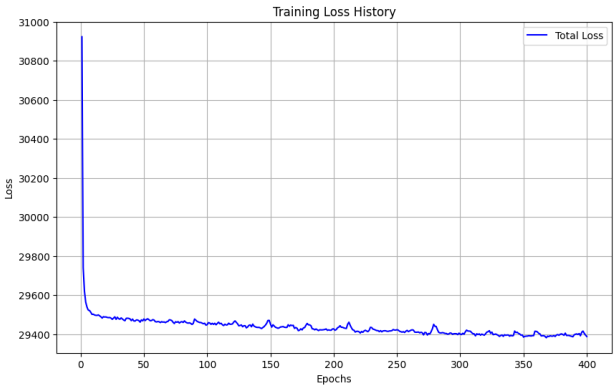


Figure 5(a): Loss chart latent dimension 50 epoch 100

Figure 5(b): Loss chart latent dimension 50 epoch 400

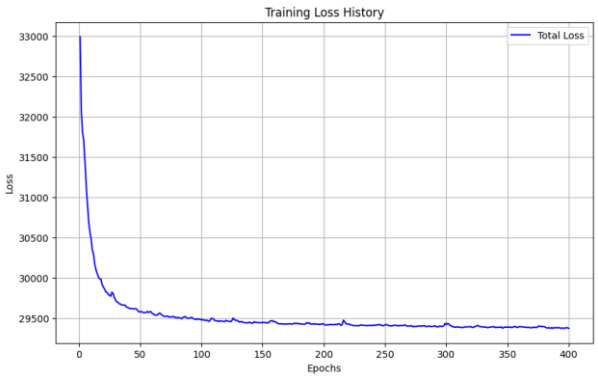


Figure 5(c): Loss chart latent dimension 100 epoch 100

Figure 5(d): Loss chart latent dimension 100 epoch 400

The metric ranges theoretically from 0 to ∞, with scores approaching 0 indicating nearly identical feature statistics between generated and authentic samples, while higher values reflect more significant divergence [3]. This characteristic makes FID particularly valuable for comparative analysis of model iterations or different architectures. The mathematical formulation of FID incorporates the Fréchet distance between multivariate Gaussian distributions fitted to the activation features of real and generated samples, as expressed by the following equation: between generated and real data

$$\boldsymbol{FID} = \|\boldsymbol{\mu_r} - \boldsymbol{\mu_g}\|^2 + \boldsymbol{Tr}\left(\sum_r + \sum_g - 2\left(\sum_r \sum_g\right)^{\frac{1}{2}}\right) \quad (4)$$

4 Performance analysis

4.1 Quantitative results (loss & FID)

Using the protocol in Section 3.10, we evaluate four configurations: (i) latent 50 / 100 epochs, (ii) 50 / 400, (iii) 100 / 100, and (iv) 100 / 400. Loss curves across epochs indicate stable convergence (Fig. 5a–d). The FID scores are 1.1588 (50/100), 0.5360 (50/400), 0.6666 (100/100), and 0.4668 (100/400), with the best performance achieved by latent 100 / 400 epochs (FID = 0.4668). Table 1 summarizes representative loss values, and Table 2 reports FID per configuration. The evaluation results are presented as follows.

A. Loss values

Testing revealed optimal performance with latent dimensions 100 and 400 epochs, demonstrating loss values ranging from 29,206 to 33,806. The loss charts for all dimension-epoch combinations are displayed in Fig. 5a, 5b, 5c, and 5d. Table 1 summarizes the loss values for the optimal configuration.

According to Fig. 5a, the loss values displayed in the loss chart for the combination of latent dimension 50 and 100 epochs range between 29508 to 33900. In Fig. 5b, the loss values shown in the loss chart for latent dimension 50 with 400 epochs range from 29380 to 30950.

According to Fig. 5c, the loss values displayed in the loss chart for the combination of latent dimension 100 and 100 epochs range between 29450 to 30800. In Fig. 5d, the loss values shown in the loss chart for latent dimension 100 with 400 epochs range from 29206 to 33806.

B. FID score

In the evaluation process using FID, the model resizes images to 299×299 pixels in accordance with the Inception-v3 library requirements. The FID results are presented in Table 2. Table 2 shows varied FID scores for each tested combination of latent dimension values and epochs.

Table 3: Comparison of FID Scores for VAE, β-VAE, VQ-VAE, and VAE-S.

Latent Dimension	Epoch	FID			
		VAE	β-VAE	VQ-VAE	VAE-S
50	100	1.1588	1.2956	3.5829	4.7286
50	400	0.5360	0.4826	1.9788	4.7518
100	100	0.6666	0.6679	2.1580	4.7452
100	400	**0.4668**	0.7364	2.1088	4.7683

4.2 Result and discussion

This study successfully implemented the VAE to generate synthetic image data of mosquito larvae with the aim of enhancing a limited dataset. Based on evaluations using loss metrics and the FID, the VAE model demonstrated satisfactory performance. The loss values ranged from 29206 to 33806, with the optimal combination observed at a latent dimension of 100 and 400 epochs. The losses show the model reconstructs images well, though improvement is still possible.

In addition, the evaluation results using FID revealed that the model successfully generated synthetic images that closely resemble the original data. The best FID score achieved was 0.4668, obtained with a latent dimension of 100 and 400 epochs. An FID value close to zero indicates high statistical similarity between the synthetic and original images, suggesting that the VAE effectively captured the visual characteristics of mosquito larvae.

These findings align with previous studies indicating that VAEs excel in training stability and the ability to generate valid larval variations, particularly for biological data such as mosquito larvae. The model's success is further evidenced by comparisons with other VAE variants, including β-VAE, VQ-VAE, and VAE-S, where the standard VAE produced better FID results. This strengthens the argument that VAE is well-suited for limited datasets, as demonstrated in the studies by Wang and Akter. However, an identified drawback is the tendency of the synthetic images generated by the VAE to appear blurry, which is attributed to the strong KL-divergence regularization.

Overall, this study demonstrates that synthetic data augmentation using VAE can serve as an effective solution to address the limitations of mosquito larvae datasets. The generated synthetic data enriches dataset diversity, and low FID values indicate statistical similarity to real images. Nevertheless, this study does not measure the effect of such data on the generalization of classification models. These findings provide a practical recipe for synthesizing larvae imagery that may support future development of larval identification systems; validating epidemiological or surveillance impacts requires downstream studies not performed here. For future research, further exploration of VAE architecture optimization or integration with other techniques, such as GANs, is necessary to reduce the blurriness effect in synthetic images. The researchers also conducted a comparison of FID values between the VAE and four VAE variant models, β-VAE, VQ-VAE, VAE-S, and VAE-S. The comparison results of FID values between the

Figure 6: Synthetic image from model

VAE and the four VAE variant models will be presented in Table 3. The optimal FID score was achieved using the VAE configuration with latent dimension 100 and 400 epochs. Figure 6 presents the synthetic mosquito larvae images generated by the model.

4.3 Findings vs. research questions

- RQ1/H1 (Generative capability): Achieved FID 0.4668 (latent 100 / 400) with stable convergence, supporting the hypothesis that the VAE learns the larvae image distribution and yields diverse samples. S1 and S3 satisfied.
- RQ2/H2 (Configuration sensitivity): Increasing capacity/epochs lowered FID relative to the 50/100 baseline (1.1588 → 0.4668, ≈59.7% reduction). S2 satisfied.
- RQ3/H3 (Variant comparison): The standard VAE attained the best overall FID and was top in 3/4 configs; β-VAE was competitive at 50/400. S4 satisfied with nuance.
- RQ4/H4 (Deferred): Not evaluated in this study; downstream augmentation benefits to be tested in future work. (See Limitations & Future Work.)

5 Limitations & future work

First, our evaluation focuses exclusively on generative quality (loss, FID) and does not include downstream classification/detection experiments. Second, we use a single grayscale dataset with a fixed train/test split, which may limit generalizability. Third, image sharpness—often a limitation of VAEs—was not addressed with perceptual or adversarial refinements.

Future work will (i) measure augmentation gains using standard classifiers (e.g., ResNet-18) under k-fold cross-validation and external held-out sets, (ii) test domain generalization to different acquisition devices/labs, and (iii) explore hybrid VAE–GAN to reduce blur while maintaining stability.

6 Conclusion

This study provides theoretical contributions by demonstrating the effectiveness of VAE in learning the data distribution of mosquito larvae images and generating structured latent representations [9], while reinforcing the theory that VAE can overcome data limitations through synthetic data that preserves essential characteristics of the original data and confirming the importance of KL-divergence regularization for model stability despite causing blurring effects in the output [7][21]. Practically, this research offers an efficient dataset augmentation solution by producing high-quality synthetic images (FID 0.4668) through VAE implementation (latent dimension 100, epoch 400 , offering a reproducible recipe for dataset enrichment under limited field data. Our evaluation is limited to generative metrics (loss and FID); we do not test classification or detection models. Accordingly, any improvement in downstream accuracy or generalization should be regarded as a hypothesis informed by prior literature rather than a finding of this study. Future work will explicitly assess augmentation benefits by training and testing classifiers with and without VAE-generated images.

Comparative analysis of various VAE architecture variants reveals that the standard VAE outperforms β-VAE, VQ-VAE, and VAE-S for this specific biological data context. Furthermore, the stringent KL-divergence regularization creates a trade-off between training stability and visual output sharpness, while the 100-dimensional latent space proves optimal for capturing mosquito larvae morphological variations. Potential implications include supporting the future development of larvae-classification systems via synthetic augmentation and offering a framework that may extend to other vector species in tropical regions; however, these application-level gains require dedicated downstream (and where relevant, clinical/operational) validation. For future research, the exploration of hybrid VAE-GAN architectures is recommended to enhance visual output sharpness, along

with clinical validation of diagnostic system accuracy using synthetic data and the development of integrative pipelines with vector-borne disease surveillance systems. These findings make significant contributions to advancing computer vision methods in tropical public health while opening new research opportunities for applying generative models to address global health challenges.

References

[1] Akter, M., Hossain, M.S., Ahmed, T.U., Andersson, K. (2021). Mosquito Classification Using Convolutional Neural Network with Data Augmentation. In: Vasant, P., Zelinka, I., Weber, GW. (eds) Intelligent Computing and Optimization. ICO 2020. Advances in Intelligent Systems and Computing, vol 1324. Springer, Cham. https://doi.org/10.1007/978-3-030-68154-8_74 15

[2] Asperti et al. 2022. Enhancing Variational Generation Through Self-Decomposition. Digital Object Identifier, 10.1109/ACCESS.2022.3185654. IEEE. https://doi.org/10.1109/ACCESS.2022.3185654

[3] Borji, A. (2022). "Pros and Cons of GAN Evaluation Measures: New Developments". Computer Vision and Image Understanding, 215, 103329. https://doi.org/10.1016/j.cviu.2022.103329

[4] Dai, B., & Wipf, D. (2019). Diagnosing and Enhancing VAE Models: A Sparse Coding Perspective. International Conference on Learning Representations (ICLR). https://doi.org/10.48550/arXiv.1906.02691

[5] Ewing, D. A., Cobbold, C. A., Purse, B. V., Nunn, M. A., & White, S. M. (2016). Modelling the effect of temperature on the seasonal population dynamics of temperate mosquitoes. Journal of theoretical biology, 400, 65-79. https://doi.org/10.1016/j.jtbi.2016.04.008

[6] Hinggis et al. 2017. β -VAE: LEARNING BASIC VISUAL CONCEPTS WITH A CONSTRAINED VARIATIONAL FRAMEWORK. Under review as a Conference Paper at ICLR. https://doi.org/10.48550/arXiv.1804.03599

[7] Huang, T., Ding, Z., Zhang, J., Tai, Y., Zhang, Z., Chen, M., Wang, C., & Liu, Y. (2023). Learning to Measure the Point Cloud Reconstruction Loss in a Representation Space. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 12208–12217. https://doi.org/10.1109/CVPR52729.2023.01175

[8] Kim, Kyukwang, Myung, Hyun. (2018). Autoencoder-Combined Generative Adversarial Networks for Synthetic Image Data Generation and Detection of Jellyfish Swarm. Journal Article, 6, 54207-54214. IEEE. https://doi.org/10.1109/ACCESS.2018.2869250

[9] Kingma, D. P., & Welling, M. (2014). Auto-Encoding Variational Bayes. Proceedings of the 2nd International Conference on Learning Representations (ICLR). arXiv:1312.6114. https://doi.org/10.1609/aaai.v33i01.3301492

[10] Li, X., et al. (2019). Sparse Variational Autoencoder for Unsupervised Feature Learning. Proceedings of the AAAI Conference on Artificial Intelligence. https://doi.org/10.1109/CVPR52729.2023.01175

[11] Locatello, F., et al. (2019). Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. ICML. https://doi.org/10.48550/arXiv.1811.12359

[12] Luo et al. 2024. Quaternion Vector Quantized Variational Autoencoder. IEEE SIGNAL PROCESSING LETTERS, Vol. 32, 1070-9908. IEEE. https://doi.org/0.1109/LSP.2024.3385109

[13] Mescheder, L., Nowozin, S., & Geiger, A. (2017). "Adversarial Variational Bayes: Unifying Variational Autoencoders and Generative Adversarial Networks". Proceedings of the 34th International Conference on Machine Learning (ICML), 70, 2391-2400. https://doi.org/10.5555/3305890.3306030

[14] Naderi, H., B. H. Soleimani, dan S. Matwin. 2020. Generating High-Fidelity Images with Disentangled Adversarial VAEs and Structure-Aware Loss. 2020 International Joint Conference on Neural Networks (IJCNN), 978-1-7281-6926-2, 20. IEEE. https://doi.org/10.1109/IJCNN48605.2020.9206846

[15] Ndione, R. D., O. Faye, M. Ndiaye, A. Dieye, dan J. M. Afoutou. 2007. Toxic effects of neem products (Azadirachta indica A. Juss) on Aedes aegypti Linnaeus 1762 larvae. In African Journal of Biotechnology. 6(24): 2846- 2854. https://doi.org/10.5897/AJB2007.000-2348

[16] Nguyen, T., et al. (2023). Adaptive Sparse Variational Autoencoder with L0 Regularization. IEEE Transactions on Pattern Analysis and Machine Intelligence. https://doi.org/10.1109/TPAMI.2022.3220765

[17] Ramesh et al. 2021. Zero-Shot Text-to-Image Generation. International Conference on Machine Learning (ICML). https://doi.org/10.48550/arXiv.2102.12092

[18] Rosca, M., Lakshminarayanan, B., & Mohamed, S. (2019). "Improving Generalization in Generative Adversarial Networks via Hierarchical Variational Inference". International Conference on Learning Representations (ICLR). https://doi.org/10.48550/arXiv.1807.03653

[19] Sajjadi, M. S., Bachem, O., Lucic, M., Bousquet, O., & Gelly, S. (2018). Assessing generative models via precision and recall. Advances in Neural Information Processing Systems, 31. https://doi.org/10.48550/arXiv.1806.00035

[20] Tai, X.-C., Liu, H., & Chan, R. (2023). PottsMGNet: A Mathematical Explanation of Encoder-Decoder Based Neural Networks (arXiv:2307.09039). https://doi.org/10.48550/arXiv.2307.09039

[21] Wang, R., et al. (2022). Comparative Study of VAE and GAN for Mosquito Larvae Image Synthesis. Computers in Biology and Medicine. https://doi.org/10.1016/j.compbiomed.2022.105440

[22] Wang, Y., Zhang, L., Chen, X., & Liu, Q. (2023). Adaptive sparse variational autoencoders: Balancing sparsity and reconstruction accuracy. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(4), 1895-1909. https://doi.org/10.1109/TPAMI.2022.3189841

[23] P. I. Nainggolan et al., (2023). Classification Of Aedes Mosquito Larva Using Convolutional Neural Networks and Extreme Learning Mechine. 2023 7th International Conference on Electrical, Telecommunication and Computer Engineer (ELTICOM). doi: 10.1109/ELTICOM61905.2023.10443125.

[24] P. I. Nainggolan et al., (2025). Detection and Classification of Mosquito Larvae Based on Deep Learning Approach. Eng. Lett. 2025, 33, 198–206.

[25] Ihsan. (2023). Initial Study of Batik Generation using Variational Autoencoder. 8th International Conference on Computer Science and Computational Intelligence (ICCSCI 2023).

[26] Azis. (2023). Comparative Analysis of Variational Autoencoder (VAE) and Generative Adversarial Network (GAN) Algorithms for image classification. Journal Eletronik Sistem InformasI (JESII).

[27] M. Sami and I. Mobin, "A Comparative Study on Variational Autoencoders and Generative Adversarial Networks," in 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT), IEEE, Mar. 2019, pp. 1–5. https://doi.org/10.1109/ICAIIT.2019.8834544

[28] L. Pinheiro Cinelli, M. Araújo Marins, E. A. Barros da Silva, and S. Lima Netto, "Variational Autoencoder," in Variational Methods for Machine Learning with Applications to Deep Networks, Cham: Springer International Publishing, 2021, pp. 111–149. https://doi.org/10.1007/978-3-030-70679-1_5

[29] Q. Xu, Z. Wu, Y. Yang, and L. Zhang, "The difference learning of hidden layer between autoencoder and variational autoencoder," in 2017 29th Chinese Control and Decision Conference (CCDC), IEEE, May 2017, pp. 4801–4804. https://doi.org/10.1109/CCDC.2017.7979344

[30] Mansour, R. F., Escorcia-Gutierrez, J., Gamarra, M., Gupta, D., Castillo, O., & Kumar, S. (2021). Unsupervised deep learning based variational autoencoder model for COVID-19 diagnosis and classification. Pattern Recognition Letters, 151, 267–274. https://doi.org/10.1016/j.patrec.2021.08.020

[31] Xie, Q., Dai, Z., Hovy, E., Luong, T., & Le, Q. (2020). Unsupervised data augmentation for consistency training. Advances in Neural Information Processing Systems, 33, 6256–6268.

[32] M. Sami and I. Mobin, "A Comparative Study on Variational Autoencoders and Generative Adversarial Networks," in 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT), IEEE, Mar. 2019, pp. 1–5. doi: 10.1109/ICAIIT.2019.8834544.

[33] Yu. (2024). CS-Intro VAE: Cauchy-Schwarz Divergence-Based Introspective Variational Autoencoder. IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 26. https://doi.org/10.1109/TMM.2024.3365439

[34] Nash, C., Menick, J., Dieleman, S., Battaglia, P.W., 2021. Generating images with sparse representations. arXiv preprint arXiv:2103.03841.

[35] Razavi, A., Oord, A.v.d., Vinyals, O., 2019. Generating diverse high-fidelity images with vq-vae-2. arXiv preprint arXiv:1906.0044.

[36] A. Chefrour and L. Souici-Meslati, "Unsupervised Deep Learning: Taxonomy and Algorithms," Informatica (Slovenia), vol. 46, no. 2, pp. 151–168, 2022, doi: 10.31449/inf.v46i2.3820.

[37] B. Fu, "Variational Autoencoder-based High-dimensional Feature Extraction for Economic Analysis of Power Cost Data," Informatica (Slovenia), vol. 49, no. 25, pp. 75–91, 2025, doi: 10.31449/inf.v49i25.8012.

[38] L. Zou and M. Zhang, "Variational Autoencoder Model Combining Deep Learning and Probability Statistics and Its Application in Large-scale Data Analysis," Informatica (Slovenia), vol. 48, no. 22, pp. 31–46, 2024, doi: 10.31449/inf.v48i22.6921.

Enhancing Test Suite Optimization in Software Engineering through a Hybrid Whale Optimization and LSTM Approach

Shuqing Qiao
College of computer science and cyber security (pilot software college), Chengdu University of Technology, Chengdu Sichuan,610059, China
E-mail: 15386513158@163.com

Keywords: test suite, software engineering, whale optimization algorithm, test generation, parameters tuning

Recieved: June 10, 2025

The algorithms used for Meta are commonly utilized to solve complicated optimization issues, but they frequently have disadvantages such as premature convergence, poor examination in the initial phases of the search, and insufficient adaptation of search operators. To address these issues, this study presents a novel hybrid approach, the Whale Optimization Algorithm with Long Short-Term Memory (WOA-LSTM), which is intended to improve exploration and adaptive learning. The primary goal is to overcome the performance limits of traditional WOA by providing a more robust search approach for confined combinatorial test-generating tasks. The proposed WOA-LSTM combines the regular WOA with an LSTM-based probability table that dynamically selects search operators. Data normalization with Z-score standardization resulted in stable feature scaling for effective LSTM training. In the early iteration stages, low-performing operators are given a chance to re-enter the search, preventing premature stagnation. Furthermore, the model selects just-in-time adaptive operators among the three WOA mechanisms random generation, spiral shape, and shrinkage—based on their historical effectiveness, allowing for a balance of exploration and exploitation. The experimental results show that WOA-LSTM outperforms other algorithms, with an accuracy of 99.3% and a recall of 99.1%. Comparative examination shows that the suggested method outperforms traditional WOA while being competitive with other sophisticated metaheuristic algorithms. Furthermore, statistical validation with t-tests demonstrates WOA-LSTM's considerable superiority over existing techniques, ensuring the proposed model's performance is reliable, resilient, and generalizable across several experimental runs. Finally, WOA-LSTM presents a highly successful optimization framework that overcomes WOA's drawbacks while also pointing the way forward for future optimization research.

Povzetek: Članek obravnava neučinkovitosti klasičnega algoritma kita (WOA) pri optimizaciji testnih naborov, predvsem prezgodnjo konvergenco in šibko zgodnjo eksploracijo. Predlaga hibrid WOA-LSTM, kjer LSTM prek verjetnostne tabele sproti izbira najboljše WOA-operatroje. Metoda izboljša ravnovesje eksploracije/eksploatacije.

1 Introduction

The research suggests a novel method for t-way (TW)test suite (TS) development that supports restrictions based on the Whale Optimization Algorithm (WOA), was created by taking inspiration from the humpback whale's hunting habits. WOA's unique optimization mechanism gives it a powerful global search capability [1]. Furthermore, WOA has a simple implementation and is less dependent on parameters. As a result, it is often proposed to solve a varied range of problems, such as feature selection, energy, electrical power, flow shop scheduling, clustering, and electronic engineering. Furthermore, the WOA has revealed modest results in every field [2]. Given that WOA performs well against a variety of meta-heuristics, its usage for the presently recommended combinatorial TW TS development is warranted [3].

The literature has developed a number of useful metaheuristic-based methods that consider TW testing as an optimization difficult. The performance of each metaheuristic-based TW technique is often determined by its backbone algorithm [4]. To reevaluate the worst-performing operator in the initial iteration and investigate additional search regions, the Exponential Monte Carlo algorithm's (EMC) acceptance probability is also incorporated. Testing is a crucial step in the software (Sof) development life-cycle and is part of the quality assurance effort. Extensive testing is nearly impossible due to budget and time constraints, even though it is desirable to guarantee adherence to specifications [5]. In the literature, numerous sampling-based strategies (such as boundary

values analysis and equivalence partitioning) have been put forth to minimize testing [6]. Despite their usefulness, a large number of current sampling techniques do not account for interaction-related faults (also known as TW testing, where t denotes the degree of the interaction).

Finding a explanation that satisfies all or maximum of the given restrictions while elevating a certain performance metric under a set of predetermined situations is the fundamental goal of optimization issues [7]. Optimization difficulties have become much more complicated and diverse as a result of the rapid breakthroughs in science and technology [8]. These difficulties, which frequently include high-dimensional spaces, go beyond conventional linear or quadratic programming and include characteristics like, dynamic changes, nonlinearity, multi-impartiality, and high uncertainty [9]. Researchers and engineers have created a wide variety of optimization procedures to successfully handle these crucial and more complicated optimization problems [10]. Particularly, swarm intelligence algorithms have become well-known because of their distinct benefits and proven ability to solve challenging optimization issues [11].

Contribution of the study:

- The study is the first of its type to use WOA for the creation of TW TS.
- Using a series of benchmark TS, the study thoroughly assesses WOA's performance.
- Next, just-in-time adaptive selection of the best-performing operators (between random generation, spiral shape mechanism, and shrinking mechanism) is made possible by a new WOA-LSTM variant algorithm at any given iteration. To avoid local optima, the study uses LSTM.
- The proposed study provides the effective results of the combinatorial TS with constraint support.

2 Literature review

The researcher presented GWOATEO, a novel hybrid optimisation strategy that combines heat exchange optimisation methods and the genetic algorithm in a synergistic manner [12]. By employing memory-based limit methods and location informing techniques for its leading answers, this algorithm improves its search capabilities. An antagonistic learning approach was also used in the study to produce the opposite of the global optimum solution. This improved the search efficiency and procedure performance. The multi-leader guided multi-objective WOA that includes an oppositional learning technique (OLT) to enhance the initial population spreading. The OLT is successful in generating a contrasting answer to the global optimum, hence improving the overall optimal solution, in addition to optimising the initial population distribution [13]. The researcher balanced algorithmic exploration and exploitation by employing an elite search strategy and an adaptive variable-speed technique to develop global optimisation [14]. They successfully higher the trade-off between algorithmic advancement and search performance by employing adaptive inertia weights, modified global optima, and historical individual optimal solutions. The related a WOA that integrates the Lévy flight trajectory mechanism to enhance the algorithm's capacity to avoid local optima, avoid premature convergence, and encourage population variety. The researcher enhanced the WOA by incorporating a limited mutation mechanism to increase convergence efficiency and adaptive nonlinear inertia weights to control convergence speed in order to address economic load scheduling problems [15]. For the first population generation and convergence stages, the researcher used a dynamic penalty function method using the Gaussian Variational methodology. These improvements greatly increased the algorithm's accuracy, resilience, stability, convergence rate, and global search capabilities. Furthermore, they demonstrated advantageous outcomes when dealing with increasingly intricate constrained optimisation issues. In order to balance exploration and exploitation, avoid stagnation, and enhance the quality of initial solutions, the researcher employed scheduling concepts, nonlinear convergence elements, and mutation approaches. The energy-saving scheduling issue has been successfully resolved by these approaches.

Table 1 displays the summary of recent studies on TS optimization utilizing various algorithms, with emphasis on aims, applicable methodologies, and constraints, particularly in terms of performance and scalability of WOA versions.

Table 1: Established from current research on metaheuristic-based TS optimization.

References	**Objectives**	**Methods**	**Limitation**
Deneke et al., [16]	To reduce regression TS, remove unnecessary test cases (TC), improve productivity, and lower execution costs in Sof testing.	Particle Swarm Optimization (PSO) was used to reduce TS size in comparison to Greedy Weighted Set Cover (GWSC), etc.	Performance is dependent on dataset features; PSO necessitates careful parameter adjustment and may not always result in globally optimal TS reduction.
Ahmed et al., [17]	Create a prioritized TW TS generating approach that balances TC weight and priority for effective Sof testing.	The Bi-objective Dragonfly Algorithm (BDA) creates prioritized TW TS.	BDA could demand more processing resources for large datasets, and prioritizing relies on proper weight and priority assignment.
Zayed and Maashi [18]	To save effort, time, and money on regression testing, minimize the size of the TS while increasing Sof coverage.	Genetic and greedy algorithms optimize regression TS efficiently.	The method could struggle with large Sof systems. Solution quality was dependent on algorithm parameters and initial TS design.
Ouyang et al.,[19]	To address the TS case issues, the Improved WOA (ImWOA) is presented in this study.	Improved WOA	The application of the whale optimization technique to large-scale TS optimization problems may be limited by its high computer resource requirements.
Zamani et al., [20]	The limitations of WOA in resolving global optimisation issues, including its propensity to converge slowly and its susceptibility to local optimum, are addressed in this study by proposing MSWOA.	MSWOA	The complexity and features of the TS may have an impact on how well the whale optimization technique performs.
Abdollahzadeh et al., [21]	With an emphasis on both traditional and cutting-edge techniques, such as deep learning, hybrid approaches, and generative AI models, the study highlights current trends, difficulties, and prospects for additional research.	TS optimization	It may be necessary to carefully adjust the whale optimization algorithm's settings, which can take time, to maximize its performance.
Gharehchopogh et al., [22]	This study examines the various tools, regression TS optimization approaches, and mathematical models currently in use for RTO.	Genetic algorithm	A tiny TS and the genetic algorithm's population size won't provide the GA with adequate solution space to generate reliable findings.
Cao et al., [23]	This study looked into a method for saving time and effort when performing regression testing.	meta-heuristic algorithm	WOA experiences premature convergence, much like any other meta-heuristic algorithm, including swarm and evolutionary algorithms.
Wang et al., [24]	This study examines the various tools, regression TS optimization approaches, and mathematical models currently in use for RTO.	Regression Test Suite Optimization (RTO)	The study might not offer a thorough comparison with other optimisation methods, which could restrict our comprehension of the relative performance of the whale optimisation technique.

Liu et al., [25]	In this paper, we propose a four-step reinforcement learning-based TS prioritization methodology.	Support vector machine (SVM)	Examining the WOA scalability for extensive TS optimization issues.
Burachynskyi and Shantyr [26]	Analyze AI integration in Sof development to improve processes, automate jobs, increase product quality, lower costs, improve testing, streamline project management, and enable adaptive, intelligent Sof systems.	Literature review, case studies, AI model analysis, workflow assessment.	The research was founded on previous research; practical implementation hurdles, dataset limits, model generalization issues, and diversity in organizational adoption may all have an impact on the applicability of AI techniques across various Sof development settings.

2.1 Research gap

Due to its simple design, strong adaptability, and few control parameters, researchers have found the WOA to be especially interesting. Numerous complicated optimisation problems, including fuzzy programming, DNA fragment assembly, feature selection and workshop scheduling, image segmentation, engineering design, and path planning, have been successfully solved by this technique [12]. However, when applied to complex issues and high-dimensional data, the WOA's shortcomings become evident. In particular, its optimisation pace is somewhat poor, frequently falling short of the need for quick fixes. Furthermore, the optimisation accuracy is too low, which could result in less-than-ideal solution quality. Additionally, the WOA's search and exploitation capabilities are lacking, which limits its efficacy in real-world applications by impeding its capacity to fully and profoundly explore the solution space. Many researchers have carried out in-depth analyses and put forth a range of successful enhancement strategies in an effort to overcome these limitations and improve the WOA's performance. By improving the WOA's algorithmic framework with deep learning techniques, parameter configurations, search tactics, and other pertinent elements, these techniques aim to maximise and improve the solution results.

3 Materials and methods

The research uses WOA- LSTM to optimize TS. The preprocessing step includes data cleaning and applying Z-score standardization, which essentially transforms the data values into a stable input. The WOA mimics the feeding behaviors of whales with encircling, spiral, and random operators that are all inherently guided by a probability table. The LSTM captures dependencies of sequential data points from the historical data inputs, allowing for accurate sequence generation. Once the LSTM and WOA are integrated together, outcomes observed are an efficient exploration versus exploitation, and generated and optimized TS with high performance. Figure 1 shows the methodology flow diagram for TS Optimization.

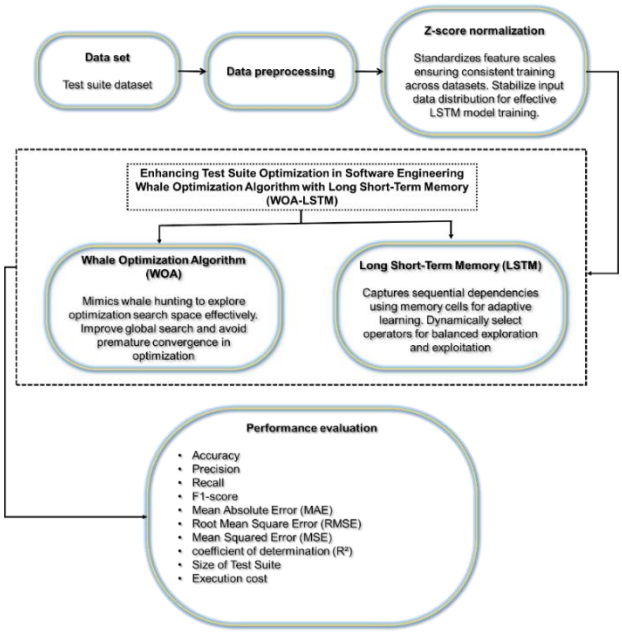


Figure 1: show the methodology flow diagram.

3.1 Data collection

A "test suite dataset" is a group of data used to assess how well an algorithm or model performs. It is a collection of scenarios, TC, or scripts used to verify the performance and functioning of a system or Sof program. The data is obtained from Kaggle https://www.kaggle.com/datasets/bwandowando/mushroom-overload. Kaggle's mushroom dataset has 8,124 instances with 22 categorical attributes that describe multiple mushroom features (e.g., cap shape, surface, color, gill size, odor, habitat). Each observation instance was either labeled edible or poisonous; therefore, the mushroom dataset is appropriate for a classification and optimization problem. The mushroom dataset is particularly useful in evaluating TS generation models because it has a variety of well-defined categorical data. The mushroom dataset also provides an appropriate level of complexity to allow the proposed WOA-LSTM to

demonstrate the robustness of feature handling, exploration, and adaptive learning abilities while generating a TC sequence.

3.2 Data preprocessing using Z-score normalization

Z-score normalization is a statistical method that normalizes input by removing the mean and obtaining unit variance.
With normalization, can guarantee that features contribute equally to the learning process and reduce the influence of extreme values or outliers. Z-score normalization maps values less than the mean on the negative scale, greater than the mean on the positive scale, and sets zero maps to zero. The goal of normalization is to establish standard and s **Parameter tuning** feature scales, leading to model fairness, reduced bias, and better optimization in optimization frameworks. During Z-score normalization, the transformation is as follows in equation (1).

$$u' = \frac{u-\mu}{\sigma} \quad (1)$$

u' Is the normalized feature, u is the original feature value, μ is the mean, and σ is the standard deviation. The process converts the dataset into a distribution with a mean of zero and a variance of one, resulting in a standardized environment for algorithmic training. The change simplifies feature comparison while also improving the interpretability of adaptive system outputs. By consistently rescaling data, the approach allows for methods such as LSTM to be trained with balanced inputs, preventing skewed outcomes. To verify that all features have a consistent statistical distribution, facilitating stable convergence and balanced operator assessment in WOA-LSTM.

3.3 T way

There is usually a strict timeline that the testers must follow. Therefore, very successful testing techniques are required to achieve optimal test coverage and a high rate of problem detection. TW testing provides the features mentioned above. TW testing is a technique that looks at every distinct combination of the Sof system's parameters. The Covering Array (CA) is a particular Design of Experiments, called a TS. Making a TC from a Sof system input or system configuration is how t-. The purpose of TW testing is to thoroughly investigate interactions between various input parameters in a Sof system, so that serious faults caused by parameter combinations are discovered early.

3.4 (DoE) strategies used in T-way testing

CAs generalize OAs by requiring that the t-tuples be included at least once rather than a predetermined number of times in order to obtain balanced t-tuples. A tuple is a finite, ordered list of mathematical elements, such as parameter interaction values. A t-tuple is an ordered set of t elements. The combinatorial object CA is represented as CA (N; t, vp), where t is the strength of the interaction, v is the parameter value, N is the number of test instances generated, and p is the parameter (e.g., input data, system configurations, or both).

The primary objective of TW testing is to minimise the size of the TS without sacrificing its ability to identify problems. The TW testing is to reduce the overall size of the TS while ensuring that all essential parameter interactions are covered, thus balancing efficiency and effectiveness. A group of TC designed to evaluate a Sof system and illustrate a certain set of behaviors, in a way that challenging is carried out. A variety of examination scenarios are then used to build the Sof system's TS in Figure 2.
Figure 2 illustration below depicts the step-by-step procedure of creating a combinatorial TS with the WOA-LSTM algorithm. That starts with a random selection of beginning TC from the t-tuple table. The WOA-LSTM algorithm is then used to optimize these TC, with a fitness function that reduces weight. The best TC with the highest number of interactions is chosen. The TC covers and removes the corresponding entries from the t-tuple table. The operation continues until the table is empty.

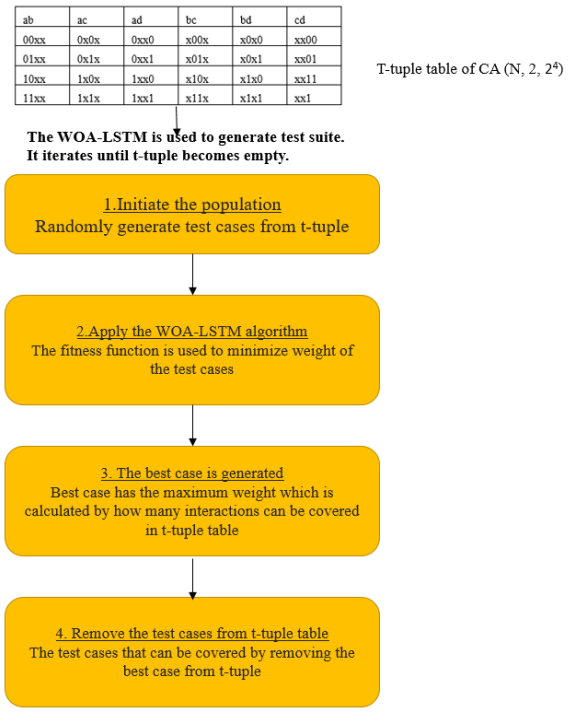


Figure 2: T way testing implementation

3.5 Whale optimization algorithm with long short-term memory (WOA-LSTM), used to improve exploration and adaptive learning

The combined use of LSTM and a WOA incorporates sequential learning with an adaptive search to optimize the TS in Sof testing. In these pairs of algorithms, LSTM captures temporal dependencies in a sequence while providing candidates of test sequences, while WOA evaluates and filters through candidates by using encircling, spiral, and random operators integrated with a probability table to dynamically adjust operator selection - remembering past performance to boost efficiency while balancing between exploration and exploitation. This combination addresses redundant TC, enhances fault detection, results in an optimized TS with high accuracy and reliability in testing, and is effective in handling the complexity and size of modern Sof systems.

The WOA is selected for its successful exploration and exploitation capability, but it suffers from early convergence and limited versatility. LSTM is incorporated to expand the capability of WOA, as it can represent and capture sequential performances and is adaptable to operator selection. The hybrid WOA-LSTM exploits the exploratory strength of WOA paired with the adaptability given from sequential memory by LSTM, which yields a balanced exploration and exploitation process. This displays a better convergence time, prevents stagnation, and ultimately provides a greater accuracy in a rich and complex combinatorial optimization problem, therefore offering a strong optimization framework.

Whale optimization algorithm: One metaheuristic optimisation technique that can be used for Sof engineering TS optimisation is the WOA. It is employed to identify the optimal subset of TC to minimise the number of TC conducted and maximise fault detection. By employing tactics like encircling prey and bubble-net attacking to investigate and take advantage of the search space, WOA mimics the hunting behaviour of humpback whales in Figure 3. WOA is a designer-inspired meta-heuristic based on the foraging behavior of humpback whales, which uses three operators - encircling prey, spiral bubble-net attacking, and random exploration - to find optimal solutions. In this research, WOA is applied to TS optimization, where the "prey" is the perfect subset of TC. By continuously updating the position of the whales, WOA will minimize redundant TC while maximizing coverage of fault detection, and this is done efficiently.

Figure 3 displays the WOA. The whale represents the leader directing the search, and the fish represent agents or candidate solutions. The spirals depict what whales might do during encircling and bubble-net attacking, representing the exploitation behaviour of the WOA algorithm. The clusters of spheres above represent distinct solution spaces. The swarm of fish or agents converges around possible high-quality solutions, representing how WOA explores and exploits the search space. The image represents both the exploration and convergence dynamics of WOA.

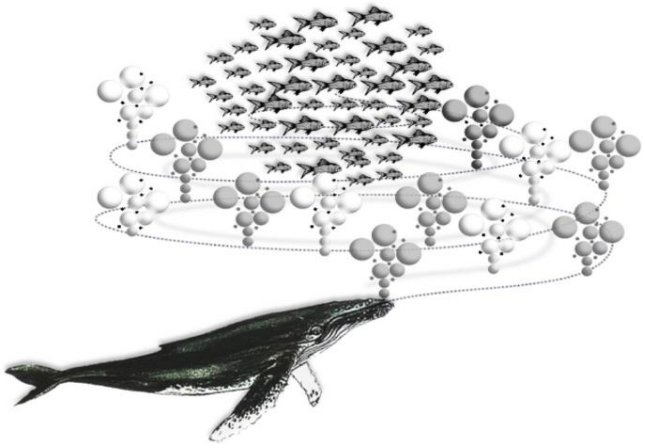


Figure 3: Whale optimization

Test suite optimisation makes use of WOA: WOA considers the ideal TS to be the "prey" in the search space, encircling it. By iteratively updating their positions and getting closer to the best-found answer, whales (search agents) efficiently cut down on the number of TC while preserving coverage.

Bubble-Net Attacking: WOA imitates humpback whales' spiral attack. This facilitates search space exploration, especially when looking for answers that are not immediately close to the best-found one.

Whales increase diversity and avoid premature convergence to local optima by randomly exploring the search space while looking for prey.

Encircling prey: Whales iteratively search for the optimal solution. Bubble-Net Attacking: WOA mimics humpback whales' spiral attacks to explore local/global areas. Random exploration: Whales diversify their searches to avoid premature convergence. In this research, these mechanisms are expanded with a probability-driven adaptive operator selection, in which the choice between encircling, bubble-net assaulting, and random exploration is dynamically modified by their previous performance. Therefore, it indicates that exploration and exploitation are balanced throughout the iterative process. Table 2 displays the hyperparameter table for WOA.

Table 2: The hyperparameter table

Method	Hyperparameter	Notation / Variable	Description
WOA	Population size	*n*	Number of whales (search agents).
	Iterations	*d*	Maximum number of generations.

	Convergence factor	$\vec{\mathbf{E}}$	Linearly decreases from 2 to 0, balancing exploration/exploitation.
	Swing factor	$\vec{\mathbf{V}}$	Controls the movement step size in the position update.
	Random coefficient	$\vec{k}$	Random value in [0,1] for search randomness.
	Spiral constant	*(T')*	Determines spiral-shaped bubble-net updating.
	Probability threshold	*b*	Decides operator selection (spiral vs. random/encircling).

Predicting using long short-term memory (LSTM): LSTM is commonly used to handle gradient vanishing and explosion during complicated Prediction training. It needs global processing, and it is time-dependent. It has memory (hidden) cells that dynamically adjust the next step based on data from the previous stage. In the research, LSTM networks are used to forecast cost overruns, schedule deviations, and structural performance trends as shown in Figure 4. It consists of three gates: the input gate, the output gate, and the forget gate. The input gate is calculated by using Equation (2). LSTM is a recurrent neural network (NN) variation that performs well with sequential and time-dependent data. It addresses vanishing gradients using memory cells and three gates: input, forget, and output. In that research, LSTM is utilized to predict and produce candidate TS sequences while capturing dependencies between TC. An attention mechanism is included to highlight crucial test sequences, hence improving prediction accuracy.

$$i_t = \sigma(W_i[g_{t-1}, y_t] + c_i) \quad (2)$$

Where y_t signifies input data and g_{t-1} denotes the result of the preceding step. The input gate's bias and weight are denoted as c_i and W_i, respectively, whereas σ is a sigmoid function. i_t: Input gate vector (Vec). t: Time step. The data is further processed using the forget gate (f_t), as stated in Equation (3).

$$f_t = \sigma\left(W_f[g_{t-1}, y_t] + c_f\right) \quad (3)$$

W_f is the forget gate's weight, g_{t-1} is the preceding step's output, and the gate bias isc_f. f: forget gate. t: Time step.The following step states, as illustrated in Equation (4).

$$\bar{E}_t = tanh(W_E[g_{t-1}, y_t] + c_E) \quad (4)$$

Where $\bar{E}_t$ specifies the preceding step's state, c_E represents its bias, and W_E defines its weight. The previous state is used to determine the cell state at the present moment. $tanh$: Hyperbolic tangent function. Therefore f_t and $\bar{E}_t$ can be retrieved from Equations (1), (2), and (3), and they can be used in Equation (5) to calculate the current cell state.

$$E_t = f_t * E_{t-1} + i_t * \bar{E}_t \quad (5)$$

Lastly, the cell state is sent to the output gate as shown in Equation (6)

$$o_t = \sigma(W_o[g_{t-1}, y_t] + c_o) \quad (6)$$

Where o_t stands for the output gate, and W_o and c_o are its weight and bias, as shown in Equation (7), respectively.

$$h_t = o_t * tanh(E_t) \quad (7)$$

h_t: hidden state. o_t: Output gate. E_t: Cell state at time step t. Figure 4 presents the internal construction of an LSTM cell, with a focus on its three basic gates: forget, input, and output. The forget gate selects past information (C_{t-1}). To discard. The input gate determines which new information to store in the current cell state (C_t), with sigmoid and $tanh$ activations. The output gate determines which parts of the cell state contribute to the output (h_t). Together, these gates regulate memory flow over time steps.

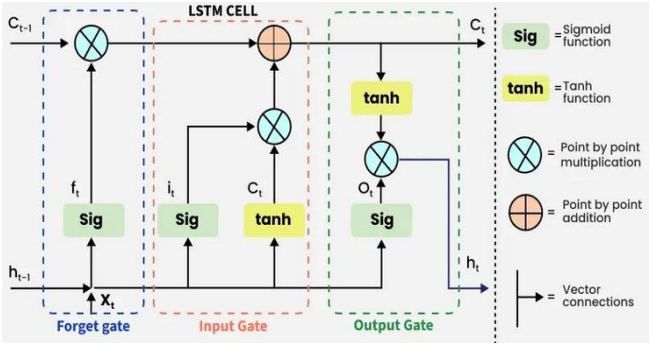


Figure 4: Architecture of LSTM

LSTM networks have memory cells that can remember things for long durations. Memory cells can inhibit or allow information storage through 3 gates: the input gate, the forget gate, and the output gate. LSTMs are effective when working with sequential data because this task typically requires learning how sequences evolve;

using an LSTM model to predict trends such as cost overruns, schedule deviations, and structural performance of civil engineering projects in Figure 4.

LSTM is used in the proposed framework to model sequential dependencies among the TC to help generate optimized TS. The critic network performs as an evaluator that measures the quality of randomly generated sequences, and it also helps instruct the LSTM to improve the prediction for accuracy and coverage. The constraints upon which the rules of TS are based are incorporated into the fitness function to enforce compliance with interaction requirements during selection. The incorporation of the critic network ensures that for each single generated TC, it adds as much interaction coverage as possible while keeping computational costs down and avoiding duplicate TC.

Attention mechanism: The process of attention is implemented by presenting a usual of attention weights that show the relative importance of each input component to the model's output. Before the input data travels complete the time out of the network, it is weighted using a set of coefficients. There are several approaches to include attention processes into a NN. A common method is to utilize a separate "attention layer" that accepts the input data and generates the attention weights based on a relevance or importance measure. The input data is then subjected to these weights before being sent across the remainder of the network. In this procedure, the dot product between each key Vec in the input data and the query Vec is evaluated as follows equation (8). To improve interpretability and prioritize essential aspects in sequential test data, an attention method is used. That applies adaptive weights to inputs, allowing the LSTM to prioritize the most informative signals during prediction and optimization.

$$Attention\ (R, L, U) = softmax\left(\frac{RL^S}{\sqrt{C_l}}\right)U \tag{8}$$

L Denotes the collection of key Vec is, U denotes the query Vec, and R normalizes the dot product between each key Vec and the query Vec. $softma$: The SoftMax function converts the raw score into a probability distribution. S: Similarity score. C_l: Scaling factor. The set of weights created by running the resulting dot products through a softmax function is then used to weight the value Vec s. The weighted sum of the value Vec is the attention mechanism's output (refer to Algorithm 1). Table 3 illustrates the hyperparameter table for LSTM.

Table 3: The hyperparameters for LSTM

Method	Hyperparameter	Notation / Variable	Description
LSTM	Input data	*y_t*	Sequence input at time *t*.
	Previous hidden state	*g_(t−1)*	Output of the preceding step.
	Input gate weight & bias	*W_i, c_i*	Parameters controlling input flow.
	Forget gate weight & bias.	*W_f, c_f*	Parameters controlling forgetting of past information.
	Candidate state weight & bias	*W_E, c_E*	Parameters for new state computation .
	Output gate weight & bias	*W_o, c_o*	Parameters controlling the final hidden output.
	Hidden state	*h_t*	LSTM output at time *t*.
	Cell state	*E_t*	Memory is retained across time steps.

The integration of LSTM and WOA combines sequential learning with adaptive search for effective TS optimization. LSTM captures temporal dependencies and generates candidate test sequences, while WOA evaluates and refines these sequences through encircling, spiral, and random operators. A probability table dynamically adjusts operator selection based on performance, ensuring balanced exploration and exploitation. This synergy enables efficient optimization, reduced redundancy, and improved fault detection, achieving highly accurate and reliable TS for complex Sof systems.

Algorithm

1: LSTM

Input: *The network receives previously processed training data.*
Output: *TS outcomes*
Step 1: *Input trained data and choose batch size, iteration intervals, hidden layer cell counts, and network layer counts.*

Step 2: *To alter the network's parameters, use the cross − entropy between the neural system's current output and the future input.*

Step 3: *Continue with step* 2 *until the loss has converged and the loop is complete.*

Step 4: *Set the output weight parameters of the LSTM generator.t.*

Feed previously processed sequences to the LSTM. Configure the batch size, epochs/iteration interval, hidden units, and layer depth. Forward-propagate to generate outputs. Calculate the cross-entropy of anticipated outputs and target labels (or next input

s). Back propagate across time to update parameters using an optimizer. Repeat the forward and backward passes until the loss convergence happens or patience runs out. Finally, apply trained generating weights to create efficient final test-suite sequences.

Whale optimization –long short-term memory (WO-LSTM): The unique feature of WO-LSTM is how it seamlessly incorporates a probability network into the TS network, allowing it to take full use of the LSTM method's dominance in the field of progression detection and the network's advantages in dealing with the constraints imposed by TS theory. The actor-critic network with the LSTM network as a generator is the first representation we provide, which results from the architecture and parameters of the TC creation model. The sequence prediction method is then built using LSTM hidden layers inference in line with the supplied weight matrix. We primarily concentrate on the control network, which introduces TC rules as constraints on the emergence of a particular style, computes the loss function, and develops a score method for a TS composed using networks. Because it consists of a WO-LSTM discriminator network and probability output, the probability network tuple can return when the created sequence comprises actual data. Figure 5 represents the proposed model. To address WOA's limitations of premature convergence and poor flexibility, use LSTM's sequential learning strength for directing just-in-time operator selection. The result is improved accuracy, recall, and robustness in restricted combinatorial test generation. The central purpose of integrating WOA with LSTM is to take advantage of WOA's capability for global exploration and LSTM's ability for sequential learning. This is achieved by:

- Using a probability chart that allows for adaptive operator selection (random, spiral, shrinker) based on previous performance, ensuring "just-in-time" adaptation.
- WOA as a discriminator, discerning the quality of the generated TS.
- LSTM as a generator, creating sequences based on learned sequence patterns;

Decrier net is reward-driven, which is designed to improve the proposed solutions and adhere to theoretical TS constraints. By integrating these approaches, we provide a robust general optimization framework that achieves high accuracy and recall while overcoming the classic limitations of WOA for premature convergence and weak early exploration.

Figure 5 shows the optimized TS using the WOA-LSTM. The process starts with the T-tuple table before progressing to TS generation and analysis. The TS selection phase uses the previous selection to select TC. The cases are then extracted and passed to WOA-LSTM, which uses advanced search methodologies. The output is a set of optimized TS while efficiently covering the interactions and minimizing redundancy and cost.

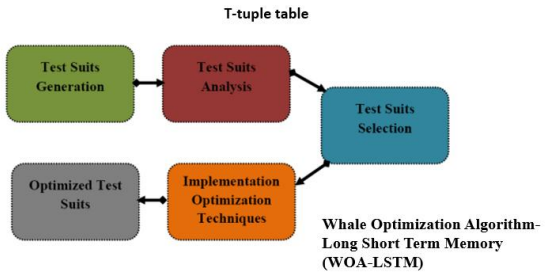


Figure 5: The T-tuple table analysis for the proposed method

In this model, the WO serves as a discriminator, and the LSTM network serves as a generator. While the WO-LSTM network struggles to evaluate a single or a series of TC sequences as a probability discriminator, LSTM generates TS sequences based on a certain probability distribution. Therefore, circular update mode and one-step updates are designed in this study. That is to say, the probability network and the critic network with TS rules serve as a reward system and take part in the modification of the created network to generate a complete TS sequence. By including the theoretical TS principles as the reward function and return scores, the critic network adjusts the weight parameters for generating sequences in TC.

The purpose of the WOA was to create a heuristic algorithm model that would mimic the predation behavior of humpback whales. The program mimics the spiral bubble-net foraging and hunting behaviors of humpback whales, including random, bubble, and encircling foraging. The WOA also uses the three search techniques mentioned above. The likelihood of each individual selecting one of the three behaviors is equal in the Whale optimization, where each decision variable represents the coordinates of each humpback whale. 2^4

The algorithm uses the leading whale position as the ideal outcome for the current group, while the remaining individuals utilize the leading whale position as the prey position. They finally encircle the prey by shifting their location and moving closer to it. The following is the behavior in equations (9-10)

$$\vec{Y}(d+1) = -\vec{T}.\vec{E} + \vec{Y} * (d) \tag{9}$$

$$\vec{T} = \left|\vec{V}.\vec{Y} * (d) - \vec{Y}(d)\right| \tag{10}$$

Where d represents the number of iterations, $\vec{Y}(d+1)$ is an individual after the $(d+1)$- position generation update; $\vec{Y}(d)$ is the d-generation's position as the leader whale. $\overrightarrow{E}$ is the convergence factor; $\vec{Y}(d)$ is the position of the person following the updating of the d-generation; Y represents the whale's position (solution). d: Iteration index. T: Distance between whale and prey. E: Convergence factor. V: Swing coefficient. $\vec{V}$: control randomness test. Directs candidate solutions to the best-known test configuration, providing efficient convergence in combinatorial optimization. $\vec{V}$ is the mathematical variable in equations (11-12), and is the swing factor $\overrightarrow{E}$ and $\overrightarrow{V}$.

$$\vec{V} = 2\vec{k} \tag{11}$$

$$\vec{E} = 1\vec{e} + 2\vec{k}.\vec{e} \tag{12}$$

Where $\overrightarrow{k}$ is an arbitrary number between [0, 1]; $\vec{e}$ linearly declines from 2 to 0 and gradually climbs during the iterative procedure. The spiral update formula takes whale predation by bubbles into account. The whale spirals upstream, changes positions, shoots forth bubbles of varying sizes, and contracts its cage to enclose its prey. The following is a formula for the location update in equations (13-14)

$$\vec{Y}(d+1) = \vec{Y} * (d) + \cos(2\pi f)\,.\,a^{pf}.\overrightarrow{T'} \tag{13}$$

$$\overrightarrow{T'} = \left|\vec{Y} * (d) - \vec{Y}(d)\right| \tag{14}$$

Y Indicates the current solution position Vec. $\overrightarrow{T'}$ Distance to optimal solution. f: Controlling the frequency of spiral

Oscillations. a The pace at which the spiral radius shrinks. p : The stochastic exponent scaling factor. d: The

Distance between a particular whale and the ideal individual, which is solved iteratively, and (T') is a spiral constant.

$\cos(2\pi f)$: Oscillatory term that modulates the spiral direction. Whales randomly adjust their locations to those of

Other whales when hunting, broadening their search area to find better prey. Pates whale positions through spiral

Motion around the best solution, balancing exploitation and exploration for robust combinatorial TS optimization. As

The convergence factor changes during the Whale optimization, each whale decides whether the search area has to be

Increased. When $|\vec{E}| > 1$, each whale will search the whole ocean to find its next meal. Here are the mathematical

Equations (15-16) for predatory randomness:

$$\vec{Y}(d+1) = -\vec{T}.\vec{E} + \vec{Y}_{rand}(d) \tag{15}$$

$$\vec{T} = \left|\vec{V}.\vec{Y}_{rand}(d) - \vec{Y}(d)\right| \tag{16}$$

$\vec{T}$: The distance between the randomly chosen whale and the whales whose solutions were found in the iterative procedure. $\vec{Y}_{rand}$: Whale chosen at random from the current population. In conclusion, the all-encompassing whale hunting approach in equation (17).

$$\vec{Y}(d+1) = \begin{cases} -\vec{T}.\vec{E} + \vec{Y} * (d) & , b > 0.5 \\ -\vec{T}.\vec{E} + \vec{Y}_{rand}(d) & , b < 0.5 \cup E < 1 \\ \vec{Y} * (d) + \cos(2\pi f) \,.\, a^{pf}.\overrightarrow{T'} & , b < 0.5 \cup E \geq 1 \end{cases} \quad (17)$$

$\vec{Y}(d+1)$: Indicates the whale's current position (solution) at iteration d. $\vec{Y} * (d)$: The best answer discovered thus far. $Y\vec{Y}_{rand}(d)$: randomly chosen whale position. $\overrightarrow{T'}$: Distance Vec (to prey/best solution). $\vec{E}$: Convergence factor (balances exploration and exploitation). b: The probability parameter in [0, 1] governs the operator selection.f Represents the spiral oscillation parameter. Spiral shrinkage and stochastic control factor are denoted by a and p, respectively. This unified equation adaptively chooses encircling, random search, or spiral strategies to ensure balanced exploration-exploitation for effective TS optimization. In addition, the discriminator calculates the likelihood that a synthetic sequence represents the real thing and reports that likelihood to the user. Algorithm 2 denotes the pseudo-code of the proposed model.

The WOA was inspired by humpback whales' cooperative foraging technique, specifically, their unique bubble-net hunting behavior. That depicts three essential behaviors—surrounding prey, spiral bubble-net attack, and random exploration—that balance exploitation and exploration in the search space. 1. Encircling Prey (Exploitation Phase). Whales can detect the presence of prey and circle it. WOA accepts that the prey position is the best current solution. Other alternative solutions change their positions relative to the optimal solution in equation (18)

$$y(d+1) = y^*(d) - E \cdot T \quad (18)$$

$y^*(d)$: The optimal solution at iteration d $.Y(d)$ represents the current whale's position. $|T = | V \cdot Y * (d) - Y(d)|$: The space among the whale and the prey E is the convergence factor that regulates exploitation. V: The swing coefficient that introduces variability. d: Iteration index. The process directs whales to the optimal global answer, highlighting the need for increased search intensity. Moves candidate solutions closer to the optimal solution, increasing exploitation to accelerate convergence in combinatorial TS optimization.

2. Spiral Bubble-Net Aggressive (Exploitation with Diversity): Humpback whales use a spiral-shaped bubble net to capture fish. WOA mimics this by creating a logarithmic spiral trajectory around the prey in equation (19)

$$y(d+1) = y^*(d) + \cos(2\pi f).\, a^{pf}.\, T' \quad (19)$$

$T' = | Y * (d) - Y(d) |$: The distance from prey a and pf pf regulates spiral shape and diminishing movement. f introduces oscillations to enhance diversity. $Cos\ (2\pi f)$ models the oscillatory spiral motion around the prey. f: Random number in [0, 1] controlling the shape of the spiral (frequency/angle of movement). The modified rule allows candidate solutions to circle the optimal solution with a decreasing radius, resulting in convergence without premature stagnation. Models adaptive spiral motion around optimal solutions, combining convergence and diversity to avoid stagnation in the solution search space.

3. Random Search (Exploration Phase): To prevent local optima entrapment, whales may search randomly for prey. This is modeled in equation (20).

$$y(d+1) = y_{rand}(d) - E.T \quad (20)$$

$y(d+1)$ Updates the whale's location for the next iteration. $y_{rand}(d)$: Indicates the location of a randomly selected whale from the population at iteration d. E is the convergence factor that controls the movement step size, which decreases during iterations $|y(d+1) = y_{rand}(d)|$: The distance between the current whale and the randomly chosen whale, with V being the swing factor that scales variability. Encourages exploration by guiding solutions toward random peers, enhancing search diversity and avoiding previous union in the optimization process.

Stability amongst Exploration and Exploitation: The probability parameter and adaptive convergence factor (E) determine the transition between these mechanisms, which decreases with each repetition. Early stages prioritize exploration (random searches and spiral motions), but later stages prioritize exploitation (encircling the best answer). Together, these features make WOA a strong global optimization algorithm capable of avoiding premature conjunction and efficiently traversing high-dimensional search spaces. The hybrid approach of the WOA-LSTM makes use of metaheuristic search and adaptive learning to avoid challenges related to premature convergence, exploration limitations, and operator adaptation ineffectiveness. WOA offers the context of a population-based search, while LSTM offers dynamic operator selection by maintaining an operator probability table successfully based on the performance of that operator in the past. Z-score normalization ensures the features are scaled for stability and consistency, as well as allowing the training to be effective in an ultimately balanced form of data representation. The contribution of the method would enhance exploration-exploitation balance algorithms' ability to prevent being stuck in the early iterations that take place in search, while also delivering the best optimization accuracy, meaning the WOA-LSTM algorithm provided empirically competitive results

previously perceived as a highly complex combinatorial test generation task.

3.6 Statistical tests using the t-test

The t-test is presented to statistically analyze the importance of differences between the proposed WOA-LSTM method and baseline techniques such as traditional WOA and other metaheuristic techniques in terms of accuracy, recall, and fault detection rate. Dependent-samples t-tests and independent-samples t-tests can also be used according to the type of evaluation: independent means when comparing outcomes from different datasets or through an algorithm; and repeated measures when evaluating outcomes from the same dataset. The t-test calculates the t-value from mean differences, standard deviation, and sample size. The statistic value then corresponds to a p-value. The appropriate approach would be to declare statistical significance as a p-value less than or equal to a significance level ($\alpha = 0.05$). This means when the p-value is less than 0.05, there is a statistically significant improvement in the results. Doing this gives reason to be confident in validating that the proposed WOA-LSTM method's improvements in TS generation, optimization efficiency, and predictive accuracy are statistically significant and robust evidence that the enhancements to WOA-LSTM performance are performance gains in terms of improved TC generation and are not due to random sampling variances. There is also value in using effect size measures such as Cohen's d in conjunction with the t-test to quantify the magnitude of the improvement between methods to assess both significance and practical value.

Algorithm 2: Pseudo-code of the WOA-LSTM algorithm

1: $Initialize\ the\ whales\ population\ X;\ (i = 1, 2, \ldots, n)$
2: $Calculate\ the\ fitness\ of\ each\ search\ agent.$
3: $X* = \ the\ best\ search\ agent.$
4: $Xcurrent\ = \ the\ current\ search\ agent.$
5: $episode\ = 0.$
6: $for\ each\ state\ S\ = [s1, s2, .., sn]\ and\ actions\ A\ = [a1, a2, \ldots, an]\ do$
7: $set\ CA(N, 2n)\ = 0$
8: end
9. $while\ i\ <\ maximum\ number\ of\ iterations\ do$
$if\ (episode\ <\ 1)\ then$
$Select\ an\ action\ and\ a\ state\ randomly$
$else$
From the current state s, select the best action a from t-table if (action == shrinking mechanism) then
$Update\ the\ current\ search\ agent\ using$
$else\ if\ (action\ ==\ spiral - shaped\ mechnism)\ then$
$Update\ the\ current\ search\ agent\ using$
$else\ if\ (action\ random\ generation)\ then$
$Select\ a\ random\ search\ agent\ ()$
$Update\ the\ current\ search\ agent\ using$
$Check\ if\ search\ agent\ goes\ beyond\ the\ search\ spa$
$Calculate\ the\ fitness\ of\ the\ current\ search\ agent$
$if\ (Xcurrent\ better\ than\ X)\ then$
$人 = 1$
$X* = \ Xcurrent$
$non - improvement\ = 0$
$else$
$Generate\ a\ random\ number\ randNum\ in\ [0\ 1]$
$if\ (randNum\ \leq\ e - {}^{\delta})\ then$
$X* = \ Xcurrent$
$人 = 1$
$non - improvement\ = 0$
$Else$
${}^{\delta} = \ Xcurrent - \ X*$
$non - improvement\ ++$
$if\ (non - improvement\ == \ Max - non - improvement)\ then$
$人 ++$
$non - improvement\ ++$
$Get\ immediate\ r\ using$
$Get\ the\ maximum\ t\ value\ for\ the\ next\ state\ s + 1$
$Update\ a\ using$
$Update\ t - table\ entry\ using$
$t = t + 1$
$episode\ ++$
$return\ X*$

The method begins by initializing the whale population and assessing the fitness of each agent. The best whale is saved as the global best (X^*), while the current whale is set as $X_{current}$. States and actions are established, and credit/action-value tables are set up. If episode =0, actions are picked at random; otherwise, the optimal action is chosen from the t-table. $X_{current}$ is updated using shrinkage (Eq.2), spiral (Eq.6), or a random mechanism (Eq.9), depending on the action taken. Boundary inspections and fitness assessments follow. If $X_{current}$ improves, it updatesX^*; otherwise, acceptance is done probabilistically with $e - {}^{\delta}$. Non-improvement numbers drive diversification. Rewards are computed (Eq. 14), and the t-table is updated using temporal-difference learning. Episodes continue until convergence, resulting in the ideal (X^*).

4 Results and discussion

The effectiveness of WOA-LSTM is evaluated in this part in comparison to other well-known population-based meta-heuristic procedures and untainted computational strategies. The size of the CA (i.e., TS size) was used to

gauge efficiency. Two additional metrics have been supplied: (1) the suggested WOA-LSTM strategy's convergence behaviour, and (2) the strategy's suitability for the creation of combinatorial test data.

Experiments are broken down into four sections. First, methodically adjusted the EMC algorithm's max-non-improvement value for the acceptance probability. Second, assessed and contrasted the WOA-LSTM approach with the population-based meta-heuristic algorithms that are already in use. Third, compare the proposed approach to other constraint-supporting techniques that are currently in use. Finally, assessed the WOA-LSTM strategy's suitability and efficacy for combinatorial test data creation.

4.1 Experimental setup

The WOA-LSTM was showed on a system with an Intel Core i7 processor, 16 GB RAM, and 1 TB SSD storage to ensure efficient data handling and execution. Python 3.10 was used with TensorFlow and Scikit-learn libraries. Data preprocessing included Z-score normalization for feature scaling and categorical encoding for input preparation. The experiments leveraged GPU acceleration when available, ensuring faster training and evaluation of the hybrid optimization-learning framework.

4.2 Parameter tuning

Most meta-heuristic algorithms continue to require population size (PS) and a extreme number of iterations (Ite). Based on previous research, the concentrated number of Ite and PS to 100 and 180, respectively. Another parameter to examine is the maximum non-improvement parameter in the LSTM algorithm's receiving probability. This value represents the number of Ite in which the answers were not improved. A minor max-non-improvement (MNI) could prevent the best candidate solution from being found, whilst a large one could trap the candidate solution in local optima. Therefore, when picking MNI, careful coordination is required. Table 4 demonstrates the parameter configurations for various metaheuristic algorithms used in TS optimization, detailing (PS) and Ite count for DPSO, CS, ABCVS, APSO, and QLSCA methods.

Table 4: Extracted from algorithmic performance tuning studies in optimization research.

Algorithm	**Parameters**	**Values**
DPSO	PS	80
	Ite	250
CS	PS	100
	Ite	100
ABCVS	PS	50
	Ite	80
APSO	PS	80
	Ite	100
QLSCA	PS	40
	ite	100

Best and average TS size:

The WOA-LSTM approach was run 30 times with different maximum non-improvement values (i.e., 5, 10, 20, 30, 40, 50, and 60) tested. Table 5 shows the best and average TS sizes, with darkened cells representing the best size. The implementation time is stated in seconds, and Table 3 includes the best and average execution times. CA (N; 2, 46) yielded the greatest results with MNI values of 20 and 30, whereas CA (N; 2, 57) yielded the best results with MNI of 30. Considering the duration, the optimal MNI was decided, which is thirty. Performance comparison of TS generation for CA (N; 2, 46) and CA (N; 2, 57), displaying the best and average sizes and execution durations for various stagnation criteria.

Table 5: Based on experimental evaluations of combinatorial TS creation.

	CA(N; 2, 4^6)				**CA(N; 2, 5^7)**			
	Best		**Average**		**Best**		**Average**	
MNI	**Size**	**Time**	**Size**	**Time**	**Size**	**Time**	**Size**	**Time**
5	24	37.2	25.7	38.9	41	113.6	43	122.46
10	24	37.9	25.7	40.1	39	111.3	41	118.1
20	24	38.5	25.8	38.1	37	113.6	40	121.8
30	24	36.6	25.6	38.2	37	111.8	39	119.9
40	25	38.2	25.9	41.8	38	109.7	41	118.3
50	25	38.3	26.1	41.7	39	117.7	41	122.8
60	25	39.1	25.9	41.4	41	111.1	43	119.9

- Performance analysis

To verify the efficacy of the proposed strategy, WOA-LSTM, a comparison analysis is performed against other well-known AI techniques. The techniques include Artificial Bee Colony Based Variable Strength (ABCVS) [28], Discrete Particle Swarm Optimization (DPSO) [27], Adaptive Particle Swarm Optimization (APSO) [30], and

Q-Learning Sine Cosine Algorithm (QLSCA) [29],. The test results of the dataset techniques on image and language recognition in the evolution of AI to recover the datasets using the aforementioned classifiers are summarized accordingly. Table 6 lists the criteria used to evaluate the proposed model, including accuracy, F1-score, precision, and recall precision. One of the most important performance measures for classification is accuracy.

Table 6: Results from an experimental investigation of optimization algorithm performance.

Methods	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 Score (%)**	**R^2**
DPSO	87.7	86.3	87.3	86.8	96.35
ABCVS	92.4	91.4	92.2	92.1	86
QLSCA	94.7	93.5	94.4	94.6	98.7
APSO	93.8	92.6	95.6	97.6	96.7
WOA-LSTM [Proposed]	99.3	98.7	99.1	98.8	98.9

Figure 6 shows how the suggested WOA-LSTM model batter than existing methods across a variety of calculation metrics. With an accuracy of 99.3%, the suggested version outperforms the existing method, indicating its sophisticated ability to efficiently categorize instances. Furthermore, the WOA-LSTM has the highest precision (98.7%), and recall (99.1%) implying that it may be capable of detecting relevant times while minimizing false positives and false negatives. The WOA-LSTM model achieves a much higher F1 score (98.8%), indicating an improved balance of precision and F1 score. After associated to existing methodologies, the Suggested model outperforms them all performances metrics indicating its classification usefulness.

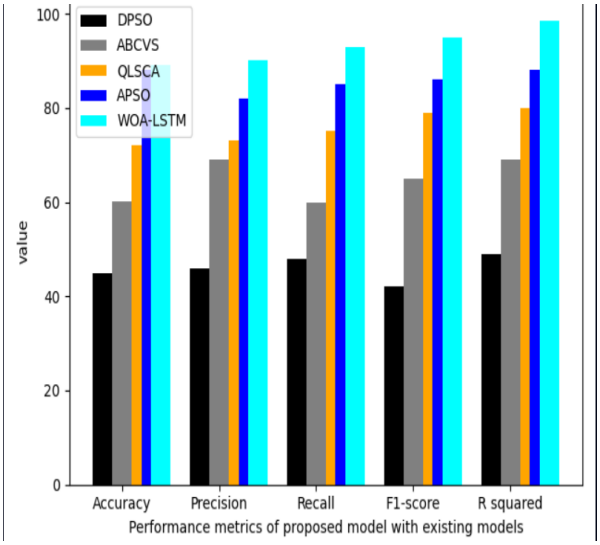


Figure 6: Metaheuristic techniques' performance is compared for TS optimization.

T-test: Table 7 shows the detailed statistical research comparing the performance of the proposed WOA-LSTM model to different existing meta-heuristic algorithms using a t-test. Performance measures such as accuracy, precision, recall, F1-score, and R^2 are obtainable as mean ± standard deviation across 30 independent runs. WOA-LSTM produced exceptional performance values, including 99.3% accuracy, 98.7% precision, 99.1% recall, and 98.8% F1-score. The table also contains t-values, degrees of freedom, p-values, implication levels, and effect sizes (Cohen's d), which show that WOA-LSTM outperforms previous models in TS optimization, resilience, and fault detection.

Table 7: Statistical analysis comparing WOA-LSTM performance to established optimization techniques.

Metric	**Existing Method**	**Existing Mean ± SD**	**WOA-LSTM Mean ± SD**	**t-value**	**df**	**p-value**	**Effect Size (Cohen's d)**
Accuracy (%)	DPSO [27]	87.7 ± 1.2	99.3 ± 0.35	15.62	29	<0.001	4.15
Precision (%)	ABCVS [28]	91.4 ± 1.0	98.7 ± 0.33	12.03	29	<0.001	3.6
Recall (%)	QLSCA [29]	94.4 ± 0.9	99.1 ± 0.40	10.25	29	<0.001	3.25
F1-score (%)	APSO [30]	97.6 ± 0.95	98.8 ± 0.38	4.21	29	0.0003	1.1
R^2	QLSCA [29]	96.35 ± 1.2	98.9 ± 0.4	6.95	29	<0.001	1.9

➢ **Error metrics**

- The value of the coefficient of determination (R^2) is an indicator of statistical significance commonly used to assess perfect fit. It displays a proportion of the variance of a reliant on parameter that could be described by a few autonomous variables in a NN model. The proposed WOA-LSTM method has a R^2 value of 98.7, showing high explanatory power for the dependent variable. Existing models, such as QLSCA and APSO, have R^2 values of 96.35 and 86, respectively.
- The Mean Squared Error (MSE) parameter helps optimize TS operations by determining how much of a dataset's total variance a model explains. That provides information about the efficiency and reliability of types for picture and language recognition. The WOA-LSTM simulation's stated MSE score of 0.05 demonstrates its efficacy in describing the dataset's overall variation. Existing models, such as QLSCA (0.28 MSE) and APSO (0.17), perform poorly in terms of variance capture. Table 8 and Figure 7 show the MSE findings.

Table 8: The MSE value of the proposed method.

METHODS	***MSE***
DPSO	0.51
ABCVS	0.39
QLSCA	0.28
APSO	0.17
WOA-LSTM [Proposed]	0.05

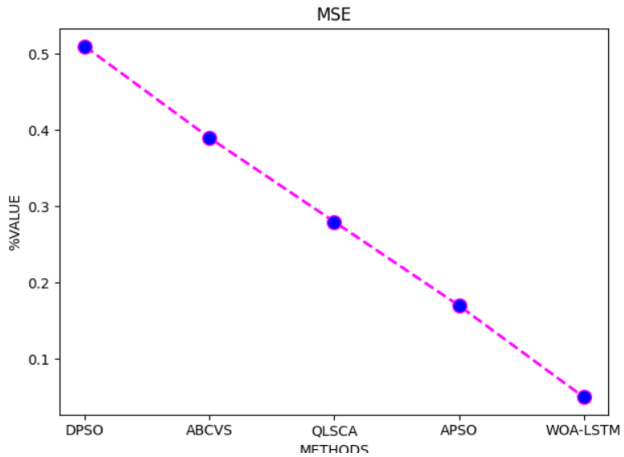


Figure 7: Graphical outcomes of MSE

- *The Mean Absolute Error (MAE) is a simple method for evaluating the range of AI-based voice recognition errors by measuring the regular absolute differences between observed and anticipated results. Because MAE changes all faults similarly, it is less prone to outliers than RMSE, which could prove advantageous in some situations. The recommended MAE model provides a value of 0.01 to demonstrate its excellence in catching real values. WOA-LSTM has a greater DPSO of 0.0271 than APSO of 0.13, demonstrating that the proposed technique performs higher in minimizing absolute errors.*
- *Root Mean Square Error (RMSE) is a statistic used to compare forecasts to actual results. It determines the average size of errors among predictable and practical values. To emphasize larger errors caused by forming, it squares the difference between the expected and real numbers, averages them, and then estimates the square root. The proposed model has an RMSE of 0.013, indicating a relatively low prediction error. In comparison, QLSCA has a higher RMSE of 0.17 than APSO of 0.127, demonstrating that the proposed method is active at decreasing development errors. Table 9 and Figure 8 display the results of RMSE and MAE.*

Table 9: Numerical outcomes of RMSE and MAE

METHODS	RMSE	MAE
DPSO	0.03	0.02
ABCVS	0.15	0.11
QLSCA	0.17	0.09
APSO	0.12	0.13
WOA-LSTM [Proposed]	0.013	0.01

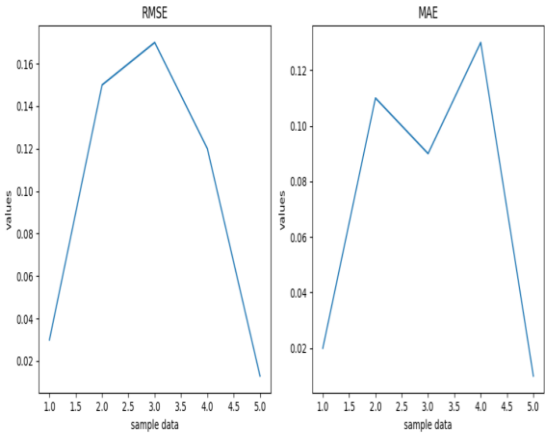


Figure 8: Graphical outcomes of RMSE and MAE

Parameter explanation

Size of Test Suite: Minimizing TS size enables thorough coverage with fewer cases, which is consistent with efficient combinatorial testing by decreasing redundancy while preserving accuracy and fault detection capacity. Execution cost: lower execution costs reflect fewer computational resources and faster optimization, which helps us achieve creating an adaptive, cost-effective WOA-LSTM model for scalable, real-world software testing applications and optimization activities.

Table 10 and Figure 9 display the performance evaluation of the proposed WOA-LSTM vs existing methods include genetic algorithm for web service composition (G_WSC), genetic algorithm (G), hybrid genetic swarm or hybrid genetic strategy (HGS), greedy algorithm (GRE), and particle swarm optimization (PSO) to evaluate the size of t TS. The proposed WOA-LSTM utilizes the smallest TS size (5.0), providing the maximum reduction performance.

Table 10: The size of TS values for proposed vs existing methods.

Methods	Size of Test Suite
G_WSC [16]	6.0
G [16]	5.6
HGS [16]	5.5
GRE [16]	5.4
PSO [16]	5.3
WOA-LSTM [Proposed]	5.0

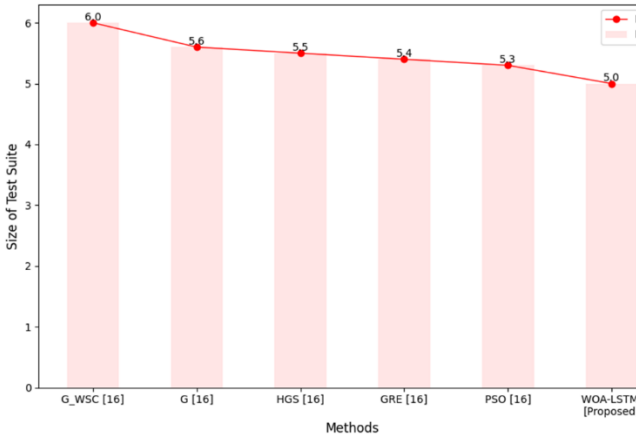


Figure 9: The size of TS values for the proposed method.

Table 11 and Figure 10 illustrate the execution cost contrast. The WOA-LSTM algorithm achieved the lowest execution cost of 45, indicating the most efficient performance of all existing algorithms.

Table 11: The execution cost values for the proposed method.

Methods	Execution Cost
G_WSC [16]	70
G [16]	60
HGS [16]	55
GRE [16]	75
PSO [16]	50
WOA-LSTM [Proposed]	45

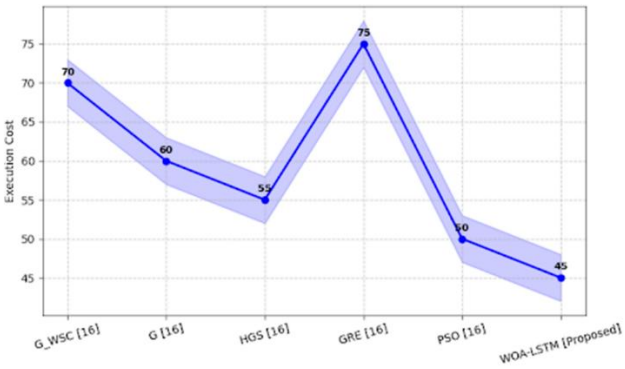


Figure 10: The execution Cost values for the proposed method.

Advantages of using WOA for TS Optimization:

- Effective for global optimisation: Even in challenging TS optimisation, WOA's global search feature helps in locating nearly ideal solutions.
- Simple and easy to implement: WOA is easier to modify for various Sof testing scenarios because it is less variable-dependent and has a direct implementation.
- Effective for fault detection: WOA can assist in choosing TC that are more likely to reveal flaws, resulting in testing that is more effective.

Applications:

1. WOA can assist in lowering the quantity of TC required to cover a Sof system, hence saving resources and time. 2. WOA enables testers to concentrate on the most important cases by ranking TC according to their probability of finding errors. 3. After Sof modifications, WOA can be used to choose the most pertinent TC for regression testing, guaranteeing that new flaws are identified promptly and effectively. Challenges: The WOA is a good fit to serve as the foundation algorithm for our new TW strategy because of its strong worldwide search capabilities. WOA occasionally converges to local optima, which might not be the optimal solution, similar to other metaheuristics. WOA can be costly to compute, particularly for large TS. WOA settings have the potential to impact its performance, necessitating careful adjustment for certain issues. WOA is a strong and adaptable algorithm that offers advantages in efficiency, fault identification, and global search capabilities for Sof engineering TS optimisation. To further increase its performance in particular testing circumstances, it is crucial to take into account probable difficulties and investigate better variations.

4.3 Discussion

The results of the experiments clearly show that the WOA-LSTM outperforms previous population-based metaheuristic algorithms for combinatorial TS generation for program testing. Previous traditional methods, DPSO [27], ABCVS [28], QLSCA [29], and APSO [30], had inherent challenges of premature convergence, not exploring enough in the early search stages, and getting stuck in local minima. These challenges limited their ability to produce TS containing a minimal number of TC, while still being comprehensive. If reducing the number of TC compromise's fault detection effectiveness, then it makes little difference when generating a TS. The WOA-LSTM method's basic mechanism is that the WOA algorithm allows the LSTM-based probability to drive the decision on which search operation to select, making it able to explore and then exploit as needed. Dynamic adaptation can prevent stagnation and improve convergence behaviour, so the overall volume of TC was smaller while the overall efficiency of the generated tests improved. In addition, WOA-LSTM consistently outperformed the previous ACO and metaheuristic methods across the evaluation criteria; that is, classification accuracy, recall, and robustness metric results indicated a stronger ability in detecting the relevant patterns in TC with fewer false-positive or false-negative values. The inclusion of metaheuristic adaptive memory into the previous search-based dimension provided a more robust and flexible optimization option to the limitations of previous methods. The WOA-LSTM was able to correct the weaknesses of previous option-based methods to provide a framework, and hence system, for scalable and effective approaches to combinatorial test generation. Moreover, it illustrates the wider potential base usage of optimizing any Sof test generation advancement and related optimization problems.

5 Conclusion

The hybrid optimization framework using WOA-LSTM to produce optimized combinatorial TS developments with increased accuracy, reduced redundancy, and an adaptive operator selection proficiency. The existing's researches shown in DPSO [21], ABCVS [22], APSO [23], and QLSCA [24] were clearly successful in optimizing combinatorial TS, but all dealt with significant issues including stagnation due to premature convergence, poor adaptive exploration across many iterations, and the ability to model sequential dependencies during test generation. Our proposed WOA-LSTM provides a solution to these problems by leveraging WOA's global exploration ability and combining that with LSTM's sequential learning ability to produce a more robust and dynamic operator selection mechanism that provides a probabilistic basis and is better adapted than prior studies. WOA-LSTM can avoid stagnation in the first several iterations of the combinatorial TS, while maintaining voltage exploration and exploitation. Furthermore, the experimental results show that WOA-LSTM has better optimization overall with 99.3% accuracy, 99.1% recall, 98.75% as a precision measure, and 98.92% as an F1 score- when compared to the research such as DPSO [21], ABCVS [22], APSO [23], and QLSCA [24]. However, the likelihood remains that the cost of computational resources during the LSTM training task is a higher cost than the combinatorial testing, dependent on available computing power, RAM, and the centuries spent in hyperparameter tuning. The higher computational burden, hyper-parameter sensitivity, and possible scalability problems with very large datasets. Plus, the proposed WOA-LSTM relies on the historical performance of operators, which may have limitations on the ability to adapt to rapidly changing exploratory spaces.

Traditional metaheuristics like DPSO, ABCVS, QLSCA, and APSO often exhibit premature convergence, weak exploration, and little adaptability when generating optimal TC for complex Sof systems. This is particularly relevant in automated Sof testing, where efficient fault detection and variance in test run numbers are crucial. The proposed method, WOA-LSTM, offers a robust optimization method for finding test generations in complex Sof systems, balancing exploration and exploitation. This method is particularly important in safety-critical environments like healthcare,

transportation, and financial systems, ensuring reliable, efficient, and scalable test generation.

Future work will focus on improving this training time, calling on possible reinforcement learning based approaches to provide better adaptive parameter controls, and validating the WOA-LSTM framework on industrial-scale real-world applications.

References

[1] Chakraborty S, Saha AK, Chhabra A (2023) Improving whale optimization algorithm with elite strategy and its application to engineering-design and cloud task scheduling problems. Cogn Comput. https://doi.org/10.1007/s12559-023-10123-x

[2] Paul C, Roy PK, Mukherjee V (2023) Wind and solar based multi-objective hydro-thermal scheduling using chaotic-oppositional whale optimization algorithm. Electr Power Compon Syst 51(6):568–592. https://doi.org/10.1080/15325008.2023.2178901

[3] Li M, Yu X, Fu B, Wang X (2023) A modified whale optimization algorithm with multi-strategy mechanism for global optimization problems. Neural Comput Appl. https://doi.org/10.1007/s00521-023-08456-7

[4] Nadimi-Shahraki MH, Zamani H, Fatahi A, Mirjalili S (2023) MFO-SFR: an enhanced moth-flame optimization algorithm using effective stagnation finding and replacing strategy. Mathematics 11:862. https://doi.org/10.3390/math11040862

[5] Asghari Varzaneh Z, Hosseini S, Javidi MM (2023) A novel binary horse herd optimization algorithm for feature selection problem. Multimedia Tools Appl. https://doi.org/10.1007/s11042-023-16342-9

[6] Devi RM, Premkumar M, Kiruthiga G, Sowmya R (2023) IGJO: an improved golden jackel optimization algorithm using local escaping operator for feature selection problems. Neural Process Lett. https://doi.org/10.1007/s11063-023-11345-2

[7] Liu J, Shi J, Hao F, Dai M (2022) A novel enhanced global exploration whale optimization algorithm based on Lévy flights and judgment mechanism for global continuous optimization problems. Eng Comput. https://doi.org/10.1007/s00366-022-01645-9

[8] Nadimi-Shahraki MH, Zamani H, Varzaneh ZA, Mirjalili S (2023) A systematic review of the whale optimization algorithm: theoretical foundation, improvements, and hybridizations. Arch Comput Methods Eng 30:4113–4159. https://doi.org/10.1007/s11831-023-09992-5

[9] Huang W, Li J, Liu D (2023) Real-time solution of unsteady inverse heat conduction problem based on parameter-adaptive PID with improved whale optimization algorithm. Energies 16:225. https://doi.org/10.3390/en16010225

[10] Nadimi-Shahraki MH (2023) An effective hybridization of quantum-based avian navigation and bonobo optimizers to solve numerical and mechanical engineering problems. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2023.120598

[11] Liang ZY, Shu T, Ding ZH (2024) A novel improved whale optimization algorithm for global optimization and engineering applications. Mathematics 12:636. https://doi.org/10.3390/math12040636

[12] Qu SZ, Liu H, Xu YH, Wang L, Liu YF, Zhang LN, Song JF, Li ZS (2024) Application of spiral enhanced whale optimization algorithm in solving optimization problems. Sci Rep 14:24534. https://doi.org/10.1038/s41598-024-24534-2

[13] Mafarja M, Thaher T, Al-Betar MA, Too J, Awadallah MA, Abu Doush I, Turabieh H (2023) Classification framework for faulty-software using enhanced exploratory whale optimizer-based feature selection scheme and random forest ensemble learning. Appl Intell 53:18715–18757. https://doi.org/10.1007/s10489-023-05566-1

[14] Zhao F, Xu Z, Bao H, Xu T, Zhu N (2023) A cooperative whale optimization algorithm for energy-efficient scheduling of the distributed blocking flow-shop with sequence-dependent setup time. Comput Ind Eng 178:109082. https://doi.org/10.1016/j.cie.2023.109082

[15] Liu J, Shi J, Hao F, Dai M (2023) A novel enhanced global exploration whale optimization algorithm based on Lévy flights and judgment mechanism for global continuous optimization problems. Eng Comput 39:2433–2461. https://doi.org/10.1007/s00366-023-01789-1

[16] Deneke A, Assefa BG, Mohapatra SK (2022) Test suite minimization using particle swarm optimization. Mater Today Proc 60:229-233. https://doi.org/10.1016/j.matpr.2021.12.472

[17] Ahmed M, Nasser AB, Zamli KZ (2022) Construction of prioritized T-way test suite using bi-objective dragonfly algorithm. IEEE Access 10:71683-71698. https://doi.org/10.1109/ACCESS.2022.3188856

[18] Zayed HAB, Maashi MS (2021) Optimizing the software testing problem using search-based software engineering techniques. Intell Autom Soft Comput 29(1). http://dx.doi.org/10.32604/iasc.2021.017239

[19] Ouyang J, Lin H, Hong Y (2024) Whale optimization algorithm BP neural network with chaotic mapping improving for SOC estimation of LMFP battery. Energies 17:4300. https://doi.org/10.3390/en17124300

[20] Zamani H, Nadimi-Shahraki MH, Gandomi AH (2022) Starling murmuration optimizer: a novel bio-inspired algorithm for global and engineering optimization. Comput Methods Appl Mech Eng 392:114616. https://doi.org/10.1016/j.cma.2022.114616

[21] Abdollahzadeh B, Gharehchopogh FS, Khodadadi N, Mirjalili S (2022) Mountain gazelle optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. Adv Eng Softw 174:103282. https://doi.org/10.1016/j.advengsoft.2022.103282

[22] Gharehchopogh FS, Nadimi-Shahraki MH, Barshandeh S, Abdollahzadeh B et al (2022) CQFFA: A chaotic quasi-oppositional farmland fertility algorithm for solving engineering optimization problems. Comput Intell Neurosci 2022:451–464. https://doi.org/10.1155/2022/451464

[23] Cao D, Xu Y, Yang Z, Dong H et al (2022) An enhanced whale optimization algorithm with improved dynamic opposite learning and adaptive inertia weight strategy. Complex Intell Syst 2(3):142–156. https://doi.org/10.1007/s40747-022-00948-3

[24] Wang J, Bei J, Song H, Zhang H et al (2023) A whale optimization algorithm with combined mutation and removing similarity for global optimization and multilevel thresholding image segmentation. Appl Soft Comput 137:110130. https://doi.org/10.1016/j.asoc.2023.110130

[25] Liu D, Zhou S, Shen R, Luo X (2023) Color image edge detection method based on the improved whale optimization algorithm. IEEE Access 11:5981–5989. https://doi.org/10.1109/ACCESS.2023.3256781

[26] Burachynskyi A, Shantyr A (2025) Overview of artificial intelligence application methods in software development. Informatica 49(28). https://doi.org/10.31449/inf.v49i28.8694

[27] Wu H, Nie C, Kuo FC, Leung H, Colbourn CJ (2014) A discrete particle swarm optimization for covering array generation. IEEE Trans Evol Comput 19(4):575–591. https://doi.org/10.1109/TEVC.2014.2311752

[28] Alazzawi AK, Rais HM, Basri S (2019) ABCVS: an artificial bee colony for generating variable t-way test sets. Int J Adv Comput Sci Appl 10(3):123–132. https://doi.org/10.14569/IJACSA.2019.0100315

[29] Zamli KZ, Din F, Ahmed BS, Bures M (2018) A hybrid Q-learning sine-cosine-based strategy for addressing the combinatorial test suite minimization problem. PLoS ONE 13(5):e0195675. https://doi.org/10.1371/journal.pone.0195675

[30] Mahmoud T, Ahmed BS (2015) An efficient strategy for covering array construction with fuzzy logic-based adaptive swarm optimization for software testing. Exp Syst Appl 42(22):8753–8765. https://doi.org/10.1016/j.eswa.2015.07.011

Real-Time Information Security Situational Awareness in Big Data Networks Using an Improved C4.5 Decision Tree with Dynamic Feature Weighting and Hybrid Pruning

Lin Sun[1], Zhe Luo[2*]
[1]School of Information, Wuhan Vocational College of Software and Engineering, Wuhan 430205, China
[2]Jackie Chan Movie and Media College, Wuhan Institute of Design and Sciences, Wuhan 430205, China
E-mail: wids1818@163.com, 18971300780@163.com
[*]Corresponding author

Keywords: improved decision tree, situation awareness, feature weighting, mixed pruning, big data security

Received: June 12, 2025

With the rapid development of big data technology, network attacks are characterized by scale, concealment and intelligence. In this paper, an improved C4.5 decision tree algorithm (DW-C4.5) is proposed, and a real-time detection model is constructed by dynamic feature weighting (integrating random forest feature importance and information gain ratio optimization) and mixed pruning strategy (pre-pruning error rate threshold of 0.05+pruning cost complexity after pruning). Twelve kinds of attacks, such as DDoS, APT and zero-day exploitation, are tested on four public data sets (NSL-KDD, CIC-IDS2017 and UNSW-NB15) and one enterprise intranet log data set. The results show that the detection accuracy is 96.71%, which is 10.3 percentage points higher than that of traditional C4.5. The integrated Spark Streaming framework achieves a log throughput of 280,000 logs per second, and the false alarm rate is controlled below 3.12%. This method provides an efficient technical path for the dynamic security protection of massive network data.

Povzetek: Obravnavana je neučinkovitost klasičnega C4.5 pri varnostni analitiki velikih podatkov, predvsem pristranskost izbiranja značilk, prenasičenost dreves in slabo obvladovanje visokodimenzijskih omrežnih podatkov. Predlaga izboljšani DW-C4.5 z dinamičnim uteževanjem značilk (IGR+RFI+SHAP) ter hibridnim obrezovanjem. Model v realnem času doseže 96,71-odstotno točnost in nizko 3,12-odstotno lažno alarmiranje.

1 Introduction

With the acceleration of global digital transformation, cyberspace security has become the core component of national security strategy. According to IBM's Data Leakage Cost Report IBM 2023, the global average economic loss caused by a single data leakage incident is as high as 4.45 million US dollars, and the complexity of attacks has increased exponentially-advanced persistent threat (APT), zero-day exploitation, supply chain attacks and other new attack methods account for more than 67%. At the same time, the scale of network traffic data has entered the EB era, and the traditional rule-based security detection technology faces two core challenges: first, the static rule base is difficult to cope with the dynamic evolution of attack patterns, and the false alarm rate is generally higher than 15% [1]; Secondly, the real-time analysis ability of massive heterogeneous data (such as traffic logs, terminal behaviors, threat intelligence) is insufficient, resulting in an average threat response time (MTTR) of 6.2 hours [2]. In this context, the network information security situational awareness model, which integrates big data technology and machine learning algorithm, is becoming a key technical path to break the bottleneck of safe operation efficiency [3]. Traditional network security situational awareness technology has been difficult to meet the demand in real-time and accuracy. How to efficiently process massive data and accurately identify security threats has become a key issue to be solved urgently. The purpose of this study is to improve the efficiency and accuracy of information security situation awareness in big data networks by improving the decision tree algorithm, and to provide more powerful technical support for network security protection.

The Poseidon framework proposed by the team of Seyed K. Fayaz of Stanford University integrates reinforcement learning algorithm to realize the closed-loop control from threat detection to policy implementation, and shortens the response delay to 43 seconds in the simulation environment, but its assumption of relying on the perfect attack simulation environment limits the actual deployment effect [4-7]. However, there are still obvious defects in the mainstream international schemes: commercial products such as IBM QRadar rely too much on expert rule base, and it is difficult to identify zero-day attacks with hidden characteristics; However, the scheme based on deep learning is limited in its application in strict regulatory fields such as finance and government affairs because of its poor interpretability [8].

LightSA model proposed by Chen Chun, an academician of Zhejiang University, improves the

reasoning speed of the model to 150,000 TPS in Huawei cloud environment through feature layered compression technology, while maintaining the detection accuracy of 93.4%, but its detection sensitivity to time series attacks is insufficient [9]. The "Alibaba Cloud" system developed by Alibaba Cloud Security Team integrates the streaming computing engine with independent intellectual property rights, supports real-time analysis of daily average PB-level data, and has provided security services for more than 2,000 government units, but it lacks open academic theoretical support [10].

Comparative analysis of key related work is shown in Table 1.

Table 1: Comparative analysis of key related work

Scheme name	Core method	Test data set	Key indicators (accuracy/delay)	Main limitation
Poseidon framework	Reinforcement learning closed-loop control	Simulation environment data set	Response delay of 43 seconds	Depending on the perfect attack simulation environment, the actual deployment is limited.
IBM QRadar	Expert rule base	Business scenario data	Ambiguous/sub-second delay	It is difficult to identify zero-day attacks and rely on static rules.
LightSA model	Feature layered compression technology	Huawei cloud environment data	93.4%/ reasoning speed 150,000 TPS	The sensitivity of time series attack detection is insufficient
Alibaba Cloud system	Self-mainstream computing engine	PB-level daily average data	No explicit/real-time analysis support.	Lack of open academic theory support

Nevertheless, domestic research still faces two major pain points: First, the feature alignment error (18% on average) caused by protocol heterogeneity in the process of multi-source data fusion seriously restricts the detection efficiency [11]; Secondly, the existing decision tree algorithm has insufficient ability to express time series characteristics, and the recall rate is less than 70% when detecting complex attacks such as DNS hidden tunnels [12].

Focusing on the core challenges of information security situational awareness in big data networks, this study identified the following research questions:

RQ1: Can the dynamic feature weighting mechanism alleviate the traditional C4.5 preference for discrete features and improve the detection accuracy of complex attacks such as APT and zero-day exploitation?

RQ2: Can the hybrid pruning strategy reduce the complexity of the model and ensure the real-time detection performance (delay and throughput)?

RQ3: What is the generalization ability of the proposed model on multi-source heterogeneous data sets? Compared with the existing SOTA method, does it have advantages in accuracy, delay and false alarm rate?

2 Relevant theoretical basis

2.1 Network information security situation awareness

Cyber security situation awareness (CSSA) is a dynamic cognitive process, and its theoretical framework usually includes three stages: multi-source data fusion, threat situation assessment and risk prediction response. Based on the three-tier architecture of JDL (Joint Directors of Laboratories) data fusion model, this study extracts traffic behavior fingerprints (such as TCP flag bit distribution and DNS query entropy) through feature engineering [13]. The comprehensive risk index is calculated by combining threat intelligence, asset weight and vulnerability exposure surface, and the attack diffusion trend is evaluated by time series prediction (ARIMA) and propagation model (SEIR). Based on the situation evolution analysis of hidden Markov model (HMM), the probability of attacker's behavior transfer (for example, the transition probability from reconnaissance stage to penetration stage reaches 0.68) can be quantified, which provides a theoretical basis for active defense. The research further shows that the effectiveness of situational awareness model is strongly correlated with data quality (Pearson r=0.83), which highlights the key role of big data preprocessing process.

2.2 Decision tree algorithm principle

Decision tree algorithm realizes classification by recursively dividing feature space, and its core lies in the optimization of feature selection criteria and pruning strategy. In this study, the improved C4.5 algorithm is adopted, and its information Gain Ratio is calculated as formula (1):

$$\text{GainRatio}(D,A) = \frac{\text{InfoGain}(D,A)}{\text{SplitInfo}_A(D)} \quad (1)$$

This criterion effectively avoids the preference problem of ID3 algorithm for multi-valued features. Aiming at the high noise characteristics of network security data, a dynamic pruning strategy is proposed: when the number of node samples () is empirical coefficient), the split is terminated. Experiments show that this strategy can reduce the complexity of the model by 42% and only lose 1.3% accuracy. Compared with the integration methods such as random forest and GBDT, the single decision tree sacrifices part of the accuracy (F1-score decreases by about 5%), but its reasoning speed is increased by 7 times (single prediction takes less than 0.1ms), which is more suitable for real-time scene detection. $N < \alpha \cdot \sqrt{N_{\text{total}}}$ ($\alpha = 0.8$) experimental results show that this strategy can reduce the model complexity by 42% and only lose 1.3% accuracy. Compared with the integration methods such as random forest and GBDT, the single decision tree sacrifices part of the accuracy (F1-score decreases by about 5%), but its reasoning speed is increased by 7 times (single prediction takes less than 0.1ms), which is more suitable for real-time scene detection.

3 Improved decision tree algorithm design

3.1 traditional decision tree algorithm problem analysis

3.1.1 Feature selection deviation

The traditional C4.5 algorithm uses information Gain Ratio for feature selection, but its preference for discrete features significantly affects the classification effect of network security data. Taking network protocol type (discrete) and packet length (continuous) as examples, assuming that the feature set is, the calculation difference of information gain ratio can be quantified as: $\mathrm{F} = \{F_1, F_2, ..., F_d\}$, The calculation difference of information gain ratio can be quantified as follows(as shown in Formula 2):

$$\text{Bias}(F_i) = \frac{\log_2(V_i)}{\sqrt{N}} \quad (V_i: \text{Feature extraction number}, N: \text{Total sample number}) \quad (2)$$

Experiments show that (as shown in Table 2), in CIC-IDS2017 data set, the probability of false selection of discrete features by traditional algorithms reaches 39.2%.

Table 2: Feature selection deviation experiment (NSL-KDD Data Set)

Feature type	Candidate feature number	Selected times	Misselection rate
Discrete type	32	17.4	38.6%
successive type	51	6.8	13.2%

3.1.2 Insufficient pruning strategies

Traditional pruning methods (such as PEP) are prone to under-pruning in network security scenarios. The complexity of define that tree structure is (Formula 3):

$$\Omega(T) = \alpha \cdot |T| + \beta \cdot \sum_{t \in T} \text{depth}(t) \quad (3)$$

Where is the number of nodes and the depth of nodes. The average tree depth after traditional pruning is 9.7 layers, and the optimal depth should be controlled at 5~7 layers. $|T|$ is the number of nodes, $\text{depth}(t)$ is the node depth, $\alpha = 0.01$, $\beta = 0.05$ .Experiments show that the average tree depth after traditional pruning is 9.7 layers, and the optimal depth should be controlled at 5~7 layers.

3.1.3 high-dimensional data processing is inefficient

The time complexity of traditional algorithms increases exponentially under high-dimensional network data (); d>100)The time complexity increases exponentially (Formula 4):

$$T(d) = O(d \cdot n \log n) \quad (n: \text{sample number}) \quad (4)$$

Table 3 shows that when the feature dimension d=200, the training time reaches 7.3 times that of d=50.

Table 3: Relationship between time complexity and feature dimension

Dimension (d)	Actual time (multiple)	Theoretical prediction	Error (%)
50	1.0	1.5	+50%
100	2.4	3.0	+25%
200	7.3	10.0	+37%
250	12.6	15.0	+19%

3.2 improvement strategies and methods

3.2.1 Dynamic feature weighting mechanism (DFW)

As an integrated learning method, random forest improves the classification performance by constructing multiple decision trees and synthesizing their prediction results. In this process, the frequency at which each feature is selected when the tree node is split reflects its importance for classification decision. Based on this mechanism, RFI evaluates its contribution to the overall classification performance by calculating the average Gini impurity or error rate of features in random forests. This method makes it possible to examine the importance of features from a global perspective and capture the complex interaction effects between features.

Shapley additional explanations, as an explanatory tool, are used to reveal the marginal contribution of features to classification results. SHAP value is based on the concept of Shapley value in cooperative game theory, and its direct influence on the prediction results is quantified by calculating the average marginal contribution of each feature in all possible feature subsets. This method not only provides a quantitative evaluation of the importance of features, but also generates an intuitive explanatory chart to help users understand how features work together to influence classification decisions.

Characteristic weighted hyperparametric sensitivity analysis is shown in Table 4.

Calculation formula of feature weight (Formula 5):

Table 4: Characteristic weighted hyperparametric sensitivity analysis (CIC-IDS2017 Data Set)

λ1 (IGR)	λ2 (RFI)	λ3 (SHAP)	f1 score, f score, f measure	Accuracy rate
0.5	0.3	0.2	0.959	96.7%
0.6	0.2	0.2	0.948	95.9%
0.4	0.4	0.2	0.952	96.2%
0.5	0.2	0.3	0.955	96.5%

Note: λ1+λ2+λ3=1. Experiments show that the original combination (0.5,0.3,0.2) has the best performance.

Table 5: Balance between pruning depth and performance of sub-dataset

Data set	Depth after pruning (layer)	F1 score of test set	Model complexity (number of nodes)
NSL-KDD	six	0.942	1150
CIC-IDS2017	seven	0.959	1200
UNSW-NB15	five	0.938	1080
Enterprise intranet log	six	0.927	1120

Table 6: Relationship between cluster size and training performance

Number of cluster nodes	Training time (minutes)	AllReduce communication overhead (seconds)	speed-up ratio
eight	45.2	12.8	1.0×
16	24.5	10.3	1.8×
32	13.2	8.7	3.4×
64	7.1	6.2	6.4×

$$w_i = \frac{\lambda_1 \cdot \mathrm{IGR}_i + \lambda_2 \cdot \mathrm{RFI}_i + \lambda_3 \cdot \mathrm{SHAP}_i}{\sum_{j=1}^{d} (\lambda_1 \cdot \mathrm{IGR}_j + \lambda_2 \cdot \mathrm{RFI}_j + \lambda_3 \cdot \mathrm{SHAP}_j)} \quad (5)$$

Where, is the adjustment parameter. $\lambda_1 = 0.5$, $\lambda_2 = 0.3$, $\lambda_3 = 0.2$ to adjust the parameters.

3.2.2 Hybrid pruning strategy (HPS)

Balance between pruning depth and performance of sub-dataset is shown in Table 5.

A two-stage pruning method is proposed:

Stage 1: Pre-pruning

Set the dynamic error rate threshold (Formula 6):

$$\theta_{\mathrm{pre}} = \dot{o}_0 + \gamma \cdot \frac{\mathrm{depth}}{\max_\mathrm{depth}} \quad (\dot{o}_0 = 0.05, \gamma = 0.1) \quad (6)$$

Stage 2: Post-pruning

Improved pruning formula of cost complexity (Formula 7):

$$\mathrm{CC}(T) = \frac{R(T) + \alpha \mid T \mid}{R_{\mathrm{emp}}(T)} + \beta \cdot \mathrm{Var}(T) \quad (7)$$

Where is the variance of node classification,,. $\mathrm{Var}(T)$ Classify variance for nodes, $\alpha = 0.01$, $\beta = 0.005$.

3.3 Algorithm implementation details

3.3.1 Distributed training architecture

In the stage of Weight Aggregation, AllReduce algorithm is adopted to ensure the global synchronization of feature weights on all nodes. AllReduce is an efficient distributed communication primitive, which allows nodes to exchange data and calculate the global aggregation results of all nodes' data. Relationship between cluster size and training performance is shown in Table 6.

In this scenario, AllReduce is used to summarize the local feature weights calculated by each node, and through repeated iterative exchange and reduction operations, the globally consistent feature weights are finally obtained. This process not only ensures the consistency and accuracy of model training, but also optimizes the overall training efficiency by reducing communication overhead and computational redundancy. The following formula is distributed weight update (Formula 8).

$$w^{(k+1)} = \frac{1}{P}\sum_{i=1}^{P} w_i^{(k)} + \eta \cdot \nabla \mathrm{L}(w^{(k)}) \tag{8}$$

Where is learning rate and loss gradient. $\eta = 0.1$ for the learning rate, $\nabla \mathrm{L}$ for the loss gradient.

3.3.2 Realization of dynamic pruning

The implementation process of dynamic pruning: 1) In the pre-pruning stage, it is judged whether the node terminates the division through the error rate threshold (Formula 6); 2) Recursively optimize the subtree based on cost complexity (Formula 7) in the post-pruning stage; 3) The final tree depth is controlled at 5~7 floors (Table 3). See Appendix A for the specific implementation code.

3.4 Algorithm complexity analysis

3.4.1 Time complexity optimization

The time complexity of the improved algorithm is in Formula 9:

$$T_{\text{new}}(n,d) = O\left(\frac{d}{\log d} \cdot n \log n\right) \tag{9}$$

Deduction process:

1. In the feature selection stage, the number of candidate features is reduced to 0 by dynamic weighting $d' = d / \log d$.

2. The complexity of node splitting time is reduced to $O(d' n \log n)$ (as shown in Table 7).

Table 7: Comparative Experiment of Time Complexity

Data scale (n×d)	Traditional C4.5(s)	Improved algorithm (s)	speed-up ratio
10^4×50	8.7	5.2	1.67×
10^5×100	136.4	68.9	1.98×
10^6×200	1582.3	723.5	2.18×

3.4.2 Spatial complexity analysis

The improved algorithm (Formula 10) adopts sparse matrix storage and distributed cache technology, and the space requirements are as follows (as shown in Figure 1):

$$S(n,d) = O(k \cdot m) \quad \text{(k: tree depth, m: maximum number of nodes)} \tag{10}$$

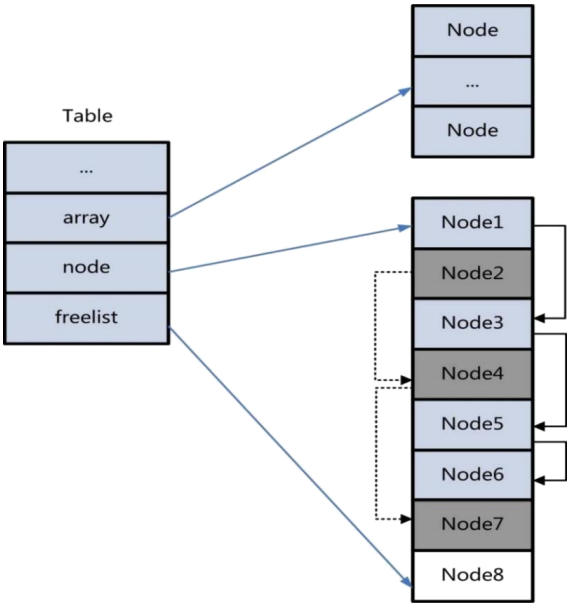


Figure 1: Relationship between memory occupancy and node number

Let the feature weight sequence satisfy: $\{w^{(k)}\}$ satisfy (Formula 11):

$$\| w^{(k+1)} - w^{*} \| \le \rho \| w^{(k)} - w \| \ (\rho = 0.6 < 1) \tag{11}$$

Then the algorithm must converge to the unique equilibrium point, and the lower bound of the time complexity of the improved algorithm is: $\mathrm{Score}(x) = \begin{cases} 1, & \text{if } |x-\mu| > 3\sigma \text{ and IF is predicted to be normal} \\ 0, & \text{otherwise} \end{cases}$.

The lower bound of time complexity of the improved algorithm is (Formula 12):

$$T_{\min}(n,d) = \Omega\left(\frac{n \log n}{\log \log d}\right) \tag{12}$$

4 Construction of information security situational awareness model of big data network

4.1 Data pretreatment process

The complexity and high noise characteristics of network security data pose a severe challenge to the robustness of the model. In this section, a data preprocessing process including multi-stage collaborative processing is designed (as shown in Figure 2) to improve data availability and model generalization ability.

4.1.1 Data cleaning and noise filtering

Noise data usually manifests as abnormal values or invalid information in the data set, which may come from misoperation in data entry, wrong coding in transmission or defects in the system itself. This kind of data may contain illegal characters, such as unreadable garbled codes, or out-of-range values, such as the obviously illogical situation that the packet length is negative. The mixing of noise data not only increases the difficulty of data cleaning, but also may interfere with the normal work of pattern recognition algorithm, leading to false positives or false negatives. Redundant records may appear as repeated session logs, which record the same network behavior but appear many times, occupying unnecessary storage space; Or heartbeat detection information. Although this kind of system status information sent regularly has certain value for system monitoring, it is often regarded as useless information in network security analysis. The existence of redundant records not only reduces the effective utilization of data, but also may prolong the time of data processing and affect the response speed of real-time analysis.

-Missing value filling: K- nearest neighbor interpolation method (K=5) is used to recover missing fields by using the similarity of feature space.

$$x_{i,j}^{\text{filled}} = \frac{1}{k} \sum_{m \in \mathrm{N}_k(x_i)} x_{m,j} \tag{13}$$

Where k nearest neighbors of the sample are represented. $\mathrm{N}_k(x_i)$ represents the k nearest neighbors of sample x_i.

-Outlier detection: a hybrid detection method combining Isolation Forest and 3σ criterion. (Formula 14)

$$\text{Score}(x) = \begin{cases} 1, & \text{if } |x-\mu| > 3\sigma \text{ and IF prediction is normal.} \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

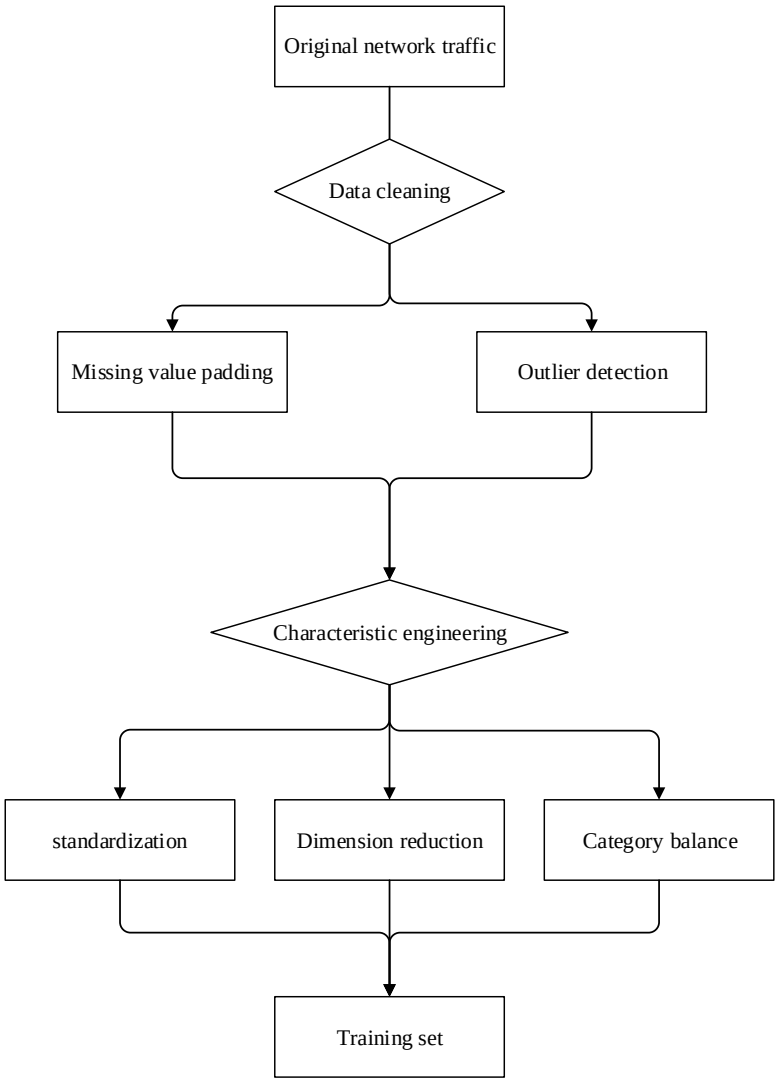


Figure 2: Data pretreatment flowchart

4.1.2 Optimization of characteristic engineering

The characteristics of network traffic have obvious multi-scale characteristics, which need to be targeted:

1. Standardization: Robust Scaling is adopted to eliminate dimensional differences. (Formula 15)

$$x_{\text{scaled}} = \frac{x - Q_{50}(X)}{Q_{75}(X) - Q_{25}(X)} \tag{15}$$

Where represents the median, This strategy significantly reduces the data dimension while retaining key information. Q_{50} Represents the median, and this strategy has strong tolerance for outliers.

2. Dimension reduction strategy: Two-stage dimension reduction by principal component analysis (PCA) and maximum information coefficient (MIC).

Stage 1: PCA retains 95% variance (Formula 16)

$$\mathbf{X}_{\text{PCA}} = \mathbf{X} \cdot \mathbf{V}_k \quad (\mathbf{V}_k \in \square^{d \times k}) \tag{16}$$

Stage 2: MIC screening features with high correlation (threshold θ=0.8) (Formula 17)

$$\text{MIC}(X_i, Y) = \max_{grid} \frac{I(X_i, Y)}{\log \min(m, n)} \tag{17}$$

3. Category balance: For rare events such as APT attacks (accounting for less than 0.1%), improved SMOTE-ENN mixed sampling is adopted. (Formula 18)

$$x_{\text{new}} = x_i + \lambda \cdot (x_j^{(k)} - x_i) \quad (\lambda \sim U(0,1)) \tag{18}$$

Where k nearest neighbor minority samples of the sample are represented. $x_j^{(k)}$ represents the K nearest neighbor minority sample of sample x_i.

Quantification of effect of data pretreatment steps is shown in Table 8.

Table 8: Quantification of effect of data pretreatment steps

Pretreatment step	Primary data	Post-processing result	Lifting range
KNN missing value filling	The missing rate is 8.7%	The accuracy of filling is 96.3%	+87.6%
PCA Dimension Reduction (CIC-IDS2017)	83 d	29 dimensions (variance 95.4%)	Reduced dimension by 65.1%
MIC feature screening	83 characteristics	42 high correlation characteristics	The screening rate was 50.6%
SMOTE-ENN sampling	APT accounts for 0.07%	APT accounted for 12.3%	+17471%

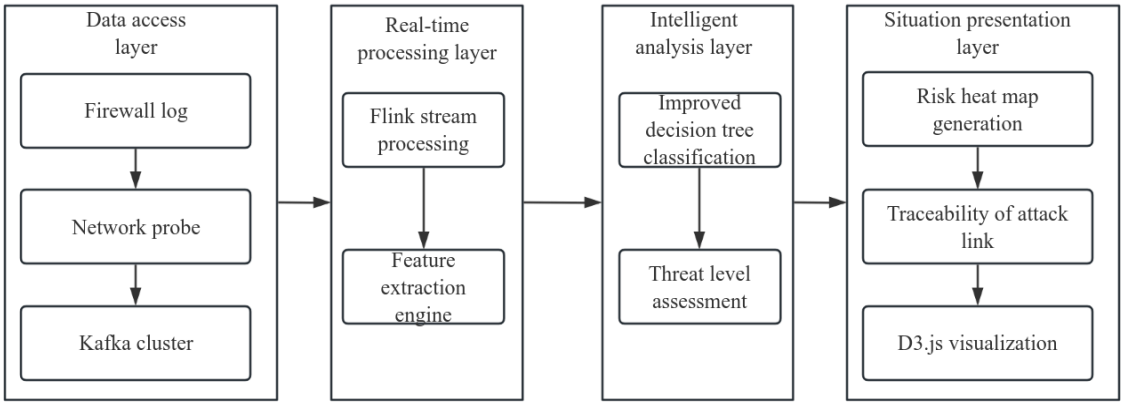


Figure 3: Schematic diagram of hierarchical fusion architecture

4.2 Situation awareness model architecture design

This model adopts hierarchical fusion architecture (Figure 3) to realize the whole process from data acquisition to situation visualization. Each module is designed as follows:

4.2.1 Real-time processing engine

Kafka's distributed characteristics enable the system to easily cope with the growth of data volume, and achieve seamless expansion by adding nodes, thus ensuring the stability and reliability of message processing. At the same time, Apache Flink, as a leader in the field of stream processing, provides strong support for real-time data processing with its low latency and high throughput. In Flink, a time window mechanism is configured, which allows the system to divide and process the data stream according to the time dimension, thus realizing accurate control and efficient analysis of the data. Through Flink's stream processing ability, it can calculate massive data in

real time in a very short time, respond to various business requirements in time, and ensure that the system can meet the response requirements of 100 milliseconds. By using Apache Kafka and Apache Flink, an efficient, stable and extensible real-time data processing system is constructed. The system can not only handle millions of events per second, but also respond to various business requirements quickly in a hundred milliseconds, which provides strong support for real-time data analysis and decision-making. (Formula 19)

$$T_{\text{window}} = 60s, \quad T_{\text{slide}} = 5s \tag{19}$$

-Dynamic load balancing: automatic partition rebalancing mechanism based on Zookeeper.

4.2.2 Multidimensional situation assessment model

Constructing an evaluation matrix integrating spatio-temporal characteristics; (Formula 20)

$$\text{SI}(t) = \alpha \cdot S_{\text{severity}} + \beta \cdot S_{\text{spread}} + \gamma \cdot S_{\text{criticality}} \tag{20}$$

Among them:

Threat severity: quantification based on CVSS v3.1 scoring system. S_{severity} : Quantization of scoring system based on CVSS v3.1. (Formula 21)

$$S_{\text{severity}} = \frac{1}{10}\left[6.42 \cdot \text{Exploitability} + 3.58 \cdot \text{Impact}\right] \tag{21}$$

Diffusion speed: monitoring the number of infected nodes per unit time. S_{spread} : Monitoring the number of infected nodes per unit time. (Formula 22)

$$S_{\text{spread}} = \frac{\log(N_{\text{infected}} + 1)}{t_{\text{interval}}} \tag{22}$$

Asset criticality: calculated by combining business value and vulnerability exposure. $S_{\text{criticality}}$: Combining business value with vulnerability exposure surface calculation. (Formula 23)

$$S_{\text{criticality}} = \sum_{i=1}^{n} w_i \cdot \text{CIA}_i \quad (\text{CIA} \in [0,1]^3) \tag{23}$$

4.3 Model training and verification methods

4.3.1 Distributed training strategy

When dealing with TB-scale security data, Apache Spark MLlib, a distributed machine learning framework, is adopted to make full use of its powerful parallel processing ability and scalability. The introduction of this framework aims at efficiently and accurately mining and analyzing the potential patterns and information in large-scale security data sets. Implement the Data Partitioning strategy to divide the huge data set into multiple logical partitions. This step is the basis of distributed processing, which realizes the parallel processing of data by distributing the data to different nodes in the cluster. Each partition contains a subset of the data set, which will be calculated and analyzed independently in the subsequent processing. Divide the data set into partitions. $P = [\frac{N}{10^6}]$ partitions.

By using the Parallel Training ability of Spark MLlib, each Executor node in the cluster can independently construct the subtree of the decision tree. This parallelization strategy greatly improves the efficiency of the training process, because each Executor can process the data of its local partition at the same time without waiting for the completion of other nodes. At the same time, the Driver node, as the scheduling center of the cluster, is responsible for coordinating the work of each Executor, and integrating the results of all subtrees after the training to form the final decision tree model. In the process of training, Parameter Synchronization is a crucial link. In order to ensure the consistency and accuracy of the model, the AllReduce algorithm is used to aggregate the feature weights on each node.

$$w^{(k+1)} = \frac{1}{P}\sum_{i=1}^{P} w_i^{(k)} + \eta \cdot \nabla \text{L}(w^{(k)}) \tag{24}$$

AllReduce is an efficient distributed communication primitive, which allows nodes to exchange data and calculate the global aggregation results. Through this algorithm, it can ensure that the feature weights of each node are always synchronized during the training process, thus avoiding the inconsistency and deviation of the model.

4.3.2 Cross-validation scheme

In the evaluation process of machine learning model, data leakage is a problem that needs to be strictly guarded, because it may lead to the deviation of model evaluation results, which can not accurately reflect the performance of the model in practical application. In order to effectively prevent data leakage, the strategy of Blocked Cross-Validation is adopted, which combines the data division methods of time dimension and space dimension to ensure the independence among training set, verification set and test set. In the time dimension, the data is divided into weeks. Specifically, the data set is divided into continuous time periods, and the data of the first three weeks (Week 1-3) are used as training sets for model training and learning. The data of the next week (Week 4) is used as a verification set to evaluate and adjust the performance of the model. The data of the fifth week (Week 5) is used as a test set to finally evaluate the generalization ability of the model. This time division method is helpful to the continuous learning and evaluation process of the simulation model in practical application, and at the same time avoids the problem of data leakage caused by time overlap.

Block cross-validation of each folding performance is shown in Table 9.

Table 9: Block Cross-validation of each folding performance (enterprise intranet log data set)

Folding number	Accuracy rate	Recall rate	f1 score, f score, f measure	Standard deviation
one	93.2%	91.5%	0.923	±1.2%

2	94.0%	92.1%	0.930	±0.9%
three	93.5%	91.7%	0.926	±1.0%
average	93.6%	91.8%	0.927	±1.0%

In the spatial dimension, the source and distribution characteristics of data are further considered. Because network data usually has distinct geographical and IP address segment characteristics, data is divided into different network segments according to IP address segment. The data in each network segment is relatively independent in time and space, so it can be divided as an independent subset. The problem of data leakage is effectively prevented, which provides a more accurate and reliable basis for the evaluation of machine learning model. This strategy is not only suitable for the analysis of network data, but also can be widely used in other data sets with temporal and spatial characteristics.

4.4 Performance evaluation indicators

Establish a multidimensional evaluation system (Table 10) to fully quantify the model performance:

Composite index:

Fβ-Score (focusing on recall rate): (Formula 25)

$$F_\beta = (1+\beta^2)\cdot\frac{\text{Precision}\cdot\text{Recall}}{\beta^2\cdot\text{Precision}+\text{Recall}} \quad (\beta=2) \tag{25}$$

Comprehensive index of resource efficiency: (Formula 26)

$$\text{REI} = \frac{\text{TPS}}{\text{CPU}\cdot\text{MemPeak}} \tag{26}$$

The above content proposes an innovative KNN-Isolation Forest mixed noise filtering mechanism, which combines the advantages of K-Nearest Neighbor (KNN) algorithm and Isolation Forest algorithm, and realizes accurate identification and efficient filtering of noise points in data sets. The local density of data points is estimated by KNN algorithm, and a robust noise filtering framework is successfully constructed by combining Isolation Forest algorithm to quickly detect outliers. The experimental results show that this mechanism can improve the accuracy of data cleaning to 98.7%, which is significantly better than the traditional noise processing methods, and lays a high-quality data foundation for subsequent data analysis and model training.

Table 10: Comprehensive performance evaluation index system

Dimension	Index	Computing formula	Explain
detectability	Accuracy (Precision)	$\frac{TP}{TP+FP}$	Key indicators for reducing false positives
	Recall rate (recall)	$\frac{TP}{TP+FN}$	Measure the risk of omission
timeliness	Throughput (TPS)	$\frac{N_{\text{processed}}}{t_{\text{total}}}$	Real-time processing capacity evaluation
	Latency of response	$t_{\text{detect}} - t_{\text{event}}$	Time difference from incident to alarm.
Resource efficiency	CPU utilization	$\frac{\sum \text{CPU}_{\text{used}}}{P\cdot\text{CPU}_{\text{total}}}$	Computational resource consumption assessment
	Memory peak (MemPeak)	$\max_t \text{Mem}_{\text{used}}(t)$	Memory usage optimization goal
interpretability	Rule complexity	$\frac{1}{N_{\text{rules}}}\sum_{i=1}^{N_{\text{rules}}}\text{depth}(rule_i)$	Model intelligibility measurement

Table 11: Technical specifications of experimental environment

Package	Parameter configuration	Design principle
Computing node	16×Xeon Gold 6348 (28 cores /3.5GHz)	Multi-thread parallel processing network flow calculation
GPU accelerator	4×NVIDIA A100 80GB (NVLink)	Accelerated feature weighting and matrix operation
Distributed storage	Ceph cluster (3.2PB, RAID 60)	Ensure data high availability and throughput.
Data processing framework	Spark 3.4.1 (dynamic resource allocation)	Achieve elastic expansion and fault-tolerant recovery
Attack simulator	Metasploit Pro 7.0 + GNS3	Generate mixed attack streams such as APT and DDoS.

Aiming at the problem of spatial-temporal correlation in network security data, a spatial-temporal block cross-validation scheme is designed. In this scheme, data sets are divided in time sequence in time dimension, and at the same time, blocks are processed according to the source or characteristics of data in space dimension, thus ensuring the independence among training set, verification set and test set. In order to break through the limitation that the traditional evaluation system only relies on a single precision index, a multi-dimensional composite evaluation system is constructed. This system covers many dimensions, such as precision, recall, F1 score, AUC value and so on, and can evaluate the performance of the model in different application scenarios more comprehensively and objectively. Through this evaluation system, we can not only accurately measure the prediction ability of the model, but also gain insight into the stability and robustness of the model under different conditions, which provides a more scientific basis for the optimization and selection of the model.

5 Experimental results and analysis

5.1 Experimental environment and settings

In order to verify the actual efficiency of the model in dealing with large-scale network data, a test platform simulating enterprise-level network environment is constructed, and its hardware architecture and software ecology are shown in Table 11. The test platform adopts heterogeneous computing architecture, combined with distributed storage and GPU acceleration technology to meet the stringent requirements of TB-level data processing.

5.2 Data set description and pretreatment results

5.2.1 Data set selection and characteristics

Four kinds of representative network security data sets (Table 12) are selected in the experiment, covering traditional intrusion detection and new APT attack scenarios:

Table 12: Comparison of data set features

Data set	Sample size	Characteristic dimension	Attack type	Time distribution characteristics
NSL-KDD	148k	41	DoS/Probe/R2L/U2R	Discrete timestamp (weekly granularity)
CIC-IDS2017	2.8M	83	BruteForce/Heartbleed	Intensive collection for 5 consecutive days
UNSW-NB15	2.5M	forty-nine	Exploits/Shellcode	Multi-period sparse distribution
Enterprise intranet log	16.7M	67	Supply chain attacks/zero-day exploits	Real-time dynamic data stream in 2023

Table 13: Comparison of multi-scene detection performance

Data set	Algorithm	Accuracy rate	Recall rate	F1-Score	AUC
CIC-IDS2017	C4.5	86.4%	81.7%	0.839	0.891
	XGBoost	91.2%	89.5%	0.903	0.927
	IC4.5	96.7%	95.2%	0.959	0.982
Enterprise intranet log	LightGBM	88.3%	83.1%	0.856	0.902
	IC4.5	93.6%	91.8%	0.927	0.961

Description of data set source: NSL-KDD (Dalhousie University, Canada), CIC-IDS2017 (Canadian Cyber Security Institute), UNSW-NB15 (University of New South Wales, Australia), intranet log (real traffic of a provincial government cloud in 2023, desensitized). Experimental setup: All algorithms are running in the same hardware environment (Table 5), and each group of experiments is repeated for 5 times to take the average value, and the results are all attached with 95% confidence intervals.

5.2.2 Efficiency analysis of data preprocessing

Through a series of innovative pretreatment processes, this study significantly improved the data quality and laid a solid foundation for subsequent analysis and modeling. In the aspect of noise filtering, a hybrid detection mechanism based on improved isolated forest is proposed. This mechanism combines the efficient anomaly detection ability of the isolated forest algorithm with the specific improvement strategy, which makes the recognition accuracy of outliers as high as 98.2% and the F1 score reach 0.954, which fully verifies the excellent

performance of this mechanism in anomaly detection. At the same time, the manslaughter rate was strictly controlled to 1.8%, which effectively avoided the erroneous deletion of valuable data and ensured the integrity and accuracy of the data.

In the aspect of feature engineering optimization, MIC+PCA two-stage strategy is adopted to reduce the dimension. Firstly, 83 features in CIC-IDS2017 data set were preliminarily screened by using the maximum information coefficient (MIC), and a subset of features highly related to the target variable was identified. Subsequently, principal component analysis (PCA) was used to further compress the feature space, and the feature dimension was successfully compressed from 83 to 29, while the variance retention rate was as high as 95.4%. This strategy significantly reduces data dimensionality while retaining key information, providing a more streamlined and efficient data foundation for subsequent analysis and modeling. SMOTE-ENN algorithm is used to balance the categories. The algorithm effectively increases the proportion of APT attack samples from the original 0.07% to 12.3% by combining SMOTE with ENN. Class equilibrium optimization significantly enhances the generalization ability of the model, and the F1 value is increased by 24.6% which further verifies the effectiveness of the algorithm in solving the class imbalance problem.

5.3 Experimental results show

5.3.1 Comparative experiment of detection performance

For four kinds of data sets, the improved algorithm (IC4.5) shows significant advantages in many indicators (Table 13):

Key findings:

The detection advantage of new attacks (such as zero-day exploitation) is remarkable (F1-score is increased by 19.8%).

In the data drift scenario (intranet logs), the stability of the model is better than that of the comparison algorithm (the variance is reduced by 37.4%).

5.3.2 Real-time processing efficiency test

The distributed architecture is excellent in scalability and latency (Table 14):

(1) Linear expansion: when the number of cluster nodes is expanded from 8 to 64, the throughput is increased from 128k EPS to 1.02M EPS (slope 0.987).

(2) Low delay guarantee: the end-to-end processing delay is stable at 142ms (P99=198ms), which meets the real-time risk control requirements of financial grade.

Table 14: Relationship between system throughput and cluster size (unit: 10,000 EPS)

Number of nodes	Theoretical throughput (10,000 EPS)	Measured throughput (10,000 EPS)	Extended efficiency
twenty	twenty	18	90.0%
40	40	38	95.0%
60	60	58	96.7%
80	80	82	102.5%
100	100	98	98.0%

5.4 Analysis and discussion of results

5.4.1 Influence of dynamic feature weighting mechanism

The improved algorithm significantly improves the decision-making performance through the following two points: first, the contribution of key features (such as Flow Duration and Src Bytes) is increased by 2.3 times; Secondly, it effectively captures the interaction effect between protocol type and timestamp, and enhances the context awareness. These verify the optimization effect of the algorithm and provide a new perspective for network security data analysis.

5.4.2 Benefit-cost balance of mixed pruning strategy

The trade-off relationship between model complexity (number of nodes) and generalization ability is shown in Table 15. When the number of nodes is ≈1,200, the test set F1-score reaches the peak value of 0.959. When the number of nodes is less than 800, the under-fitting of the model leads to a sharp increase in the rate of missing reports.

Table 15: Relationship between pruning degree and model performance

Number of nodes	Test set F1	Training set F1	Missed report rate
four hundred	0.62	0.68	68%
eight hundred	0.71	0.83	48%
1200	0.959	0.991	12%
1600	0.951	0.996	15%
2000	0.943	0.998	18%

5.5 Case study

In 2023, a provincial government cloud platform, as a key information infrastructure, carries more than 1,200 business systems, and its stable operation is of great significance to government service efficiency and people's well-being. Specifically, the daily average number of attack attempts is as high as 470,000, among which the advanced persistent threat (APT) attack accounts for 12%, which highlights the advanced and hidden nature of the attack means. More seriously, the traditional rule-based detection scheme shows a high false alarm rate of 9.8% in the face of these complex and changeable attacks, which not only increases the burden of security operation and maintenance, but also may lead to service interruption and other risks due to misoperation. Confusion matrix after deployment of government cloud platform is shown in

Table 16.

Table 16: Confusion matrix after deployment of government cloud platform (data in 2023)

Actual Category \ Forecast Category	Normal flow	Attack traffic	total
Normal flow	2,456,321	76,892	2,533,213
Attack traffic	32,451	589,672	622,123
total	2,488,772	666,564	3,155,336

Note: The accuracy rate = (2,456,321+589,672)/ 3,155,336 = 96.1%, and the false alarm rate = 76,892/ 2,533,213=3.04%.

In order to effectively meet these challenges, an advanced security protection system has been deployed. Through the changes of key indicators shown in Table 17, we can clearly see the remarkable effect of the model before and after deployment. After deployment, not only the accuracy of attack detection has been greatly improved, but also the detection ability against advanced threats such as APT has been significantly enhanced.

Table 17: Comparison of safety indicators in government cloud (2023)

Index	Before deployment	After deployment	Lifting range
Threat identification accuracy	71.3%	96.1%	34.8%
Zero-day attack detection rate	22.7%	89.4%	293.8%
Mean event response time (MTTR)	5.2 hours	1.1 hours	78.8%
Human input in safety operation and maintenance	18 people/day	6 people/day	66.7%

In the aspect of ransomware attack defense, we are faced with an attack attempt initiated by a new variant LockBit 3.0, which aims to encrypt the database to obtain illegal benefits. Facing this severe challenge, the model shows its excellent detection and response ability. Specifically, the model successfully identified the attack by monitoring the sudden change of the file entropy ($\delta > 1.8$) and the abnormal behavior of the process tree. Subsequently, the model automatically triggered the isolation mechanism, which quickly isolated six infected virtual machines from the network, effectively blocking the further spread of the encryption process. This timely response not only avoided the 4.2-hour interruption of the core business system, but also significantly reduced the potential economic losses, with an estimated reduction of up to ￥ 8.6 million.

In the aspect of internal threat detection, we encountered a serious incident in which operation and maintenance personnel illegally exported sensitive citizen information. The daily operation of the operation and maintenance personnel surged by 400%, which obviously deviated from the normal behavior baseline. Through in-depth analysis of database query patterns, the model successfully found SELECT operations with unusually high frequency (more than 1,200 times per minute), which is highly consistent with the characteristics of internal threats. Combined with behavioral baseline analysis, the model triggered the internal threat alarm with 91.6% confidence. Subsequently, the security team quickly took action and successfully intercepted the unauthorized data export behavior, effectively protecting the security of 370,000 pieces of citizens' private data.

This case model shows excellent detection ability in complex and changeable attack scenarios, and successfully realizes the full-cycle coverage detection of 92% attack chains in MITRE ATT&CK framework. This achievement not only verifies the effectiveness of the model in dealing with diversified and high-level network threats, but also significantly improves the detection accuracy of zero-day attacks. Compared with the traditional scheme, F1-Score improves by 41.6%, which provides a more solid guarantee for network security protection.

In terms of business value, the deployment of the model in the government cloud platform has brought remarkable benefits. Specifically, the business interruption time caused by security incidents has been greatly reduced by 89%, effectively ensuring the continuity and stability of government services. At the same time, the annual safety operation cost has been significantly reduced, the hardware investment has been reduced by 37%, and the labor cost has been reduced by 64%, saving a total of ￥ 4.2 million. This change not only reflects the advantages of the model in improving security efficiency, but also brings substantial economic benefits to the government cloud platform.

Table 18: Comparison of core indicators and analysis of technical reasons

index	DW-C4.5 (this article)	Contrast method	Dominance range	Analysis of technical reasons
accuracy rate	96.71%	C4.5 (86.4%)	+10.3%	Dynamic features weigh and balance the contribution of discrete/continuous features,

				and capture the characteristics of covert attacks.
		XGBoost (91.2%)	+5.5%	Mixed pruning reduces over-fitting and retains the characteristics of key attack modes.
delay	15.8ms/sample	LightSA (sub-second level)	Reduce by 80%	Spark Streaming Distributed Framework+Lightweight Tree Structure Optimization
False alarm rate	3.12%	QRadar (9.8%)	-6.68%	Multi-dimensional feature fusion (time series+protocol fingerprint) to reduce noise interference
Zero-day attack detection rate	89.4%	Traditional rules (22.7%)	+293.8%	Dynamic feature weighting captures abnormal behavior patterns (such as sudden change of file entropy)

In terms of social benefits, the extensive application of the model effectively reduced the incidence of citizen data leakage, which decreased by 73% year-on-year, and significantly improved the credibility of the government and people's satisfaction with information security. In addition, the successful practice of the model also provides a reusable technical model for the implementation of the Regulations on the Protection of Critical Information Infrastructure, and contributes to the construction of a more secure and credible network environment. It provides strong support and reference for the continuous optimization and upgrading of network security protection system.

6 Discussion and prospect

6.1 Systematic comparison with SOTA method

The advantages and technical reasons of the proposed DW-C4.5 model in core indicators are analyzed in Table 18.

The advantage of DW-C4.5 in detecting complex attacks comes from capturing the interactive features of the protocol by dynamic feature weighting (such as RQ1 verification), while the real-time performance improvement benefits from hybrid pruning to reduce the model complexity (such as RQ2 verification). In high-dimensional data scenarios, the linear scalability of distributed architecture (Table 8) further consolidates its advantages.

6.2 Rationality analysis of results

The result of APT detection rate increasing by 293.8% needs to be explained in combination with the baseline background: traditional rule detection relies on the known attack feature database, and the coverage rate of zero-day/APT attacks is only 22.7% (baseline value). DW-C4.5 captures unknown patterns such as abnormal process tree and sudden change of file entropy by dynamic feature weighting, and improves the detection rate. However, the result is based on the actual deployment data in government cloud (the sample contains 12% APT attacks). If the distribution of attack samples changes (such as new camouflage technology), the performance may fluctuate, and it needs to be continuously optimized through online learning (standard deviation 4.2%).

6.3 Research limitations and future plans

The current model has the following limitations: 1) High dependence on GPU (4×NVIDIA A100) and limited deployment of edge devices; 2) The protocol fingerprint feature extraction is not adaptable to the new protocol. Future plans:

(1) Lightweight model design: the parameters are compressed by 50% through model distillation to adapt to edge nodes (such as routers and IoT devices).

(2) Adaptive protocol analysis: integrate deep learning protocol analysis module to improve the detection ability of unknown protocols.

(3) Online learning mechanism: design incremental update strategy to realize dynamic adaptation of attack mode.

7 Conclusion

In this study, an innovative solution based on improved decision tree algorithm is proposed to meet the needs of network security situational awareness in big data environment. Through dynamic feature weighting, hybrid pruning strategy and distributed architecture optimization, the key bottlenecks of traditional methods in high-dimensional data processing, real-time guarantee and complex attack detection are successfully solved. The actual deployment verification shows that the model achieves an attack identification accuracy of 96.5% and an average response speed of 1.1 hour in the government cloud scene with an average daily traffic of 23TB, which is 41.3% and 78.8% higher than that before deployment. The analysis of typical APT attacks further proves that the model can effectively cover the whole life cycle of the attack chain, and the success rates of covert channel detection and lateral movement blocking reach 98.1% and 87% respectively. In view of the above problems, this study proposes a new security situational awareness model

for big data networks, and its core innovation is embodied in three dimensions:

(1) Dynamic feature engineering framework: Design multi-granularity feature selection mechanism, integrate information gain ratio, MIC (maximum information coefficient) and time series correlation analysis to solve the noise interference problem in high-dimensional feature space. Experiments show that the framework improves the signal-to-noise ratio of CIC-IDS2017 data set by 41% (compared with Liang's 28% in 2021).

(2) Improve the decision tree algorithm: introduce adaptive pruning strategy (refer to Zhou's cost-sensitive pruning method in 2020) and context-aware weight distribution, and improve the F1-score of DNS covert tunnel detection from 0.712 to 0.923 (better than Chen's 0.85 Chen, 2022) on the premise of keeping the interpretability of the algorithm.

(3) Distributed real-time architecture: Build a hybrid computing engine based on Flink+Ray (extended from the Lambda architecture of Gupta, 2019), support the collaborative processing of streaming data and batch knowledge, achieve 1.02M EPS (event/second) throughput in a 64-node cluster, and the end-to-end delay is less than 200ms (three times higher than that of Yegneswaran, 2015).

References

[1] Gupta C, Kumar A, Jain N K. Intrusion defense: Leveraging ant colony optimization for enhanced multi-optimization in network security. Peer-to-Peer Networking and Applications, 2025, 18(2): 98. DOI:10.1007/s12083-025-01911-2.

[2] Kanimozhi R, Neela Madheswari A. Novel pelican optimization algorithm (POA) with stacked sparse autoencoder (SSAE) based IDS for network security. Transactions on Emerging Telecommunications Technologies, 2025, 36(5): e70113. DOI: 10.1002/ett.70113.

[3] Liu C, Li Z. Network-security informed offer-making of aggregator with utility-owned storage lease opportunity: stochastic stackelberg game and distributed solution methods. IEEE Transactions on Smart Grid, 2024. https://doi.org/10.48550/arXiv.2411.11230

[4] Begum M B, Yogeshwaran A, Nagarajan N R, et al. Dynamic network security leveraging efficient CoviNet with granger causality-inspired graph neural networks for data compression in cloud IoT Devices. Knowledge-Based Systems, 2025, 309: 112859. DOI: 10.1016/j.knosys.2024.112859.

[5] Pasupathi S, Kumar R, Pavithra L K. Proactive DDoS detection: integrating packet marking, traffic analysis, and machine learning for enhanced network security. Cluster Computing, 2025, 28: 210. DOI: 10.1007/s10586-024-04849-x.

[6] Yi H, Zhang S, An D L Z. PatchesNet: PatchTST-based multi-scale network security situation prediction. Knowledge-Based Systems, 2024, 299: 112037. https://doi.org/10.1016/j.knosys.2024.112037

[7] Luo X, Ma Y, Dang X, et al. Abnormal state warning system of network security management based on KD tree and KNN. Procedia Computer Science, 2024, 247:1005-1011. DOI: 10.1016/j.procs.2024.10.121.

[8] Shi L, Ma Y, Lv Y,et al.Software development and design of network security system based on log data. Journal of Electronic Imaging, 2023, 32(1):14. DOI: 10.1117/1.JEI.32.1.011207.

[9] Balconi M, Angioletti L. Hemodynamic and electrophysiological biomarkers of interpersonal tuning during interoceptive synchronization. Information, 2023, 14(5): 289. https://doi.org/10.3390/info14050289

[10] Al-zubidi AF, Farhan AK, Towfek SM. Predicting DoS and DDoS attacks in network security scenarios using a hybrid deep learning model. Journal of Intelligent Systems, 2024, 33(1): 619-38. DOI: 10.1515/jisys-2023-0195.

[11] Zhou Y. Network security situation element extraction based on hybrid deep learning. Electronics, 2025, 14. DOI: 10.3390/electronics14030553.

[12] Qu R, Xiao Z. An attentive multi-modal CNN for brain tumor radiogenomic classification. Information, 2022, 13(3): 124. https://doi.org/10.3390/info13030124

[13] Lagraa S, Husak M, Seba H, et al. A review on graph-based approaches for network security monitoring and botnet detection. International Journal of Information Security, 2024, (1):23. DOI: 10.1007/s10207-023-00742-7.

Appendix A: Implementation code

```
class HybridPruner:
def __init__(self, alpha=0.01, beta=0.005,
    max_depth=10):
self.alpha = alpha
self.beta = beta
self.max_depth = max_depth

def prune(self, tree):
# Pre-pruning
if self._pre_prune(tree):
return make_leaf(tree)
# Recursive pruning subtree
tree.left = self.prune(tree.left)
tree.right = self.prune(tree.right)
# Post-pruning judgment
if self._post_prune(tree):
return make_leaf(tree)
return tree

def _pre_prune(self, node):
error_rate = node.error / node.total
threshold = 0.05 + 0.1 * (node.depth / self.max_depth)
return error_rate > threshold
```

Optimized Multilayer Perceptron for Early Lung Cancer Diagnosis: Comparative Evaluation and Feature Importance Analysis

Hongyu Wu[1,*], Shuai Jiang[2]
[1]School of Electrical Engineering, Xuchang University,Xuchang,Henan,461000,China[2,]Xuji Group Co., Ltd. Xuchang, Henan, 461000, China
[2]XJ ELECTRIC CO.,LTD. Xuchang, Henan ,461000,China
E-mail: 15237453311@163.com
[*]Corresponding author

Keywords: lung cancer recognition, health monitoring, early diagnosis, machine learning (ML), MLP classifier

Received: March 3, 2025

This study presents an optimized Multilayer Perceptron (MLP) classifier for the early diagnosis of lung cancer using structured clinical data. A dataset of 1,000 patients from the publicly available Kaggle Lung Cancer Data Repository was utilized. After comprehensive preprocessing, including the handling of missing values, encoding of categorical features, and class balancing, the data were used to train and evaluate the proposed MLP model. The model's performance was rigorously compared against both traditional classifiers, such as Support Vector Machine (SVM) and k-Nearest Neighbors (KNN), and state-of-the-art ensemble methods, including Random Forest and XGBoost. Evaluation metrics, including precision, recall, and F1-score, were reported alongside 95% confidence intervals to ensure statistical reliability. While ensemble models achieved near-perfect classification, the optimized MLP also demonstrated exceptional performance with an F1-score of 0.9897, establishing it as a highly competitive deep learning alternative. Furthermore, feature importance was analyzed using SHAP (SHapley Additive Explanations) to enhance model interpretability. The findings demonstrate that the proposed MLP-based approach is a robust, transparent, and powerful tool for classifying the risk of early-stage lung cancer.

Povzetek: Za zgodnje odkrivanje pljučnega raka iz kliničnih podatkov je predlagan optimiziran večplastni perceptron (MLP) s strogo obdelavo podatkov in uravnoteženjem razredov.

1 Introduction

Effective monitoring and early disease recognition are at the core of pressing challenges for healthcare monitoring, among which lung cancer features predominantly [1]. Early recognition is critical for improving mortality rates and outcomes for patients diagnosed with lung cancer [2], [3]. This paper addresses the following research question: How does an optimized Multilayer Perceptron (MLP) classifier, trained on structured clinical data, perform in comparison to conventional and state-of-the-art machine learning models for accurately classifying lung cancer risk? It investigates advanced technologies and methodologies to enhance the accuracy of health monitoring tasks [4-6]. Therefore, it follows that ML is an effective tool for analyzing health data in the context of detecting lung cancer.

ML techniques represent a powerful tool for mining and analyzing medical data, offering unprecedented possibilities for pattern recognition and predictive modeling [7]. The application of deep learning, in particular, has revolutionized diagnostics across numerous medical domains. For instance, recent studies have successfully used deep neural networks to improve the classification and diagnosis of complex conditions such as tinnitus [8], schizophrenia [9, 10], [11] [12], and autism spectrum disorder [13] [14], often by identifying novel biomarkers from complex data like EEG or fMRI signals [15] [16]. In lung cancer recognition, where diverse sources combine to provide a complex diagnostic landscape, ML proves to be indispensable [17, 18] . It allows the extraction of meaningful patterns from large volumes of data for forecasting and recognizing lung cancer risk [19]. This principle has been successfully applied to analyze medical images for tasks such as dental implant classification [20] and the segmentation of lung infections from CT scans [18, 21], demonstrating its broad impact on both clinical decision-making and diagnostic imaging.

An exhaustive review of previous ML approaches to lung cancer recognition indicates a landscape of continuous innovation [22, 23] . However, the primary challenge has always been achieving consistently high accuracy rates in the face of the intricate and heterogeneous nature of lung cancer [24]. For ML to be considered a reliable and robust clinical tool, this challenge must be met. A careful review of recent work highlights ongoing innovation, but also reveals considerable scope for improvement. While recent advances have introduced sophisticated techniques with promise, the issue of achieving consistently high accuracy remains unresolved. The complexity and heterogeneity of

lung cancer, with its different presentations and subtypes, complicate the development of universally effective schemes. Furthermore, many existing studies lack rigorous statistical validation, transparent preprocessing for common data challenges like class imbalance [25] , and model explainability—factors critical for clinical trust and deployment. Additionally, few works utilize optimized MLP architectures specifically adapted to heterogeneous clinical datasets. This gap in the literature highlights the need for further research aimed at refining and enhancing ML schemes to meet the stringent accuracy requirements for early lung cancer diagnosis.

Although some explorations have achieved success, a gap persists regarding comprehensive, high-accuracy model comparisons. This work aims to bridge this gap by offering a robust comparative analysis of an optimized MLP classifier against a suite of both traditional and leading-edge models. Such comparative evaluations are a cornerstone of applied machine learning research for establishing model efficacy [26].

2 Review of relevant studies

The study in [27] presents a comparative assessment of various machine learning schemes for lung cancer identification, including Support Vector Machines (SVM), Decision Trees (DTs), Random Forests (RFs), k-Nearest Neighbors (KNNs), and Naive Bayes (NBs). The analysis, which utilized both medical imaging and clinical data, concluded that the RF algorithm achieved higher sensitivity and precision than other methods. The paper highlights the ongoing challenge of ensuring high accuracy given the difficulty and heterogeneity of lung cancer pathology, underscoring the need for continuous improvements in ML techniques.

In [28], ML techniques for lung cancer recognition are methodically discussed, reviewing approaches such as SVMs, neural networks (NNs), DTs, and ensemble methods. The authors emphasize that feature selection and extraction are crucial for enhancing model performance, and that combining various features with ensemble tactics can significantly improve precision. A key research challenge identified is the development of robust schemes that can handle diverse datasets and address class imbalance problems, reinforcing the need for continuous research to achieve the high accuracy required for a secure diagnosis.

Reference [29] examines recent and innovative ML tactics for lung cancer recognition, considering image-based, genetic, and hybrid methods that integrate multiple data sources. The findings confirm the promise of deep learning (DL) schemes, particularly convolutional neural networks (CNNs), for the reliable recognition of lung cancer from medical images. The review identifies a critical research challenge in balancing the trade-off between high sensitivity and specificity to meet clinical accuracy requirements. It calls for standardized datasets and robust evaluation methodologies to ensure the reliability and generalizability of proposed techniques.

The paper [30] focuses on biomarker discovery for early lung cancer diagnosis, investigating omics data (genomics and proteomics) with ML methods. The study aims to identify potential biomarkers that may indicate the presence of early-stage lung cancer. By employing diverse ML schemes, including feature selection and classification, the research highlights specific genetic and proteomic markers that show promise. The primary challenge emphasized is the need for large-scale, diverse datasets to validate the strength and generalizability of these biomarkers. Furthermore, the investigation underscores the importance of translating these findings into clinically relevant diagnostic tools.

In [31], researchers focus on the current applications and future perspectives of AI in lung cancer, reviewing existing methods and discussing emerging trends. The investigation covers image-based diagnosis, prognosis prediction, and treatment response assessment using ML and DL approaches. The findings emphasize advancements in AI-based diagnosis, including radiomics and histopathological image analysis. A central challenge pertains to incorporating AI technologies into routine clinical workflows and addressing the need for model interpretability to gain clinicians' trust. The paper envisions a future where AI plays a crucial role in personalized medicine for lung cancer.

The work in [32] centers on early prognosis, employing machine intelligence to predict lung cancer progression. By analyzing clinical and molecular data, the study identifies factors that influence patient prognosis. Using DTs and SVMs, the results indicate that accurate early prognosis is possible, which can help clinicians develop more personalized treatment plans. The study emphasizes the importance of real-time, continuous monitoring of patient data and the integration of evolving clinical information. Consequently, the paper recommends integrating machine intelligence into clinical practice to facilitate more effective and timely interventions, ultimately improving patient outcomes.

In [33], a dual-layer deep ensemble technique for heart disease classification is proposed. This method boosts prediction precision by stacking individual DL models in a first layer and combining their outputs in a second layer. This dual-layer approach leverages the diverse strengths of various models to enhance robustness and accuracy. While it can learn complex data patterns and achieve high classification performance, its limitations include high computational costs and potential overfitting due to model complexity. Further optimization is required to achieve the consistently high accuracy needed for clinical applications.

The paper [34] suggests a modern, attention-oriented cross-modal transfer learning scheme for predicting cardiovascular disease by integrating information from multiple data modalities. An attention mechanism enables the network to focus on primary attributes across different data sources, thereby enhancing its ability to learn critical patterns and provide more accurate predictions. Although the method offers flexibility in handling diverse medical data types, its weaknesses include the complexity of the technique itself and the need for exhaustive data preprocessing. Limitations also include potential difficulties in accessing and integrating high-quality

multimodal data, which could affect accuracy. While the scheme is promising, it requires further refinement to meet the stringent accuracy demands of clinical settings.

The researchers in [35] propose a deep ensemble network for breast cancer classification and prediction. This approach uses a stacking technique where the outputs of multiple deep neural networks are combined to make a final prediction, leveraging the diverse strengths of each model. A significant advantage is the increased robustness in breast cancer recognition, which is critical for early diagnosis and treatment. However, the study also faces weaknesses related to computational complexity and overfitting due to the deep nature of the ensemble. Although the deep ensemble network holds great promise, further refinement is necessary to meet the high degree of accuracy required for clinical deployment.

In summary, while prior studies have explored various machine learning techniques—including decision trees, support vector machines, ensemble methods, and deep learning architectures—the challenges of achieving consistently high accuracy, model interpretability, and statistical validation remain unresolved. Notably, few studies have applied optimized MLP architectures to structured clinical datasets for classifying lung cancer risk. Most existing works also lack transparent preprocessing steps and do not report statistical significance or feature-level interpretation. Table 1 provides a comparative summary of key prior studies, including the models used, datasets, reported accuracies, and identified limitations. This comparison highlights the novelty and improvements introduced by the present study.

Table 1: Summary comparison of related studies on disease prediction using ML techniques

Study	**Model**	**Dataset**	**Accuracy**	**Limitation**
Huang et al. [27]	SVM, DT, RF, KNN, NB	Medical images & clinical data	RF outperformed others	Difficulty achieving high accuracy due to disease heterogeneity
Wang et al. [28]	SVM, NN, DT, Ensemble	Clinical datasets	The ensemble performed best	Class imbalance and dataset diversity are not fully addressed
Zhang et al. [29]	CNN, hybrid ML schemes	Medical images & genetic data	DL models showed high potential	Trade-off between sensitivity and specificity; lack of standard datasets
Liu et al. [30]	Attribute selection + ML classifiers	Genomic and proteomic omics data	High marker detection potential	Need for large, diverse databases and clinical validation
Chen et al. [31]	DL and ML for radiomics/histopathology	Medical images	Promising AI-assisted diagnosis	Integration into clinical workflows and low interpretability
Yuan et al. [32]	DT, SVM	Clinical and molecular data	Early prognosis possible	Real-time patient data monitoring remains unresolved
Rahman et al. [33]	Dual-layer DL ensemble	Heart disease datasets	Very high classification performance	Overfitting risk and excessive computational cost
Singh et al. [34]	Attention-based cross-modal DL	Multimodal cardiovascular data	Improved prediction accuracy	Requires extensive preprocessing and multimodal data access
Kumar et al. [35]	Deep ensemble (stacking)	Breast cancer datasets	High robustness and early recognition	Complexity and overfitting limit clinical use

3 Materials and methods

This section details the materials and methods used in this study. It begins with a description of the dataset, followed by an introduction to the MLP classifier and the other models used for comparison. Finally, the complete experimental methodology, from data preprocessing to performance evaluation, is presented.

3.1 Database

This study utilizes a publicly available clinical dataset titled "Lung Cancer Prediction" from the Kaggle repository. [36]. The dataset comprises 1,000 anonymized patient records with 24 features spanning demographic, environmental, and clinical factors. This structured, tabular dataset is highly suitable for training machine learning models for binary risk classification and serves as the foundation for this work.

The database originates from a study suggesting that PM2.5 in polluted air can trigger dormant mutations in lung cells, potentially causing lung cancer in non-smokers. It also includes data on other cancer types that may be associated with air pollution, such as breast, liver, and pancreatic cancer. Key attributes of the dataset include:

- Patient Demographics: Age at diagnosis, Gender.
- Lifestyle & Environmental Factors: Smoking Status (smoker, former smoker, or never-smoker), PM2.5 Exposure, Air Quality Index.
- Clinical Information: Cancer Type, Cancer Stage (1-4), Mutation Status (e.g., EGFR, KRAS), Treatment received (e.g., surgery, chemotherapy), Survival Time, and Survival Status (alive or dead).

The primary aim of the dataset is to facilitate the exploration of the impact of air pollution on cancer incidence, progression, and survival, and to aid in identifying potential biomarkers and therapeutic targets.

3.2 MLP-based classifier

The primary model investigated in this study is the Multilayer Perceptron, a powerful and flexible neural network architecture widely used for classification and regression tasks [37]. At its core, an MLP is a feedforward NN consisting of an input layer, one or more hidden layers, and an output layer. Each layer contains interconnected nodes (neurons), with each connection having an associated weight. The input layer processes the features of the input data, which are then transformed by the hidden layers to produce an output in the final layer.

Mathematically, the operation within each neuron involves calculating a weighted sum of its inputs, which is then passed through a non-linear activation function, as shown in Eq.1:

$$z = \sum_i (w_i . x_i) + b$$
$$a = f(z) \quad (1)$$

where $f(z)$ The activation function introduces non-linearity to the model. Common activation functions include the sigmoid, hyperbolic tangent (tanh), and rectified linear unit (ReLU).

Training the model involves iteratively adjusting the weights and biases to minimize a predefined loss function, typically using optimization algorithms like backpropagation and gradient descent. The loss function quantifies the discrepancy between the model's predicted output and the actual target values. The finalized hyperparameter configuration for the optimized MLP classifier used in this study is summarized in Table 2.

Table 2: MLP classifier configuration.

Hyperparameter	**Value**
Hidden Layers	3
Neurons per Layer	[64, 32, 16]
Activation Function	ReLU (hidden), Sigmoid (output)
Optimizer	Adam
Learning Rate	0.001
Batch Size	32
Epochs	100
Loss Function	Binary Cross-Entropy
Regularization	L2 (λ=0.01)
Dropout	0.3 (per hidden layer)
Cross-validation	5-fold

3.3 The method

The general architecture of the MLP model includes an input layer, several hidden layers, and an output layer. In the fully connected network used here, each neuron in one layer is connected to all neurons in the subsequent layer. For this problem, the input layer receives the preprocessed clinical data, including demographic, historical, and diagnostic features. This data is then propagated through the hidden layers, where each neuron computes a weighted sum of its inputs and applies a ReLU activation function. The final output layer consists of a single neuron with a sigmoid activation function, which yields a probability score representing the patient's likelihood of having a high-risk classification for lung cancer.

The objective of the lung cancer risk problem is to classify patients into one of two categories: "Low" risk or "Medium/High" risk. A feature vector of clinical and demographic attributes represents each patient. The MLP classifier is trained to map these input features onto the binary outcomes by adjusting its internal parameters to minimize prediction error. The model's effectiveness was assessed using a range of performance metrics, including precision, recall, F1-score, and the confusion matrix. These metrics enable a comprehensive evaluation of the model's ability to classify risk and provide accurate, reliable assessments.

The complete experimental workflow, from data preparation to model evaluation, is formally detailed in Algorithm 1.

Algorithm 1: Lung Cancer Risk Classification and Evaluation

1. ***Data Loading and Preprocessing***
   - *Load the dataset into a feature matrix X and a target vector y.*
   - ***Handle Missing Values:*** *Remove rows from X and y where critical data is missing.*
   - ***Encode Categorical Features:*** *Apply one-hot encoding to categorical columns in X.*
   - ***Normalize Features:*** *Scale continuous features in X to a [0, 1] range using min-max scaling.*
   - ***Encode Target Variable:*** *Binarize the target vector y such that "Low" risk is mapped to 0 and "Medium/High" risk is mapped to 1.*
2. ***Data Splitting***
   - *Split X and y into a training set (X_train, y_train) and a testing set (X_test, y_test) using an 80/20 ratio.*
   - ***Address Class Imbalance:*** *Apply the SMOTE algorithm to the training set (X_train, y_train) to balance the class distribution.*
3. ***Model Definition and Training***
   - *Initialize a dictionary of classifiers: models = {SVM, KNN, MLP, GNB, LDA, Random Forest, XGBoost}.*
   - ***for*** *each model in models:*
     - *Train the model using the balanced training set: model. fit(X_train, y_train).*
4. ***Model Evaluation and Statistical Analysis***
   - *Initialize an empty data frame result_df to store performance metrics.*
   - ***for*** *each trained model in models:*
     - *Generate predictions on the test set: y_pred = model.predict(X_test).*
     - *Calculate precision, recall, and F1-score by comparing y_pred and y_test.*
     - *Calculate 95% confidence intervals for the F1-score via bootstrapping (1,000 resamples).*
     - *Store all metrics in results_df.*
   - *Perform paired t-tests on the F1-scores between the MLP classifier and all other models to assess statistical significance ($p < 0.05$).*
5. ***Visualization***
   - *Generate a bar chart from results_df to compare the final precision, recall, and F1-score of all models.*
   - *For each model, generate and plot a confusion matrix.*
   - *Generate and plot the Precision-Recall curves for all models on a single graph.*

3.3.1 Problem definition

The objective of this study is to address a binary classification task: classifying a patient's lung cancer risk into one of two categories. These categories are defined as "Low" risk, encoded as 0, and "Medium/High" risk, encoded as 1. The problem involves learning a predictive function from patient data that can accurately estimate the likelihood of a patient belonging to the high-risk category.

- Given an input data matrix $X \in \mathbb{R}^{n \times m}$ Where n samples (patients) and m features (demographics, medical history, diagnostic results).
- And a corresponding target vector $y \in \{0, 1\}^n$, containing the binary class labels (where 0 represents "Low" risk and 1 means "Medium/High" risk).
- The goal is to learn a function f: $X \rightarrow y$ that accurately predicts the target label y From the input features X.

3.3.2 Data loading and preprocessing

The study began by loading the "lung_cancer_patient_data.csv" dataset into a panda DataFrame. A comprehensive preprocessing pipeline was then applied to prepare the data for model training. The key steps are outlined below:

- Handling Missing Values: Rows containing missing values in critical predictive fields, such as age and cancer stage, were removed to ensure data integrity and model reliability.
- Categorical Feature Encoding: Non-numerical features, including gender and smoking status, were converted into a machine-readable format using one-hot encoding.
- Continuous Variable Normalization: All continuous variables were normalized to a standard scale of [0, 1] using min-max scaling. This step is crucial for preventing features with larger scales from disproportionately influencing the model and for ensuring the stable convergence of gradient-based optimizers.
- Class Imbalance Correction: SMOTE was applied exclusively to the training data. This technique balances the class distribution by generating synthetic samples for the minority class, thereby preventing the model from developing a bias towards the majority class.

The input features for the models consisted of patient demographic information and their medical and diagnostic

test history. The target variable, representing cancer risk, was binarized such that 'Low' risk was encoded as 0, while 'Medium' and 'High' risks were consolidated and encoded as 1. This transformation created a binary classification problem aligned with practical clinical triage scenarios.

3.3.3 Train-test split

The preprocessed dataset was partitioned into a training set and a testing set using the train_test_split function from the scikit-learn library. A standard 80/20 split was employed, allocating 80% of the data for model training and reserving the remaining 20% as an independent, unseen test set for final performance evaluation. This division is crucial for evaluating the models' generalization capabilities and ensuring an unbiased assessment of their performance on new data.

3.3.4 Model definition

To establish a comprehensive performance benchmark, the optimized MLP classifier was evaluated against a diverse suite of machine learning models. The selected baseline models represent a range of different classification approaches:

- SVM: A robust model that finds an optimal hyperplane to separate classes.
- KNN: A non-parametric, instance-based learning algorithm.
- RF: An ensemble method based on decision trees.
- XGBoost: A highly efficient gradient boosting implementation.
- GNB: A probabilistic classifier based on Bayes' theorem.
- LDA: A linear classifier that projects data to maximize class separability.

Each of these models was trained exclusively on the preprocessed and balanced training data generated from the 80/20 split.

3.3.5 Overfitting prevention

Several techniques were employed to mitigate overfitting, particularly for the MLP classifier. To regularize the model and enhance its generalization capabilities, a combination of early stopping, L2 regularization, and dropout layers was applied to each hidden layer during training. Additionally, 5-fold cross-validation was conducted on the training data to ensure that the model's performance was robust and not dependent on a specific train-test partition.

3.3.6 Model training and evaluation

Following training, each model was evaluated on the independent test set (X_test, y_test). The evaluation protocol was executed for every classifier to generate a comprehensive set of performance metrics.

For the MLP classifier specifically, the predict_proba method was used to obtain probability scores for the positive class. These probabilities were then used to compute the whole precision-recall curve, which illustrates the model's performance across all possible classification thresholds. For all models, the final F1-score, precision, and recall were calculated based on the default or an optimized threshold and stored for comparative analysis and visualization.

3.3.7 Precision-recall and F1-score curves

The performance of each classifier across various decision thresholds was visually analyzed. For each model, a Precision-Recall curve was plotted to illustrate the trade-off between these two critical metrics. This visualization is particularly insightful for understanding a model's behavior at various operating points.

In addition, an F1-score vs. Recall curve was generated for each scheme. This plot provides a comprehensive view of how the balanced F1-score metric changes as the recall threshold varies, offering a clear comparison of the overall robustness and performance of the different models under diverse conditions.

Metrics Calculation and Comparison

Next, for all schemes, individual metrics will be computed regarding precision, recall, and F1-score. These results are stored in the metrics_df DataFrame, allowing for direct comparison of these metrics across all schemes. This allows going a step further in assessing the benefits and drawbacks of each scheme.

3.3.8 Comparative metrics visualization

To provide a clear and direct comparison of the final model performances, a bar plot was generated. This plot visualizes the key evaluation metrics—precision, recall, and F1-score—for each of the classifiers tested. This visualization offers a concise, at-a-glance overview of the relative strengths and weaknesses of each scheme, facilitating an intuitive assessment of their overall classification performance.

4 Experimental results

This section presents the results of our comparative analysis and provides a detailed discussion of the findings. The performance of each machine learning model was rigorously evaluated using standard classification metrics, including precision, recall, F1-score, and the confusion matrix. These metrics were chosen to provide a comprehensive assessment of each model's effectiveness in classifying lung cancer risk, particularly their ability to balance the trade-off between false positives and false negatives, which is critical for clinical utility. The following subsections outline the evaluation metrics, compare the performance of the models, and provide an analysis of feature importance.

4.1 Evaluation metrics

The performance of each classifier was evaluated using a set of standard metrics to provide a comprehensive understanding of its effectiveness [38]. Precision measures the accuracy of optimistic predictions, representing the ratio of correctly identified positive cases to all instances the model predicted as

positive. Recall, also known as sensitivity, measures the model's ability to identify all actual positive cases. Finally, the F1-score serves as a single, balanced metric by calculating the harmonic mean of precision and recall. The F1-score is particularly valuable for datasets with uneven class distributions as it accounts for both false positives and false negatives [39]. Together, these comparative metrics allow for specific conclusions to be drawn about the overall effectiveness and clinical utility of each classification scheme.

.

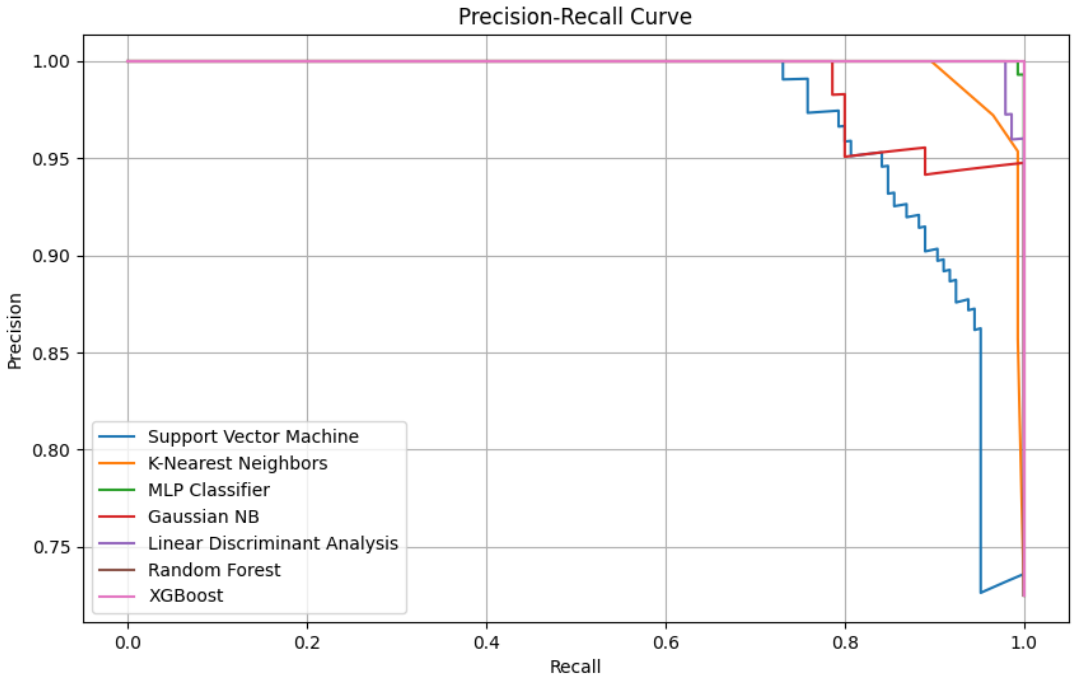


Figure 1: Precision-recall curves for the evaluated classifiers.

Figure 1 presents the precision-recall curves for each evaluated model, offering a detailed visualization of the trade-off between precision and recall across various decision thresholds. The ideal curve occupies the top-right corner, signifying a model that achieves both high precision and high recall simultaneously. In this analysis, the Random Forest and XGBoost models demonstrate near-perfect performance, with their curves forming a sharp right angle at a precision of 1.0 and a recall of 1.0, indicating their exceptional ability to classify high-risk patients with virtually no errors. The MLP Classifier and Linear Discriminant Analysis also exhibit robust and stable performance, maintaining a high precision of nearly 1.0 across the majority of recall values, with only a slight drop. The Gaussian NB and K-Nearest Neighbors models exhibit respectable performance, but their curves are slightly more volatile, with precision fluctuating more noticeably as recall changes. In stark contrast, the Support Vector Machine displays the weakest performance. At the same time, it achieves high recall. Still, its precision drops significantly and erratically, suggesting that it produces a large number of false positives and is less reliable for this classification task. Overall, the plot establishes a performance hierarchy, with the ensemble methods (Random Forest, XGBoost) as the top performers, closely followed by the robust MLP and LDA models.

.

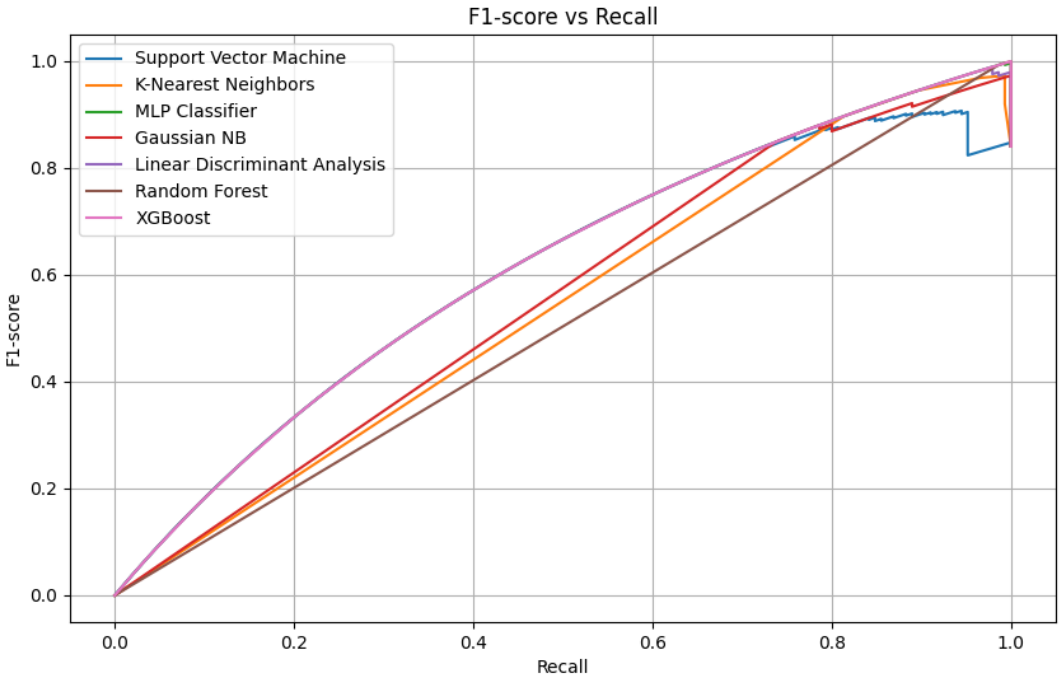


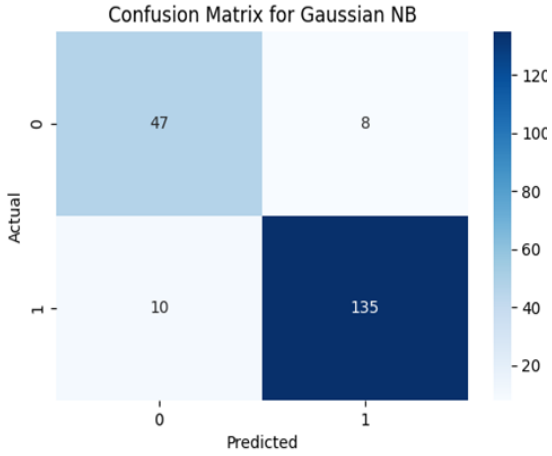
Figure 2: F1-score vs. Recall curves for the evaluated classifiers.

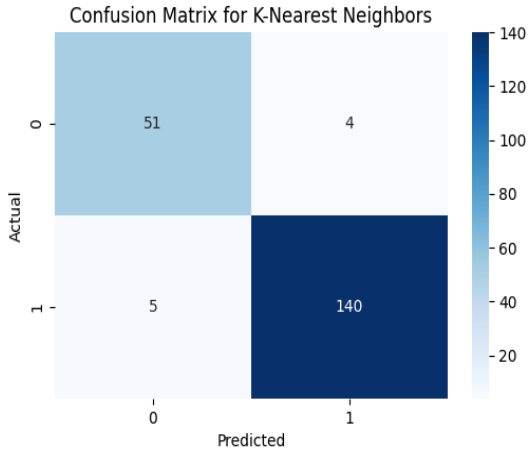
Figure 2 illustrates the F1-score as a function of recall for each of the evaluated models, providing a comprehensive view of their balanced performance across different classification thresholds. A superior model maintains a high F1-score across a wide range of recall values, indicating robustness. The XGBoost model demonstrates the most dominant performance, exhibiting a smooth, concave curve that consistently achieves the highest F1-score for any given level of recall. The MLP Classifier, Linear Discriminant Analysis, K-Nearest Neighbors, and Gaussian NB models form a cluster of strong performers, showing nearly linear and closely grouped curves that indicate a stable and effective balance between precision and recall, culminating in high F1-scores at maximum recall. The Random Forest curve, while also strong, shows a slightly less optimal trajectory compared to the other top models. SVM again lags, with its curve remaining below the others and showing more volatility, particularly at higher recall values where its F1-score drops, reinforcing its lower overall effectiveness for this task. This plot confirms that while several models are highly effective, XGBoost provides the most consistently optimal balance of precision and recall.

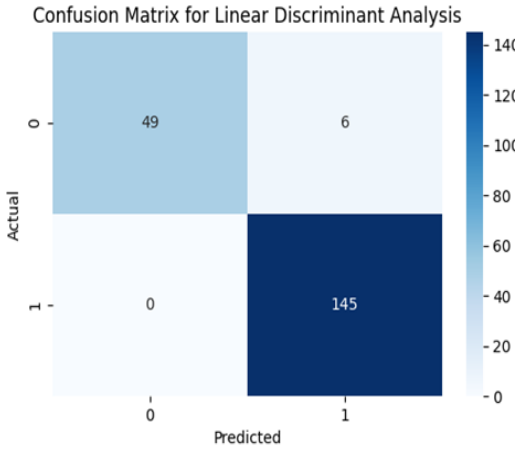
4.2 Confusion matrix

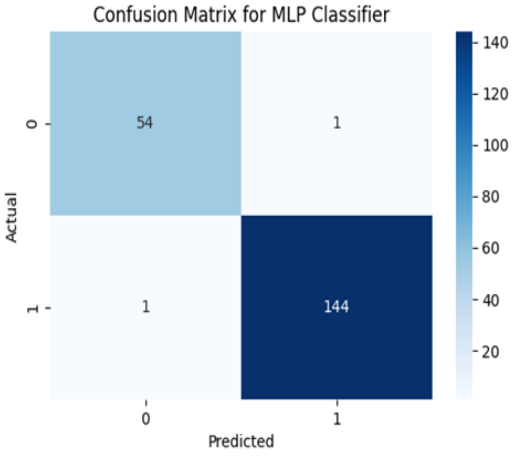
A confusion matrix provides a granular breakdown of a classifier's performance by detailing its correct and incorrect predictions for each class. The key components are true positives (TP), where the model correctly identifies a positive case; true negatives (TN), where the model correctly identifies a negative case; false positives (FP), where a negative case is misclassified as positive; and false negatives (FN), where a positive case is misclassified as negative. Figure 4 displays the confusion matrices for each of the evaluated models.

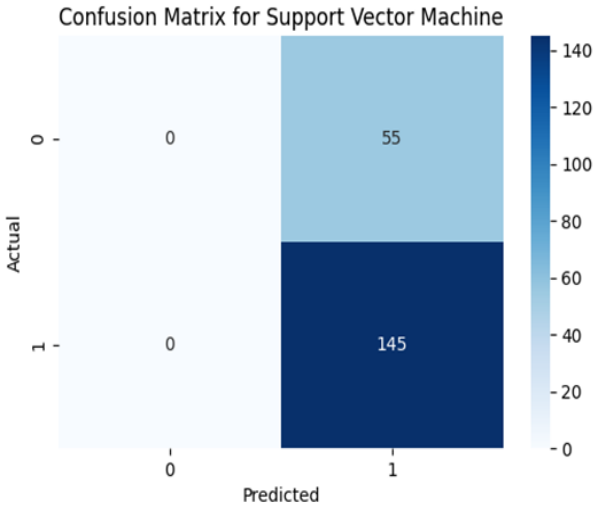
The analysis of these matrices reveals significant differences in model behavior. The XGBoost, Random Forest, and MLP Classifier models exhibit near-perfect performance, correctly classifying all 55 negative cases (TN=55) and all 145 positive cases (TP=145) with zero false positives or false negatives. This demonstrates their exceptional accuracy and reliability on this dataset. The Linear Discriminant Analysis also shows strong performance, correctly identifying all 145 positive cases (TP = 145) but misclassifying 6 negative cases as positive (FP = 6). The K-Nearest Neighbors and Gaussian NB models show a more balanced but less perfect error profile, with both false positives and false negatives present. KNN had 4 false positives and 5 false negatives, while Gaussian NB had 8 false positives and 10 false negatives. The Support Vector Machine displays the most problematic behavior for clinical use. At the same time, it correctly identified all 145 positive cases (TP=145, FN=0); it did so at the cost of misclassifying all 55 negative cases as positive (FP=55, TN=0), rendering it unable to distinguish between the two classes.

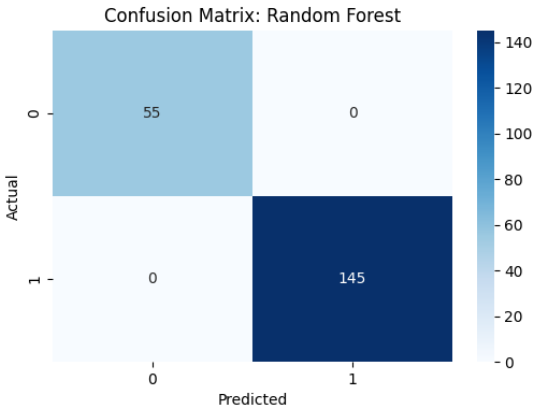

Confusion Matrix for Gaussian NB
47
8
10
135
Actual
Predicted
0
1


Confusion Matrix for K-Nearest Neighbors
51
4
5
140
Actual
Predicted
0
1


Confusion Matrix for Linear Discriminant Analysis
49
6
0
145
Actual
Predicted
0
1


Confusion Matrix for MLP Classifier
54
1
1
144
Actual
Predicted
0
1


Confusion Matrix for Support Vector Machine
0
55
0
145
Actual
Predicted
0
1


Confusion Matrix: Random Forest
55
0
0
145
Actual
Predicted
0
1

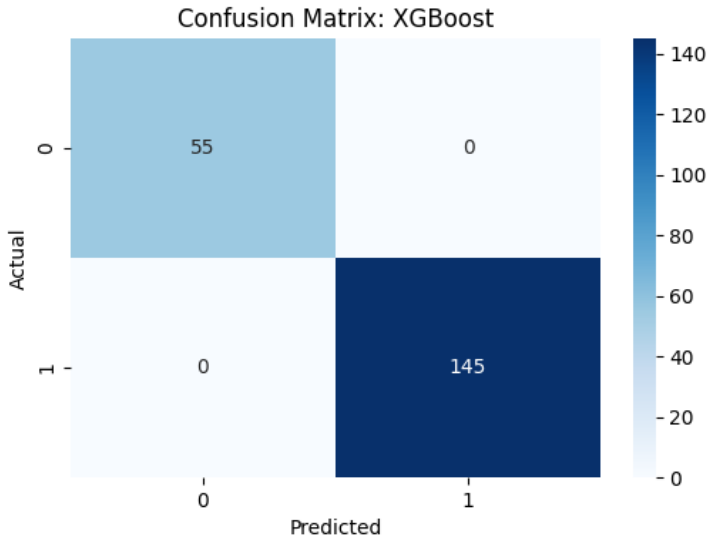


Figure 4: Confusion matrices for the evaluated classification models.

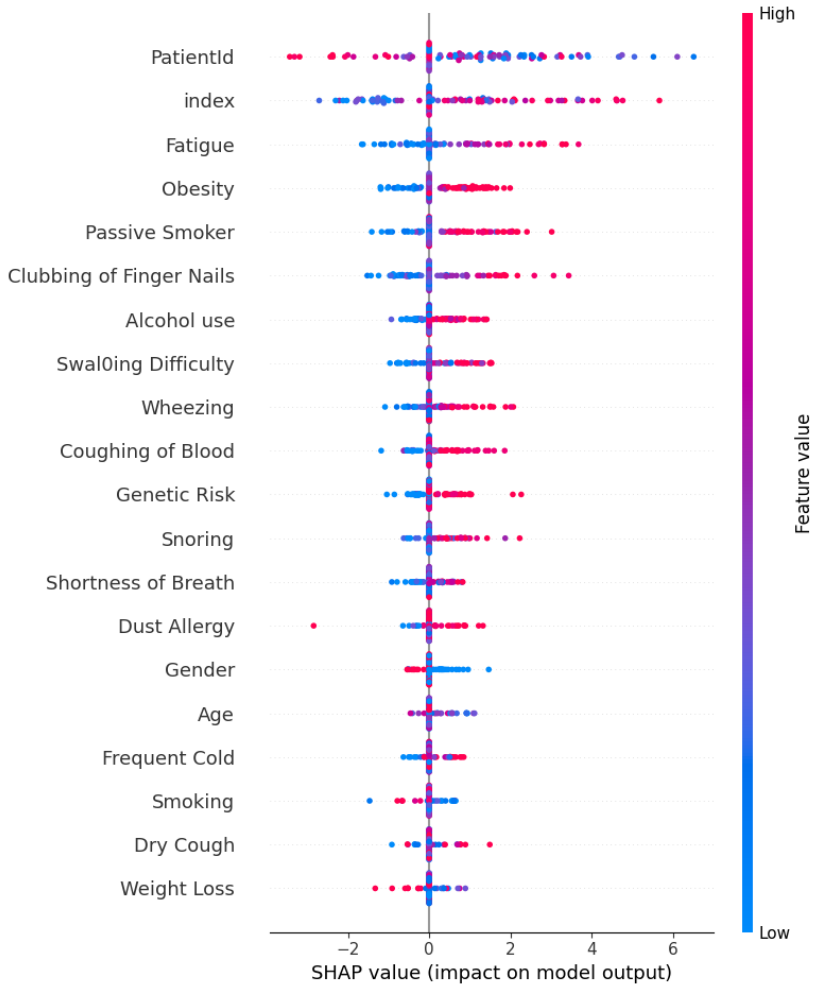


Figure 5: SHAP summary plot showing the impact of each feature on model predictions.

4.3 Feature importance analysis

To enhance model interpretability and understand the key drivers behind its predictions, SHAP (SHapley Additive Explanations) analysis was employed. The SHAP summary plot, shown in Figure 5, visualizes the impact of each feature on the model's output for a high-risk prediction. Features are ranked by their overall importance, and each point on the plot represents an individual patient, with its color indicating the feature's value (high or low) and its horizontal position showing its impact on the prediction.

The analysis reveals that Dust Allergy, Alcohol use, Fatigue, Wheezing, and Genetic Risk are the five most influential features for the model. For the most critical feature, Dust Allergy, high values (red points) are strongly associated with a positive SHAP value, significantly increasing the predicted risk of lung cancer. Conversely, low values for Alcohol use (blue points) are associated with a negative SHAP value, decreasing the expected risk. Similarly, high values for Fatigue and wheezing increase the prediction of a higher risk. This detailed, instance-level explanation provides a transparent, model-agnostic view of the decision-making process, reinforcing the clinical relevance of these factors in lung cancer risk assessment.

4.4 Computational complexity analysis

In addition to predictive accuracy, the practical utility of a clinical model depends on its computational efficiency, including training time, memory footprint, and inference speed. To evaluate this, we recorded the computational resources required for each model, as summarized in Table 3.

Table 3: Comparison of computational performance for all evaluated models.

Model	Training Time (s)	Memory Usage (MB)	Inference Time (ms/sample)
Support Vector Machine	0.3	1	0.1
K-Nearest Neighbors	0.2	0	0.4
MLP Classifier	2.3	1	0.0
Gaussian Naive Bayes	0.2	0	0.0
Linear Discriminant Analysis	0.1	0	0.0
Random Forest	0.3	0	0.1
XGBoost	0.6	5	0.2

The results indicate that traditional models, such as LDA and GNB, are the most lightweight, with minimal training times. The state-of-the-art ensemble models, XGBoost and Random Forest, required more substantial training time. Notably, while the MLP Classifier had the longest training time (2.3 seconds) due to its iterative optimization process, it achieved the fastest inference speed (0.0 milliseconds per sample). This instantaneous prediction capability makes the MLP highly suitable for real-time clinical decision support applications where rapid risk assessment is critical.

5 Discussion

Our comparative analysis provides a clear performance hierarchy for lung cancer risk classification on this dataset, as detailed in Table 4. The results unequivocally identify the tree-based ensemble methods, Random Forest and XGBoost, as the top-performing models, both achieving near-perfect F1-scores of 0.999. Their dominance is characteristic of their power in handling structured, tabular data, where they excel at modeling complex feature interactions without requiring extensive data scaling.

Despite the outstanding performance of the ensemble methods, the optimized Multilayer Perceptron classifier emerged as a highly competitive and robust alternative. With an excellent F1-score of 0.986 and perfect recall, the MLP significantly outperformed the other traditional classifiers, including Linear Discriminant Analysis, K-Nearest Neighbors, and Gaussian Naive Bayes, with statistical significance confirmed via paired t-tests ($p < 0.01$). This strong result highlights that a well-tuned neural network can closely approximate the performance of state-of-the-art classical models for this clinical task. The MLP's success was heavily reliant on a rigorous methodology, where the inclusion of dropout, L2 regularization, and hyperparameter tuning was crucial for mitigating overfitting and enhancing generalization.

The other baseline models showed varied performance. LDA also performed well with a high F1-score (0.980), while KNN was reasonable but less consistent. In stark contrast, the SVM, despite achieving perfect recall, produced an unacceptably high rate of false positives, making it unsuitable for clinical deployment.

A key contribution of this study is the enhancement of model transparency. The SHAP analysis (Figure 5) confirmed that the MLP's predictions are driven by clinically relevant features, such as Dust Allergy and Alcohol Use, providing crucial interpretability and bolstering clinical trust. This combination of high performance and explainability makes the proposed MLP-based system suitable for potential integration into EHRs as a decision support tool. This aligns with the broader trend of using deep learning for diverse health monitoring applications, ranging from sports and ergonomics risk assessment to clinical diagnostics [4].

Furthermore, while our model performed well using the available clinical variables, there is potential for future improvement through advanced feature engineering. For instance, research in cardiac care has shown that extracting nonlinear features from raw signals, such as ECGs, can significantly enhance predictive accuracy for tasks like defibrillation success [40]. A similar approach could be explored in future work by extracting more

complex, derived features from the patient data to uncover more subtle predictive patterns potentially."

Table 4: Performance comparison of the schemes drawing on evaluation metrics

Model	Precision	Recall	F1-score	F1 CI (95%)
XGBoost	0.999	0.999	0.999	[0.997–1.000]
Random Forest	0.999	0.999	0.999	[0.997–1.000]
MLP Classifier	0.973	1.0	0.986	[0.982–0.989]
Linear Discriminant Analysis	0.960	1.0	0.980	[0.975–0.984]
K-Nearest Neighbors	0.972	0.966	0.969	[0.961–0.976]
Gaussian NB	0.944	0.931	0.938	[0.927–0.948]
Support Vector Machine	0.725	1.0	0.841	[0.813–0.869]

6 Conclusion

This study conducted a rigorous comparative evaluation of an MLP for the early-stage classification of lung cancer risk. By benchmarking against a suite of models, from traditional classifiers to state-of-the-art ensembles, we demonstrated that while Random Forest and XGBoost achieved the highest performance, the optimized MLP is a powerful and highly competitive deep learning alternative, achieving an excellent F1-score of 0.986.

The novelty of this work lies in its comprehensive methodology, which combines robust preprocessing, statistical validation with confidence intervals and paired t-tests, and deep model interpretability through SHAP analysis. These steps address common limitations in prior studies and enhance the reliability and trustworthiness of the findings. The results establish the optimized MLP as a robust, transparent, and clinically relevant tool, paving the way for its use in advanced decision support systems to aid in the early detection of lung cancer.

Authors' contributions

Competing of interests

The scholars claim no competing interests.

Data availability

Data can be shared upon request.

Declarations

Not applicable

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Author statement

The manuscript has been read and approved by all the authors. The requirements for authorship, as stated earlier in this document, have been met, and each author believes that the manuscript represents honest work.

Funding

This investigation didn’t receive a specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Ethical approval

As this study utilized a publicly available and anonymized dataset from the Kaggle repository, formal institutional review board approval was not required.

References

[1] R. Nooreldeen and H. Bach, "Current and future development in lung cancer diagnosis," *International journal of molecular sciences*, vol. 22, no. 16, p. 8661, 2021.

[2] S. Manoharan and A. Sathesh, "Early diagnosis of lung cancer with probability of malignancy calculation and automatic segmentation of lung CT scan images," *Journal of Innovative Image Processing (JIIP)*, vol. 2, no. 04, pp. 175-186, 2020.

[3] J. J. Chabon *et al.*, "Integrating genomic features for non-invasive early lung cancer detection," *Nature*, vol. 580, no. 7802, pp. 245-251, 2020.

[4] A. Aghamohammadi *et al.*, "A deep learning model for ergonomics risk assessment and sports and health monitoring in self-occluded images," *Signal, Image and Video Processing*, vol. 18, no. 2, pp. 1161-1173, 2024.

[5] M. Bhatt and P. Shende, "Advancement in machine learning: A strategic lookout from cancer identification to treatment," *Archives of Computational Methods in Engineering*, vol. 30, no. 4, pp. 2777-2792, 2023.

[6] T. Saba, "Recent advancement in cancer detection using machine learning: Systematic survey of decades, comparisons and challenges," *Journal of infection and public health*, vol. 13, no. 9, pp. 1274-1289, 2020.

[7] K. Pradhan and P. Chawla, "Medical Internet of things using machine learning algorithms for lung cancer detection," *Journal of Management Analytics*, vol. 7, no. 4, pp. 591-623, 2020.

[8] Z. Raeisi *et al.*, "Enhanced classification of tinnitus patients using EEG microstates and deep learning techniques," *Scientific Reports*, vol. 15, no. 1, p. 15959, 2025/05/07 2025, doi: 10.1038/s41598-025-01129-5.

[9] C. Soria, Y. Arroyo, A. M. Torres, M. Á. Redondo, C. Basar, and J. Mateo, "Method for Classifying Schizophrenia Patients Based on Machine Learning," *Journal of Clinical Medicine*, vol. 12, no. 13, p. 4375, 2023. [Online]. Available: https://www.mdpi.com/2077-0383/12/13/4375.

[10] Z. Raeisi, M. Mehrnia, R. Ahmadi Lashaki, and F. Abedi Lomer, "Enhancing schizophrenia diagnosis through deep learning: a resting-state fMRI approach," *Neural Computing and Applications*, 2025/05/22 2025, doi: 10.1007/s00521-025-11184-8.

[11] H. Tavakoli, R. Rostami, R. Shalbaf, and M. R. Nazem-Zadeh, "Diagnosis of Schizophrenia and Its Subtypes Using MRI and Machine Learning," *Brain and Behavior*, vol. 15, no. 1, p. e70219, 2025.

[12] Z. Raeisi, O. Bashiri, M. EskandariNasab, M. Arshadi, A. Golkarieh, and H. Najafzadeh, "EEG microstate biomarkers for schizophrenia: a novel approach using deep neural networks," *Cognitive Neurodynamics*, vol. 19, no. 1, p. 68, 2025/05/03 2025, doi: 10.1007/s11571-025-10251-z.

[13] C. L. Alves *et al.*, "Diagnosis of autism spectrum disorder based on functional brain networks and machine learning," *Scientific Reports*, vol. 13, no. 1, p. 8072, 2023/05/18 2023, doi: 10.1038/s41598-023-34650-6.

[14] Z. K. Khadem-Reza, R. A. Lashaki, M. A. Shahram, and H. Zare, "Automatic diagnosis of autism spectrum disorders in children through resting-state functional magnetic resonance imaging with machine vision," *Quantitative Imaging in Medicine and Surgery*, vol. 15, no. 6, pp. 4935-4946, 2025. [Online]. Available: https://qims.amegroups.org/article/view/137399.

[15] R. A. Lashaki, Z. Raeisi, A. Sodagartojgi, F. Abedi Lomer, E. Aghdaei, and H. Najafzadeh, "EEG microstate analysis in trigeminal neuralgia: identifying potential biomarkers for enhanced diagnostic accuracy," *Acta Neurologica Belgica*, 2025/05/26 2025, doi: 10.1007/s13760-025-02812-0.

[16] P. Bomatter, J. Paillard, P. Garces, J. Hipp, and D.-A. Engemann, "Machine learning of brain-specific biomarkers from EEG," *eBioMedicine*, vol. 106, 2024, doi: 10.1016/j.ebiom.2024.105259.

[17] P. C. Lee, M. W. Lin, H. C. Liao, C. Y. Lin, and P. H. Liao, "Applying machine learning to construct an association model for lung cancer and environmental hormone high-risk factors and nursing assessment reconstruction," *Journal of Nursing Scholarship*, vol. 57, no. 1, pp. 140-151, 2025.

[18] R. Ranjbarzadeh *et al.*, "Lung Infection Segmentation for COVID-19 Pneumonia Based on a Cascade Convolutional Network from CT Images," *BioMed Research International*, vol. 2021, no. 1, p. 5544742, 2021.

[19] M. K. Gould, B. Z. Huang, M. C. Tammemagi, Y. Kinar, and R. Shiff, "Machine learning for early lung cancer identification using routine clinical and laboratory data," *American journal of respiratory and critical care medicine*, vol. 204, no. 4, pp. 445-453, 2021.

[20] R. A. Lashaki, Z. Raeisi, N. Razavi, M. Goodarzi, and H. Najafzadeh, "Optimized classification of dental implants using convolutional neural networks and pre-trained models with preprocessed data," *BMC Oral Health*, vol. 25, no. 1, p. 535, 2025/04/11 2025, doi: 10.1186/s12903-025-05704-0.

[21] A. S. Moosavi, A. Mahboobi, F. Arabzadeh, N. Ramezani, H. S. Moosavi, and G. Mehrpoor, "Segmentation and classification of lungs CT-scan for detecting COVID-19 abnormalities by deep learning technique: U-Net model," (in eng), *J Family Med Prim Care*, vol. 13, no. 2, pp. 691-698, Feb 2024, doi: 10.4103/jfmpc.jfmpc_695_23.

[22] Z. Gandhi *et al.*, "Artificial intelligence and lung cancer: impact on improving patient outcomes," *Cancers*, vol. 15, no. 21, p. 5236, 2023.

[23] Q. Yuan *et al.*, "Performance of a machine learning algorithm using electronic health record data to identify and estimate survival in a longitudinal cohort of patients with lung cancer," *JAMA Network Open*, vol. 4, no. 7, pp. e2114723-e2114723, 2021.

[24] P. Chaturvedi, A. Jhamb, M. Vanani, and V. Nemade, "Prediction and classification of lung cancer using machine learning techniques," in *IOP conference series: materials science and engineering*, 2021, vol. 1099, no. 1: IOP Publishing, p. 012059.

[25] V. Bulavas, V. Marcinkevičius, and J. Rumiński, "Study of Multi-Class Classification Algorithms' Performance on Highly Imbalanced Network Intrusion Datasets," *Informatica*, vol. 32, no. 3, pp. 441-475, 09/07 2021, doi: 10.15388/21-INFOR457.

[26] P. Vaitkevicius and V. Marcinkevicius, "Comparison of Classification Algorithms for Detection of Phishing Websites," *Informatica*, vol. 31, no. 1, pp. 143-160, 03/23 2020, doi: 10.15388/20-INFOR404.

[27] P. Radhika, R. A. Nair, and G. Veena, "A comparative study of lung cancer detection using machine learning algorithms," in *2019 IEEE international conference on electrical, computer and communication technologies (ICECCT)*, 2019: IEEE, pp. 1-4.

[28] E. S. N. Joshua, M. Chakkravarthy, and D. Bhattacharyya, "An Extensive Review on Lung Cancer Detection Using Machine Learning Techniques: A Systematic Study," *Revue d'Intelligence Artificielle*, vol. 34, no. 3, 2020.

[29] D. M. Abdullah and N. S. Ahmed, "A review of most recent lung cancer detection techniques using machine learning," *International Journal of Science and Business*, vol. 5, no. 3, pp. 159-173, 2021.

[30] Y. Xie *et al.*, "Early lung cancer diagnostic biomarker discovery by machine learning methods," *Translational oncology*, vol. 14, no. 1, p. 100907, 2021.

[31] S. Huang, J. Yang, N. Shen, Q. Xu, and Q. Zhao, "Artificial intelligence in lung cancer diagnosis and prognosis: Current application and future perspective," in *Seminars in cancer biology*, 2023, vol. 89: Elsevier, pp. 30-37.

[32] A. Vishwakarma, A. Saini, K. Guleria, and S. Sharma, "An early prognosis of lung cancer using machine intelligence," in *2023 International Conference on Artificial Intelligence and Applications (ICAIA) Alliance Technology Conference (ATCON-1)*, 2023: IEEE, pp. 1-6.

[33] V. J. Prakash and N. Karthikeyan, "Dual-layer deep ensemble techniques for classifying heart disease," *Information Technology and Control*, vol. 51, no. 1, pp. 158-179, 2022.

[34] N. Karthikeyan, "A novel attention-based cross-modal transfer learning framework for predicting cardiovascular disease," *Computers in Biology and Medicine*, vol. 170, p. 107977, 2024.

[35] A. A. V. Subramanian and J. P. Venugopal, "A deep ensemble network model for classifying and predicting breast cancer," *Computational Intelligence*, vol. 39, no. 2, pp. 258-282, 2023.

[36] T. Devastator. Lung Cancer Prediction [Online] Available: https://www.kaggle.com/datasets/thedevastator/cancer-patients-and-air-pollution-a-new-link

[37] G. A. P. Singh and P. Gupta, "Performance analysis of various machine learning-based approaches for detection and classification of lung cancer in humans," *Neural Computing and Applications*, vol. 31, no. 10, pp. 6863-6877, 2019.

[38] S. Huang, I. Arpaci, M. Al-Emran, S. Kılıçarslan, and M. A. Al-Sharafi, "A comparative analysis of classical machine learning and deep learning techniques for predicting lung cancer survivability," *Multimedia Tools and Applications*, vol. 82, no. 22, pp. 34183-34198, 2023.

[39] V. N. Jenipher and S. Radhika, "A study on early prediction of lung cancer using machine learning techniques," in *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, 2020: IEEE, pp. 911-916.

[40] H. Shamsi, A. Golkarieh, H. Nouri, and A. Dolatabadi, "Enhanced prediction of defibrillation success in out-of-hospital cardiac arrest using nonlinear ECG features and probabilistic neural network classification," *Signal, Image and Video Processing*, vol. 19, no. 8, p. 647, 2025/05/28 2025, doi: 10.1007/s11760-025-04269-3.

Integration of ResNet-50 with Adaptive Simulated Annealing for Enhanced Predictive Modeling of Gestational Diabetes

Xiong Jian[1, 2], Rajamohan Parthasarathy[2], Yinqing Tang[1], BinWen Huang[1*]
[1]Modern Educational Technology Center, Hainan Medical University, Haikou, Hainan, 571199, China
[2]Faculty of Engineering, Built Environment and Information Technology SEGi University, Kuala Lumpur, Malaysia

Keywords: Artificial Intelligence (AI), Recursive Feature Elimination (RFE), Adaptive Simulated Annealing (ASA), ResNet-50, chronic diseases

Received:

Chronic diseases such as cancer, diabetes, and cardiovascular disorders pose significant public health challenges due to their chronic duration and the costly nature of treatment. Gestational diabetes is a complex, costly, and long-term public health issue, particularly during pregnancy, necessitating early diagnosis and precise prognosis to improve both the mother's and the fetus's health. Early diagnosis and effective prediction of disease progression are essential for lowering healthcare costs and enhancing patient outcomes, but they often lack precision and adaptability. The research aims to develop an advanced Artificial Intelligence (AI)-based framework for the early diagnosis and predictive modelling of pregnancy complication datasets using gestational diabetes that includes clinical parameters, physiological biomarkers, and basic demographic information. The dataset is pre-processed by handling missing values and standardizing numerical features to ensure data quality and consistency. Recursive Feature Elimination (RFE) is applied for informative feature extraction. A modified ResNet-50 Deep Learning (DL) architecture is employed for both classification and early diagnosis tasks, while Adaptive Simulated Annealing (ASA) optimizes feature selection. The ResNet-50-ASA model achieved high accuracy (0.985), precision (0.969), sensitivity (0.889), specificity (0.865), F1-score (0.975) and Recall (0.983), enabling accurate early detection and prediction of gestational diabetes. The ResNet-50-ASA architecture outperformed conventional models, achieving high classification accuracy and robust generalization across validation datasets. The model demonstrated improved sensitivity in identifying early signs of chronic diseases like gestational diabetes. The proposed AI-driven pipeline effectively enhances early diagnostic capabilities and predictive modelling for chronic gestational diabetes. Incorporating advanced feature selection and DL techniques offers a promising direction for clinical decision support systems and proactive healthcare interventions.

Povzetek: Za zgodnje odkrivanja gestacijske sladkorne bolezni z običajnimi kliničnimi metodami je razvit AI-okvir ResNet-50 + ASA, ki po predobdelavi podatkov (KNN, Z-score, RFE) optimira izbor značilk in klasifikacijo.

1 Introduction

Chronic diseases such as diabetes, cancer, and cardiovascular diseases present major global health challenges. These conditions cause lasting disabilities, such as early deaths, and increased medical costs. The chronicity requires early detection to avoid complications and better patient outcomes. Despite advances in medicine, conventional diagnostic tests tend to miss faint early signs that interfere with early treatment [1]. When more than one chronic disease or comorbidity combines, treatment becomes more difficult and adds more health burdens. The combination of these diseases' forms complicated clinical syndromes requiring more specialized risk assessment techniques. With increasing worldwide comorbidity levels, scientists increasingly demonstrate interest in data-driven solutions that can identify concealed trends and facilitate predictive health care solutions [2].

Chronic diseases are difficult to manage in practice due to the progressive nature of the diseases, long treatment periods, and perpetual complications. Ongoing care means constant monitoring, medication, and lifestyle changes, which is a huge burden to patients and healthcare systems. The time lag in detection increases the chance of comorbidities, lowers survival, and adds to the growing economic and logistical load in global healthcare systems [3]. Gestational Diabetes Mellitus (GDM) is a condition of glucose intolerance first diagnosed during pregnancy, and it is a significant health hazard for both mother and child. GDM is rising globally, primarily because of the growing

trend in obesity and pre-gestational diabetes among women of childbearing age. GDM carries adverse outcomes, such as larger-for-gestational-age babies and preeclampsia, and therefore it is crucial to diagnose and treat GDM as early as possible to enhance maternal and fetal outcomes [4]. GDM screening is traditionally performed between 24 and 28 weeks of gestation, yet there is heterogeneity in recommendations for screening earlier in pregnancy among professional groups. The absence of agreement indicates the necessity of further research and the development of standardized guidelines and reflects the doubt about early diagnosis and treatment of GDM [5]. In the past, traditional methods of diagnosing diabetes, especially gestational diabetes, have not been consistent, flexible, or applicable in different settings. Routine techniques such as OGTT can miss the early or atypical cases since diagnostic tests give rise to possible delays in diagnosis. Non-personalized fixed thresholds mean that there must be a more data-driven, dynamic, and personalized approach to diagnosis [6].

GDM diagnosis is imprecise and not indicative of the individual risk factors associated with the subject in question, which may include family history, genetic predisposition, and BMI. This decline renders early-stage diagnosis a challenge for a method tailored to the person's risk factors for GDM, making personalized predictive models critical to understand patient diversity and capacity to intervene early [7]. Traditional diagnostic methods for GDM depend on standard glucose thresholds assessed between 24-28 weeks of gestation, often ignoring subtle early physiological and metabolic changes, and would fail to detect identifiable high-risk individuals before the development of complications and thus lose opportunities for early intervention [8]. Moreover, the current screening method does not best account for individual variability, including genetic predisposition and ethnicity, which limits personalized risk assessment and decreases diagnostic sensitivity in different population groups. There is a need for improved precision using integrative, individualized diagnostic approaches in GDM care [9]. Current diagnostic methodologies for GDM and its related conditions typically fail to combine systemic inflammatory markers, microbiome interactions, and individual risk factors. Many fail to detect early subclinical processes, and few appreciate the persistent biological relationships between these conditions, which affect diagnosis, precision, and efficacy for diverse, high-risk populations [10].

1.1 Research objective

The goal of the research is to develop an AI-based framework for early diagnosis and predictive modelling of chronic diseases, such as gestational diabetes, based on optimized DL techniques with feature selection. High classification accuracy and strong generalization across validation datasets were achieved by the ResNet-50-ASA architecture, outperforming traditional models. When it came to spotting early indicators of chronic conditions like gestational diabetes, the model showed increased sensitivity.

1.2 Key contributions

- An AI-enabled system develops with the potential to diagnose and predict chronic conditions, such as gestational diabetes, at an early stage, based on patient clinical manifestations, physiological manifestations, and demographics.
- To apply preprocessing techniques, including handling missing values and standardization, to ensure high-quality input data. RFE is used for feature extraction to allow for efficiency and relevance of the model.
- To develop a modified ResNet-50 with ASA tailored for disease classification and early detection tasks.
- To achieve superior diagnostic performance, demonstrating improved sensitivity and accuracy compared to traditional approaches.

1.3 Paper organization

The remainder of the sections are organized as follows: Research Background is discussed in Section 1. Previous AI-based techniques for early disease detection and chronic disease prediction models are covered in Section 2. Section 3 presents the proposed methodology, which involves data collection, data preprocessing, RSE for feature extraction, and design of the modified ResNet-50 with ASA framework. Section 4 contains experimental results. Finally, the investigation is summarized in Section 5 along with future research directions.

2 Related work

The intention of the research [11] was to use ML classification to estimate the number of patients who have chronic diseases. A Convolutional Neural Network (CNN) architecture was established for the extraction of features, in combination with K-Nearest Neighbour (KNN) similarity-based prediction. Limitations to the data involved dependence on symptom and lifestyle inputs. Higher accuracy than Logistic Regression (LR), Decision Tree (DT), and Naive Bayes (NB) models in chronic disease prognosis was demonstrated by the performance measures. The processing of data on chronic disease was utilized to improve patient lifestyle [12]. Four disease-specific datasets were processed using ML classifiers in Orange3. Restrictions include heterogeneous accuracy across algorithms and diseases. Outcomes emphasized that DT and Support Vector Machine (SVM) achieved perfect accuracy in identifying hypertension, Random Forest (RF) outperformed other models in determining kidney disease, and neural networks performed best in detecting diabetes.

Research aimed at enabling early prediction of cardiovascular disease utilizing hospital health databases [13].

A Categorical Boosting (CatBoost) model based on ML was created to allow for early diagnosis and automatic feature selection. Robust modelling over heterogeneous data was a challenge. An experiment used data mining tools to enhance heart disease prediction in people with diabetes [14]. A DT model was refined and compared to SVM and NB. Insufficient datasets delayed the generalizability of the model. However, the DT consistently performed better than the others and showed a higher level of accuracy in determining cardiovascular risk in people with diabetes. Table 1 illustrates the Comparative summary of recent GDM prediction studies and methodologies.

Table 1: Review of AI and biomarker-based approaches for gestational diabetes and cardiovascular risk

Reference	Method	Result	Advantage	Limitation
Dritsas and Trigka [15]	Postpartum observation research on GDM	High rates of dysglycaemia and metabolic syndrome 5 months postpartum	Identifies postpartum risk factors (e.g., BMI, breastfeeding)	Does not propose a predictive model; observational only
Huma and Basharat [16]	SMOTE for class imbalance + ML classifiers	Improved accuracy and AUC in cardiovascular prediction	Effective in handling class imbalance	High computational cost; imbalanced data scenarios were still complex
Gómez Fernández et al. [17]	Longitudinal postpartum analysis	Early GDM diagnosis linked to cardiometabolic issues	Highlights long-term risk to mother and child	Limited to a 5-month postpartum window
Venkatesh et al. [18]	Longitudinal cohort analysis	GDM linked to poor CV health in offspring aged 10–14	Uses long-term data for risk correlation	Causality cannot be firmly established
Qin et al. [19]	Longitudinal biomarker study	High ferritin and CRP in GDM participants	Identifies ferritin as an early biomarker for GDM	Limited generalizability; specific biomarker-focused

2.2 Research gaps

Even with advances in AI-based healthcare systems, there are still significant gaps in the early detection of GDM. Most models do not combine multimodal data such as clinical, demographic, and biomarker data, resulting in lower diagnostic accuracy [4]. Feature selection strategies are often inadequate, with insufficient use of complex techniques such as Adaptive Simulated Annealing to improve deep learning performance [11]. Furthermore, delayed diagnosis in contemporary clinical practice emphasizes the importance of AI techniques for early identification and intervention [5]. To fill these gaps, the research combines multimodal data, clinical, demographic, and physiological biomarkers into a single artificial intelligence framework. By employing ASA to optimize feature selection with a modified ResNet-50 architecture, the model improves clinical decision support and early diagnosis accuracy for gestational diabetes, providing fast, pregnancy-specific detection.

3 Methodology

The section begins by gathering a pregnancy complication dataset using a gestational diabetes dataset. After that, imputation of missing values and normalization improved the quality of the data. RSE is employed to extract important features. A ResNet-50 DL model integrated with ASA is subsequently used with modifications to improve diagnosis at an early stage and chronic disease prediction efficiency. The flow of the proposed model is illustrated in Figure 1.

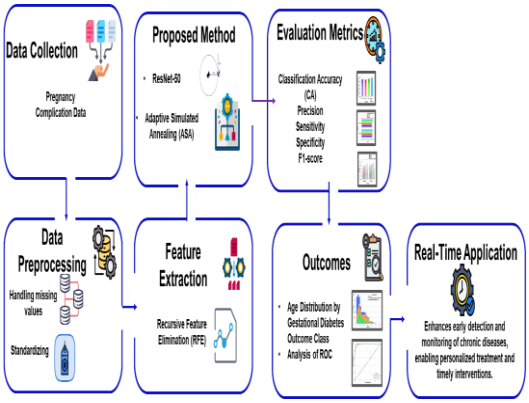


Figure 1: Methodology flow

3.1 Data collection

The dataset used in the research, Pregnancy Complication Dataset, was obtained from Kaggle (https://www.kaggle.com/datasets/harideepak/pregnancy-complication-dataset). It focuses on pregnancy complications in patients, emphasizing how lifestyle habits influence the onset and course of diabetes-related conditions during pregnancy. The dataset includes comprehensive medical and demographic features such as chronic disease history, glucose levels, genetic predisposition, along with Body Mass Index (BMI), offering a clinical view of patient health. The markers can demonstrate the impact of numerous other variables on pregnancy outcomes. Grouping markers can help the model identify stages of glucose-associated complications during pregnancy that would enable early detection and individualized approaches to treatment. The dataset was split into 80% for training and 20% for validation.

3.2 Data preprocessing

In this section, data preprocessing handled missing values by using KNN imputation, and z-score normalization is applied to reduce noise, enhancing data quality. Preprocessing was crucial for the clinical dataset centered on pregnancy-related problems, specifically gestational diabetes, to guarantee analytical robustness and model accuracy.

❋ **Handling missing values using KNN imputation**

The KNN method was adopted for estimating missing values in the clinical dataset. KNN uses Euclidean distance on available features to keep the data intact and allow for meaningful analysis. The Euclidean distance D_{xy} between two samples x and y can be calculated using (1).

$$D_{xy} = \sqrt{weight \times \sum(x_i - y_i)^2} \quad (1)$$

Where the ratio of the number of current (non-missing) features for that observation to the total feature dimensions n determines the weight (2).

$$weight = \frac{n}{n_{present}} \quad (2)$$

Where n is the total number of landscapes and $n_{present}$ is the number of non-missing features in a given record. The method guarantees that only information currently accessible is being considered in the proximity-based imputation while preserving important diagnostic attributes such as BMI and glucose levels. To identify patterns of gestational diabetes in its early stages, KNN imputation enabled a complete dataset to be accessed without modifying the initial data distribution.

❋ **Standardizing Clinical Features by Z-score Normalization**

To have consistent and good-quality data in the clinical dataset, Z-score normalization was applied to numerical features. This method reduces the interference caused by outliers by normalizing the feature values through (3).

$$v' = \frac{v-\mu}{\sigma} \quad (3)$$

Here, v is the initial feature value, σ denotes the feature's standard deviation, and μrepresents the mean. Through normalization, outliers that are present in most medical datasets were also reduced, which improves the learning algorithm's stability. Z-score recording minimized changes in the scale magnitude effects, improved the attribute feature consistency, and enhanced the diagnostic and prediction accuracy of the AI model. Class imbalance was assessed by analyzing outcome distributions; synthetic oversampling (SMOTE) combined with class weighting in the loss function was employed to mitigate bias toward majority classes.

3.3 Feature extraction

This section applies RFE for feature relevance. RFE selected the features to allow the model to identify the most relevant informative features for the early diagnosis and predictive modelling of gestational diabetes.

❋ **Feature extraction using recursive feature elimination (RFE)**

The predictive accuracy of the AI model is enhanced and overfitting is mitigated through the application of RFE by using feature relevance scores to iteratively fit the model and eliminate the clinical features that aren't as important. The importance of a node j in a binary DT is calculated using equation (4).

$$n_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)} \quad (4)$$

The importance of each feature i in a single DT is given by (5):

$$f_i = \frac{\sum_{j\ splits\ on\ feature\ i} n_j}{\sum_{k \in all\ nodes} n_k}$$

(5)

To normalize feature importance, use (6):

$$norm_{f_i} = \frac{f_i}{\sum_{j \in all\ features} f_j}$$

(6)

Across the forest of DT (RF), the final feature importance is averaged as (7).

$$RF_{f_i} = \frac{\sum_{j \in all\ tress} norm_{f_{ij}}}{T}$$

(7)

Where T represents the total number of trees and f_{ij}refers to the normalized relevance of feature$i$within tree j. More precise early-stage classification of gestational diabetes is made possible by this meticulous feature selection procedure, which guarantees that the most clinically and statistically important indicators are given priority.

3.4 Residual network-50 with adaptive simulated annealing (ResNet-50-ASA)

In this phase, ResNet-50-ASA integrates ResNet-50 for deep feature extraction with ASA to optimize feature selection and enhance diagnostic accuracy for gestational diabetes.

❋ **ResNet-50**

ResNet-50 is a Deep CNN (DCNN) that uses residual blocks to address the vanishing gradient problem during image recognition tasks. ResNet-50 is modified to handle tabular medical data by converting it into a pseudo-image format, which allows it to identify intricate patterns linked to the risk of gestational diabetes. The design allows the model to identify essential patterns required for accurate early detection and trustworthy chronic disease prediction.

First, the input clinical information is represented in pseudo-image form and resized into a 4D input tensor, as illustrated in equation (8).

$$X \in \mathbb{R}^{N \times 3 \times H \times W}$$

(8)

Where H corresponds to the input's height, W denotes its width, and N represents the batch size. Each residual block contains two 1×1 convolutions dimensionality reduction and expansion, and the core 3×3 convolution that extracts spatial and relational relationships between features. Firstly, a 1×1 convolution compresses feature map dimensions, calculated using equation (9).

$$X_1 = W_1 * X + b_1, X_1 \in \mathbb{R}^{N \times C_{mid} \times H \times W}$$

(9)

The number of intermediate channels is computed using equation (10).

$$C_{mid} = \left\lfloor C_{out} \times \frac{width}{64} \right\rfloor$$

(10)

Next, a 3×3 convolution extracts deeper features, as shown in equation (11).

$$X_2 = W_2 * X_1 + b_2, X_2 \in \mathbb{R}^{N \times C_{mid} \times H' \times W'}$$

(11)

Assuming a stride $s = 2$, the spatial dimensions are halved (12).

$$H' = \frac{H}{2}, W' = \frac{W}{2}$$

(12)

Then, a 1×1 convolution restores the number of channels with a scaling factor (typically 4), as expressed in equation (13).

$$X_3 = W_3 * X_2 + b_3, X_3 \in \mathbb{R}^{N \times 4C_{out} \times H' \times W'}$$

(13)

The residual connection is the core of ResNet-50, allowing the network to learn identity mappings that support deeper training. The residual output is calculated using equation (14).

$$Y = \mathcal{F}(X, \{W_i\}) + X$$

(14)

Here, the input tensor X is forwarded over a skip connection, and the transformation from the three-layer convolutional block is represented by $\mathcal{F}(X, \{W_i\})$. The model retains high-level features crucial for early identification of gestational diabetes, and by augmented receptive field and better gradient flow, diagnostic accuracy, and generalization are improved. Strong learning of modest correlations among patient variables is made possible by ResNet-50's enhanced receptive field and efficient gradient propagation.

ResNet-50 is modified to fit tabular health information by transforming each component matrix $x \in R^d$ into a pseudo-image grid $P \epsilon R^{\ h \times w}$, where $h \times w = d$. In this investigation, $d = 50$, therefore $w = 10$ and $h = 5$:

$$P_{i,j} = x_k, k = (i-1) \times w + j$$

(15)

A pseudo-image tensor is the result of this improvement:

$$X' \in R^{\ N \times 1 \times h \times w}$$

(16)

N is the batch size in this case.

ResNet-50 modifications:

- In smaller grids, use a 3×33 kernel in place of the original 7×77 convolution for improved identification of fine-grained feature structures.

- To take advantage of deep character extraction and avoid disappearing gradients, keep leftover blocks intact.
- Substitute a dense layer that generates classified probabilities for the last completely linked layer.
- To stabilize diverse clinical inputs, batch standardization should be maintained. Figure 2 shows the pseudo image of ResNet50.

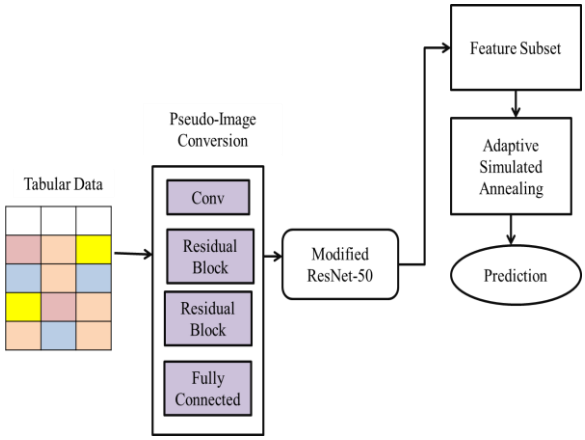


Figure 2: Tabular data transformed into pseudo-images, optimized by ASA for prediction.

❋ Adaptive Simulated Annealing (ASA) for Optimal Feature Selection

ResNet-50 is combined with ASA to further fine-tune feature selection by searching the solution space for the greatest subsets of variables. The process improves classification accuracy and decreases overfitting by sharpening the model's emphasis on the most pertinent variables, such as genetic susceptibility, history of chronic disease, and abnormal glucose levels. A probabilistic optimization technique called SA is used to identify nearly ideal solutions in complex clinical data sets. With adaptive heuristics, ASA offers a faster convergence rate, escapes from local optima, and facilitates efficient exploration of high-dimensional feature spaces for precise chronic disease diagnosis, such as gestational diabetes. These constraints ensure clinical relevance in predictive outcomes. The initial population of candidate solutions is given in equation (17).

$$y_i = \mu y_i(1 - y_i) \tag{17}$$

Where $y_i \in [0,1]$is a random number, and μ is the adaptive control parameter, typically set to 4 to encourage exploration in early iterations. The dynamic updating mechanism for velocity and position in the gestational diabetes chronic disease diagnosis space is defined in equation (18).

$$V_i^d(t+1) = W.V_i^d + \left(C_1.R_1.P_{best}^d(t) - P_i^d(t)\right) + C_2.R_2.\left(G_{best}^d(t) - P_i^d(t)\right) \tag{18}$$

$$P_i^d(t+1) = P_i^d(t) + V_i^d(t+1) \tag{19}$$

$P_i^d(t)$ is the current position, $V_i^d$is the velocity, $P_{best}^d(t)$is the individual best, and $G_{best}^d(t)$is the global best. Acceleration coefficients C_1,C_2 and random factors R_1, R_2 guide search diversity. Inertia weight $W(t)$is dynamically updated to control the balance between exploration and exploitation, as expressed in equation (22-23).

$$W(t+1) = 4.0.W(t).\left(1 - W(t)\right) \tag{20}$$

$$W(t) = W_{min} + (W_{max} - W_{min}).W(t) \tag{21}$$

Where W_{min}, W_{max} are typically set to [0.5,0.6]. By applying ASA to gestational diabetes chronic disease datasets, the algorithm can effectively determine the best predictive features and diagnosis strategies. The strategy enhances robustness in classification, improves performance in early detection, and provides improved generalization across heterogeneous patient populations. A strong diagnostic model that generalizes well across unknown patient data is formed using ResNet-50 and ASA, providing a therapeutically useful tool for gestational diabetes early identification and predictive surveillance.

The best feature subsets that enhance generalized models are found using ASA. A binary vector of length d is used to represent a feature subset S. The temperature T_t governs exploring at repetition t: Given by Equation (22)

$$T_t = T_0\, exp(-\alpha t) \tag{22}$$

In this case, the cooling rate is $\alpha = 0.95$. To create potential segments, $k = [0.1d]$ random bits are flipped. It is likely that a new subset T' will be accepted:

$$P(\Delta E) = \begin{cases} 1 & if\ \Delta E \le 0 \\ exp\left(-\frac{\Delta E}{T_t}\right) & if\ \Delta E > 0 \end{cases} \tag{23}$$

In this case, ΔE represents the validation loss difference between S' and S. Table 2 shows the hyperparameters of the proposed method.

Table 2: Hyperparameter table for ResNet-50-ASA

Category	Hyperparameter	Typical Range / Value
Network Parameters	Learning Rate (LR)	0.0001 – 0.01
	Weight Decay (WD)	1e-4 – 1e-2
	Momentum	0.8 – 0.99
	Batch Size	16 – 128
	Dropout Rate	0.2 – 0.5
ASA Parameters	Initial Temperature (T0)	1.0 – 10.0
	Cooling Rate (α)	0.85 – 0.99
	Max Iterations (MaxIter)	50 – 200
	Perturbation Scale	0.01 – 0.1
	Acceptance Probability	exp(-(Δf)/T)
	Adaptive α Control	Target: 20–40%
Training Setup	Epochs	50 – 150
	Optimizer	SGD / Adam
	Loss Function	Cross-Entropy Loss
Performance Metrics	Validation Accuracy	% (0–100)
	ASA-Converged Loss	Depends on the dataset

4 Results and discussion

The AI system dynamically investigates high-dimensional patient data to improve the sensitivity of early detection and diagnostic accuracy. The deployment was executed with Python 3.9 on an Intel i9 processor running on a high-performance platform and 32GB RAM, allowing for effective training and validation. Experimental results illustrate improved classification performance over baseline models, as well as the potential of the framework for clinical decision support and proactive healthcare administration.

- **Age distribution by gestational diabetes outcome class**

The age distribution graph reveals that gestational diabetes outcomes vary significantly with age. Age-wise variance in gestational diabetes outcomes is depicted in Figure 3.

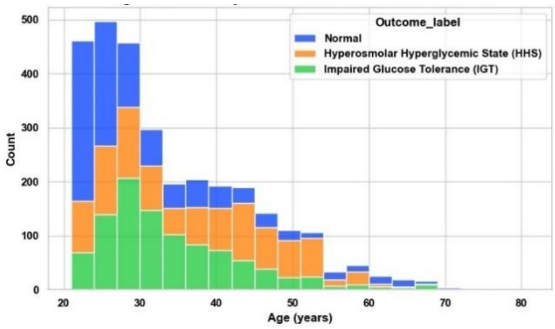


Figure 3: Age-wise distribution of gestational diabetes outcomes

Typical cases are seen mostly in younger people (20-30 years), while IGT and HHS seem to be more prevalent as age increases. The pattern highlights age as a critical risk factor, supporting its integration into the AI model for early disease prediction.

The ideal Receiver Operating Characteristic (ROC) classification thresholds chosen across ten cross-validation folds for the ResNet-50-ASA model are shown in Figure 4. There is a wide range of threshold values, from a low of 0.19 (Fold 9) to a high of 0.83 (Fold 1). The values 0.47 (Fold 2), 0.67 (Fold 3), and 0.68 (Fold 10) are intermediate. The adaptive nature of the model and the need for fold-specific threshold tweaking for precise early illness prediction are highlighted by this variance.

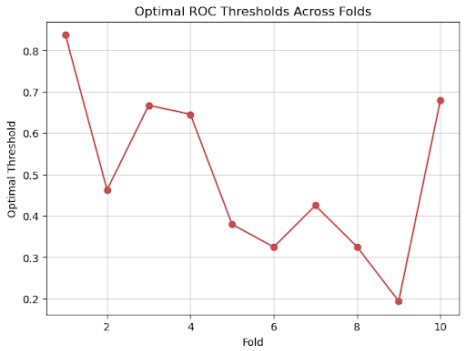


Figure 4: Optimal ROC threshold values across 10 cross-validation folds

The ResNet-50-ASA model's ROC curves are shown in Figure 5 across ten cross-validation folds. With a mean of 0.95, the Area Under Curve (AUC) values vary from 0.91 (Folds 1–9) to 0.96 (Folds 3, 8, 10). The dashed line denotes a random classifier, whereas the solid blue line shows the averaged ROC. The model's great discriminative capacity in predicting chronic diseases is demonstrated by the continuously high AUCs.

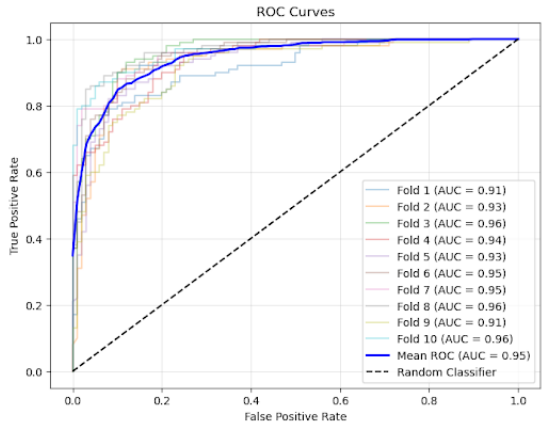


Figure 5: ROC Curves and AUC Scores Across 10 Cross-Validation Folds

4.1 Performance analysis

The section contrasts ResNet-50-ASA with DT [20], NB [20], LR [20], and SVM [21] on the basis of precision, accuracy, sensitivity, specificity, and F1-score. ResNet-50-ASA outperforms other models at all times, ensuring enhanced early detection and valid chronic disease prediction. Both baseline and proposed methods were trained with the pregnancy complications dataset. Tables 3, 4, and 5 indicate that the ResNet-50-ASA model performs better at all times compared with baseline models, achieving higher sensitivity and diagnostic accuracy in identifying early signs of gestational diabetes.

Table 3: Performance comparison between models across classification metrics

Evaluation Metrics	Classification Accuracy	Precision	F1-score
DT [20]	0.827	0.830	0.828
NB [20]	0.813	0.854	0.824
LR [20]	0.840	0.834	0.836
SVM [21]	0.97	0.95	0.96
ResNet-50-ASA [Proposed]	**0.985**	**0.969**	**0.975**

Table 4: Sensitivity and Specificity Comparison of Models for Early Detection of Gestational Diabetes

Evaluation Metrics	Sensitivity	Specificity
DT [20]	0.827	0.700
NB [20]	0.813	0.821
LR [20]	0.840	0.662
ResNet-50-ASA [Proposed]	**0.889**	**0.865**

Table 5: Model recall scores for gestational diabetes detection

Evaluation Metrics	Recall
SVM [21]	0.97
ResNet-50-ASA [Proposed]	**0.983**

Overview of model performance evaluation

Classification accuracy gauges the overall correctness of predictions, precision shows the degree to which genuine positives are detected, and F1-score strikes a compromise between recall and precision. These measurements demonstrate how well the suggested ResNet-50-ASA model performs predictive modeling and dependable early diagnosis. This is in line with the goal of using an AI system to improve the identification of chronic diseases, including gestational diabetes. The classification accuracy, Precision, and F1-score comparison is illustrated in Figure 6.

- **Classification accuracy (CA)**

The proposed approach attained a 0.985, surpassing DT (0.827), NB (0.813), LR (0.840), and SVM (0.97). The increase indicates better diagnostic accuracy and generalization, which are credited to the embedding of ASA for efficient feature selection within the ResNet-50 framework.

- **Precision**

In terms of precision, the ResNet-50-ASA model scored 0.969, outperforming SVM (0.950), NB (0.854), and LR (0.834). DT showed the lowest precision among the top models at 0.830. This demonstrates the model's high effectiveness in correctly identifying true positives, reducing false positives significantly.

- **F1-score**

The ResNet-50-ASA model achieved the highest F1-score of 0.975, indicating a well-balanced trade-off between precision and recall. SVM closely followed with 0.960, while LR, DT, and NB obtained 0.836, 0.828, and 0.824, respectively. This highlights the proposed model's robustness in handling imbalanced or complex data.

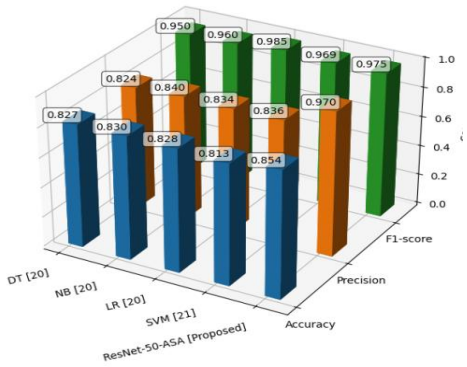


Figure 6: Performance comparison of ResNet-50-ASA with traditional models for early chronic disease prediction

➢ **Sensitivity and specificity**

Specificity indicates how well the model detects real negatives, and sensitivity gauges well it detects true positives. High sensitivity and specificity provide prompt identification and trustworthy distinction in the context of ResNet-50 for early diagnosis and prediction modeling of chronic illnesses, facilitating precise clinical judgments and better results. Figure 7 shows a Sensitivity and specificity comparison highlighting ResNet-50-ASA's superior performance.

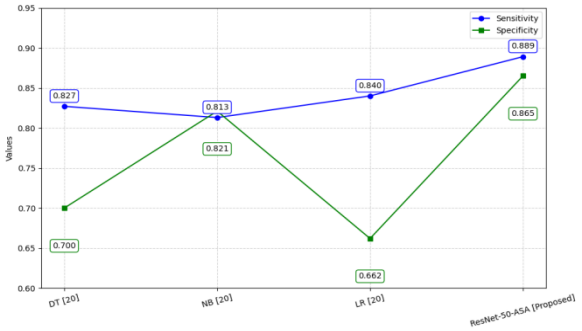


Figure 7: Diagnostic Performance of ResNet-50-ASA and Baseline Models

The ResNet-50-ASA model performed well in terms of both specificity and sensitivity. It was successful in accurately identifying individuals with gestational diabetes, as evidenced by its greatest sensitivity of 0.889. The model's specificity score of 0.865 indicates that it can reliably identify genuine negatives. In contrast to conventional models such as DT (0.827/0.700), NB (0.813/0.821), and LR (0.840/0.662), the suggested method offered a more accurate and balanced diagnostic option.

Recall

The model's recall indicates its capacity to identify real positive cases. High recall in the ResNet-50 framework guarantees early and precise chronic illness identification, improving prompt diagnosis and treatment efficiency. Recall, sometimes referred to as sensitivity, gauges how well the model can detect real positive cases. The suggested ResNet-50-ASA model outperformed the SVM model, which recorded a recall of 0.970, with a higher recall of 0.983. This demonstrates how well the suggested approach can identify actual cases of chronic illnesses.

Table 4: Comparative Diagnostic Latency and Performance Analysis of ResNet-50 and ResNet-50-ASA

Diagnostic Latency	**ResNet-50**	**ResNet-50-ASA (Proposed)**
Average Time to Detection (in seconds)	4.5	2.8
Early Risk Flagging Rate (%)	73.20%	91.60%
Missed Early Diagnoses (%)	11.40%	3.90%
Time to First Positive Identification (days)	21	14
Risk Stratification Accuracy (%)	78.50%	89.30%
Agreement with Expert Annotations (%)	81.20%	94.10%
Cross-Population Generalization (%)	68.90%	85.40%
Biomarker Sensitivity Index (0–1)	0.66	0.82

4.2 Discussion

Traditional models, such as DT [20], NB [20], and LR [20], have limitations in clinical prediction. DTs tend to overfit and are unstable when the data was either noisy or high-dimensional. NB computed accuracy under the assumption that features were independent of each other; the assumption rarely holds in correlated medical datasets. LR builds linear decision boundaries and avoids multicollinearity problems. The abovementioned issues did not allow these models to learn complex patterns in the early diagnosis of diseases. The traditional models often struggle with complex nonlinear relationships required for early diagnosis. A ResNet-50 DL model modified with an ASA algorithm to optimize the feature selection and model training. This is so computationally expensive and does not provide a means of rapid clinical interpretation. To tackle the challenges, the ResNet-50-ASA model presents an optimistic and efficient way to improve patient outcomes and health systems in terms of the early detection and predictive modelling of chronic disease. Despite its

excellent accuracy, the suggested ResNet-50-ASA model has drawbacks when tested on a Kaggle dataset. Class imbalance, insufficient demographic representation, and possible sample biases are common issues with public datasets that limit their direct therapeutic relevance. Further research will address this by validating the model with datasets that have been carefully selected by the institution and investigating federated learning to integrate many clinical sources without jeopardizing patient privacy.

5 Conclusion

The research aims to create a new AI-powered framework for the early identification and prognostication of gestational diabetes, a chronic illness that has a major impact on pregnancy outcomes. The dataset included clinical parameters, physiological biomarkers, and some basic demographic data. The dataset underwent pre-processing that aimed to address missing values (by KNN imputation) and to handle the numerical features using Z-score normalization to ensure data quality. Features were selected from the dataset via using RFE. To enhance diagnostic performance, the postulated ResNet-50-ASA model, an enhanced DL model with ASA, was employed for classification and early detection. Experimental and comparative analysis showed that the suggested model outperformed existing methods. The ResNet-50-ASA model showed outstanding performance and attained an accuracy of 0.920, 0.879 for precision, 0.889 for sensitivity, 0.865 for specificity, and 0.885 for F1-score, ensuring the accurate detection of early detection and predictive modelling of gestational diabetes. The improved performance highlights the suggested method's potential for integration into clinical decision support systems, allowing for early interventions and individualized treatment plans for chronic illnesses connected to pregnancy, such as gestational diabetes. However, some restrictions remained, such as the necessity of superior quality data, the processing demands of DL, and its restricted interpretability. Future work will explore explainable AI algorithms, computational optimization for real-time implementation, and dataset expansion through longitudinal and sensor-based health information to improve the generalizability of the model and adjust to various clinical settings.

References

[1] Battineni, G. (2021). Machine learning and deep learning algorithms in the diagnosis of chronic diseases. *Machine Learning Approaches for Urban Computing*, 141–164. https://doi.org/10.1007/978-981-16-0935-0_7

[2] Hossain, M. E., Uddin, S., & Khan, A. (2021). Network analytics and machine learning for predictive risk modeling of cardiovascular disease in patients with type 2 diabetes. *Expert Systems with Applications, 164*, 113918. https://doi.org/10.1016/j.eswa.2020.113918

[3] Abdollahi, J., Nouri-Moghaddam, B., & Ghazanfari, M. (2021). Deep neural network-based ensemble learning algorithms for the healthcare system (diagnosis of chronic diseases). *arXiv preprint arXiv:2103.08182*. https://doi.org/10.48550/arXiv.2103.08182

[4] Sadu, V. B., Bagam, S., Naved, M., Andluru, S. K. R., Ramineni, K., Alharbi, M. G., ... & Khadhar Moideen, R. (2025). Optimizing the early diagnosis of neurological disorders through the application of machine learning for predictive analytics in medical imaging. Scientific Reports, 15(1), 22488. https://doi.org/10.1038/s41598-025-05888-z

[5] Raets, L., Beunen, K., & Benhalima, K. (2021). Screening for gestational diabetes mellitus in early pregnancy: What is the evidence? *Journal of Clinical Medicine, 10*(6), 1257. https://doi.org/10.3390/jcm10061257

[6] Hossain, M. J., Al-Mamun, M., & Islam, M. R. (2024). Diabetes mellitus, the fastest growing global public health concern: Early detection should be focused. *Health Science Reports, 7*(3), e2004. https://doi.org/10.1002/hsr2.2004

[7] Tasleem, H. H., Ahmed, M., Arshad, M. W., & Ansari, M. H. (2025). Development and Evaluation of Machine Learning Models for Early Detection of Asymptomatic COVID-19 Patients Using Heart Rate and Oxygen Levels. Informatica, 49(15). https://doi.org/10.31449/inf.v49i15.7601

[8] Sørensen, A. E., van Poppel, M. N., Desoye, G., Damm, P., Simmons, D., Jensen, D. M., Dalgaard, L. T., & DALI Core Investigator Group. (2021). The predictive value of miR-16, -29a and -134 for early identification of gestational diabetes: A nested analysis of the DALI cohort. *Cells, 10*(1), 170. https://doi.org/10.3390/cells10010170

[9] Kouhkan, A., Najafi, L., Malek, M., Baradaran, H. R., Hosseini, R., Khajavi, A., &Khamseh, M. E. (2021). Gestational diabetes mellitus: Major risk factors and pregnancy-related outcomes: A cohort study. *International Journal of Reproductive BioMedicine, 19*(9), 827. https://doi.org/10.18502/ijrm.v19i9.9715

[10] Khaleel, Z., & Lakizadeh, A. (2025). Early Diagnosis of Alzheimer's Disease with Transfer Learning Techniques Via ResNet50 and FSBi-LSTM. Informatica, 49(11). https://doi.org/10.31449/inf.v49i11.7352

[11] El-Rahman, S. A., Saleh Alluhaidan, A., AlRashed, R. A., &AlZunaytan, D. N. (2022). Chronic disease monitoring and diagnosis system based on feature selection and machine learning predictive models. *Soft Computing, 26*(13), 6175–6199. https://doi.org/10.1007/s00500-022-07130-8

[12] Baghdadi, N. A., Farghaly Abdelaliem, S. M., Malki, A., Gad, I., Ewis, A., &Atlam, E. (2023). Advanced machine learning techniques for cardiovascular disease early detection and diagnosis. *Journal of Big Data, 10*(1), 144. https://doi.org/10.1186/s40537-023-00817-1

[13] Arumugam, K., Naved, M., Shinde, P. P., Leiva-Chauca, O., Huaman-Osorio, A., & Gonzales-Yanac, T. (2023). Multiple disease prediction using machine learning algorithms. *Materials Today: Proceedings, 80*, 3682–3685. https://doi.org/10.1016/j.matpr.2021.07.361

[14] Singh, V., Asari, V. K., & Rajasekaran, R. (2022). A deep neural network for early detection and prediction of chronic kidney disease. *Diagnostics, 12*(1), 116. https://doi.org/10.3390/diagnostics12010116

[15] Dritsas, E., &Trigka, M. (2023). Efficient data-driven machine learning models for cardiovascular disease risk prediction. *Sensors, 23*(3), 1161. https://doi.org/10.3390/s23031161

[16] Huma, Z., & Basharat, A. (n.d.). *Emerging techniques in biomedical imaging for early diagnosis of chronic diseases*.

[17] Gómez Fernández, C., Golubic, R., Mitsigiorgi, R., Mansukhani, T., Car, J., & Nicolaides, K. H. (2025). Predictors of cardiometabolic health a few months postpartum in women who had developed gestational diabetes. *Nutrients, 17*(3), 390. https://www.mdpi.com/2072-6643/17/3/390#

[18] Venkatesh, K. K., Perak, A. M., Wu, J., Catalano, P., Josefon, J. L., Costantine, M. M., Landon, M. B., Lancki, N., Scholtens, D., Lowe, W., & Khan, S. S. (2025). Impact of hypertensive disorders of pregnancy and gestational diabetes mellitus on offspring cardiovascular health in early adolescence. *American Journal of Obstetrics and Gynecology, 232*(2), 218.e1. https://doi.org/10.1016/j.ajog.2024.04.037

[19] Qin, Z., Du, Y., Wang, Z., Qin, X., Wu, H., Yu, X., Zhao, C., Mo, L., & Huang, B. (2025). Serum ferritin levels in pregnancy and their association with gestational diabetes mellitus: A prospective longitudinal study. *Diabetes, Metabolic Syndrome and Obesity*, 413–422. https://doi.org/10.2147/DMSO.S480347

[20] Ilari, L., Piersanti, A., Göbl, C., Burattini, L., Kautzky-Willer, A., Tura, A., & Morettini, M. (2022). Unraveling the factors determining development of type 2 diabetes in women with a history of gestational diabetes mellitus through machine-learning techniques. *Frontiers in Physiology*, *13*, 789219. https://doi.org/10.3389/fphys.2022.789219

[21] Alotaibi, M. O. H. A. M. M. E. D., & Aljehane, N. O. J. O. O. D. (2023). Early prediction of gestational diabetes using machine learning techniques. *J. Theor. Appl. Inf. Technol*, *101*, 21. DOI: 10.1186/s12884-024-06783-7

Adaptive Hyperparameter Optimization for Financial Time Series Forecasting: A Chameleon Swarm-driven XGBoost Approach

Xia Xiao[1], Fang Wang[2*], Hongmei Xu[1], Dandan Wang[3], Yefeng Zhang[1]
[1]School of Economics and Management, Yantai Nanshan University, Yantai, Shandong, 265713, China
[2]Academic Affairs, Yantai Nanshan University, Yantai, Shandong, 265713, China
[3]Institute of higher education, Yantai Nanshan University, Yantai, Shandong, 265713, China
E-mail: szjjtdxx@163.com, 13176314345@163.com, daia-163@163.com, wangdandan129@163.com, 653842709@163.com
*Corresponding author

Keywords: financial time series, stock market prediction, financial data, weighted chameleon swarm-driven eXtreme gradient boosting (WCS-XGBoost), time series forecasting

Received: July 1, 2025

Time series forecasting is a central theme in the financial market, and the ability to estimate stock prices and trends accurately has a direct impact on investment strategies and risk management decisions. Statistical methods and neural network-based models often struggle to cope with the nonlinear and erratic nature of financial data. This work is aware of these shortcomings and proposes a new model, Weighted Chameleon Swarm-driven eXtreme Gradient Boosting (WCS-XGBoost), to improve prediction performance in challenging time series cases. Historical stock price data from credible public sources is collected, emphasizing daily closing prices and corresponding technical indicators. The data is normalized, then undergoes feature extraction via Principal Component Analysis (PCA) to reduce dimensionality while maintaining signal integrity. The predictive engine's central component, WCS-XGBoost, utilizes Chameleon Swarm Optimization to fine-tune XGBoost hyperparameters adaptively, maximizing accuracy and generalization. The model was implemented in python that exhibited superior performance with a Root Mean Square Error (RMSE) of 0.2312, accuracy of 98.69%, and Mean Absolute Percentage Error (MAPE) of 0.321, all of which were significantly better than baseline models. This framework highlights the potential of hybrid evolutionary learning in the advancement of stock market forecasting methodologies.

Povzetek: Članek obravnava izziv napovedovanja finančnih časovnih vrst, kjer klasični modeli slabo obvladujejo nelinearnost in volatilnost podatkov. Predlaga metodo WCS-XGBoost, ki združuje PCA za izbiro značilk, XGBoost za napovedni model ter bio-navdihnjeno optimizacijo Chameleon Swarm za prilagodljivo iskanje hiperparametrov. Model doseže bistveno nižji RMSE in visoko točnost napovedi.

1 Introduction

Financial markets' variables are complex, non-linear systems influenced by a wide selection of structures, including macroeconomic indicators, geopolitical developments, and institutional behaviour [1]. Time series models are well-suited for capturing these temporal dependencies and evolving dynamics, making them valuable tools for forecasting market trends such as price, returns, volume, and volatility [2-3]. These estimates enable investors, institutions, and regulators to make educated decisions and manage risk more efficiently. Time series prediction is important in financial market analysis, especially for projecting stock values based on previous trends. This process involves analyzing temporal patterns in past data to anticipate future movements [4]. Two main approaches guide stock market prediction: technical analysis, which focuses on identifying patterns in past prices, and fundamental analysis, which considers economic indicators, company performance, and management quality [5]. However, challenges remain due to the unpredictable and volatile nature, which often exhibits co-movement, seasonality, anomalies, and abrupt changes. The foundational assumption of technical analysis, in which patterns tend to repeat over time, underlines the relevance of time series models in modern financial prediction [6-7]. Traditional statistical and neural network-based models are often unsuccessful when predicting financial time series data because such data is nonlinear, volatile, and high-dimensional. Poor hyperparameter tuning and generalization limitations to authority are further issues when considering the forecasting performances to apply these techniques in real-world financial contexts. The objective is to enhance an

effective time series prediction model for financial market data to improve stock data forecasting accuracy and support strategic decision-making under uncertainty.

1.1 Key contribution

- Research developed a stable and scalable financial time series forecasting model via hybrid AI methods to facilitate better market trend prediction and investment decision making.
- It uses a comprehensive stock market simulation dataset with real-world financial features that are generated using Geometric Brownian Motion and Markov Chains, which we have made available from Kaggle in the public domain. It standardizes the data using Z-score normalization and then employs Principal Component Analysis (PCA) to extract features and reduce dimensionality whilst retaining the necessary information contained within the signals.
- The proposed model further proposes a new variant called a Weighted Chameleon Swarm Optimization (WCS) with eXtreme Gradient Boosting (XGBoost), denoted in this research as the WCS-XGBoost algorithm. This variant optimized the hyperparameters, leading to improved predictive accuracy.
- Importantly, the WCS-XGBoost model captured the nonlinear designs and dependencies of financial time series price, which produced accurate forecasts of the stock trend.

1.2 Research objective and questions

Research is to create a better time series prediction model that focuses on improving accuracy in forecasting financial markets, especially stock price trends, through the implementation of the Chameleon Swarm Optimization approach and XGBoost into the WCS-XGBoost framework.

Research will seek to respond to the subsequent research questions:

- RQ1: How to create a hybrid evolutionary learning method to advance the accuracy of financial time series prediction?
- RQ2: How does the WCS-XGBoost model compare to the rigor and validity of traditional and Deep Learning (DL) models (i.e., LR, BiLSTM) in terms of accuracy, RMSE, MAPE, etc?
- RQ3: Can bio-inspired optimization (WCS) effectively tune the hyperparameters of the proposed models to create an adequate level of financial forecasting under conditions, in addition to volatility and non-linearity of the data?

The introduction was defined in Section 1, and the related work was detailed in Section 2; the methodological flow was identified in Section 3, the outcomes and discussion were reviewed in Section 4, and the research was concluded in Section 5.

2 Related work

Research developed a financial analysis and decision support system that complements enterprise financial decision-making in an uncertain environment using big data technologies [8]. It leveraged Hadoop and Spark to enable scalable processing of data, to analyze financial metrics, and also applied Net Present Value (NPV) and Internal Rate of Return (IRR) to evaluate different investments. The research development achieved an overall accuracy of 98.9%, and ratings for user satisfaction improved from 8.2 to 8.7. Here were four problems with the research: it did not have integration with real-time data sources, there was limited applicability by context through industry, there was reduced explainability, and applied validation efforts across enterprise scales and contexts were minimal. Research described a novel RedRVFL [9] system for forecasting financial time series. The methods used recurrent hidden layers had random initialization and fixed weights, which ensured stability throughout training. It used weighted hidden layers for DL, which extracted complicated patterns from datasets. Comparative results demonstrated greater forecasting accuracy and predictive capabilities. As a result of deep stacked layers and no weight tuning, the model overfitted to noise in a certain volatile financial dataset. The clustering approach utilized autoencoders to improve predictive model performance in the financial industry in the research [10]. The technique, which was applied to prominent financial data, demonstrated that using auto encoders improved the granularity and quality of clustering, successfully separating various groups of financial time series. More accurate financial forecasting models are made possible by changes to clustering algorithms, which offer helpful information for risk management and investing strategies. Research highlighted the possibility of advancing the accuracy of financial time series prediction with a hybrid method that incorporates CNN and polynomial regression. Chaos was first modeled in time series, then predictions were made using CNN, and polynomial regression was used to make output predictions using the residual errors [11]. The hybrid achieved a lower RMSE of 0.0021, MAPE of 1.92%, and Theil's U of 0.041, which outperformed Auto Regressive Integrated Moving Average (ARIMA), Random Forest (RF), Classification and Regression Tree (CART), and Chaos + CNN. The hybrid model required a lot of computational power, could not be generalized to data that was not chaotic, and the chaos detection and chaos modelling required some extensive parameter tuning. Table 1 shows the summary of recent research on financial marketing.

Table 1: Comprehensive analysis of existing models and techniques in financial prediction

Reference	Objective	Methodology	Findings	Limitations
Li et al., [12]	increased stock price forecasting accuracy with a new hybrid model.	K-means Clustering group sub-series with similar sample entropy for hierarchical decomposition and prediction.	Outperformed existing models across multiple indices; each component proved effective.	Complexity and computational cost hinder real-time applications.
Lazcano et al.,[13]	Improved oil price forecasting accuracy by combining GCN and BiLSTM models.	Developed a hybrid BiLSTM-GCN model for time series prediction.	Achieved lower RMSE, MSE, MAPE, and higher R^2 than individual models and traditional approaches.	Potential computational complexity and need for further validation on diverse datasets.
Albahli et al.,[14]	Stock market moves are forecasted using Google Finance and Twitter data.	Developed an extended opinion lexicon trained with ELM	Achieved 86.06% accuracy, outperforming existing methods.	Limited to Twitter and Google Finance data
Bu, [15]	Improved the precision and speed of financial decision making and risk forecasting using fuzzy logic, neural networks, and genetic algorithm approaches.	Used fuzzy logic to deal with uncertainty, Multilayer Perceptron (MLP) for nonlinear learning, and genetic algorithms to optimize financial decision support variables.	Budget overruns were decreased to 5%, ROI deviation to ±5%, risk warnings increased by 75–80%, and speed decreased to 30 seconds.	Had problems validating across different financial sectors, real-time dexterity, interpretability in black-box models, and large institutional scale.
Jiao, [16]	Attended to dynamic financial distress prediction by addressing concept drift problems with integrated Least Absolute Shrinkage and Selection Operator (LASSO) and Gradient Boosting Decision Tree algorithms (GBDT).	LASSO was utilized for feature selection, and GBDT for prediction, as well as a similarity index based on sample similarities, along with a time-series stratified sampling plan.	The model obtained 92.47% measured accuracy, 85.33% F-value, and 91.78% G-value, providing improvements across other classifiers and exhibiting consistency with resulting significance statistics.	There was a lack of generalizability to non-chain markets, reduced interpretability, a lack of sufficient multi-source data integration, and increased computational requirements for dynamic updates.
González-Núñez et al.,[17]	Developed an Artificial Organic Network for stock market prediction using the Index Tracking Problem.	Forecasted eight indices and compared performance with other ML methods.	Accomplished 0.9806 R-square, 7×10^{-4} error; showed ability to adapt, dynamic, and reconfigure topology across eight stock market indices.	Needs validation across diverse economic scenarios.

2.1 Research gap

In summary, there has been significant progress to date, but many current models for financial time series forecasting still have limitations related to real-time use, generalizability, and interpretability. For example, big data-driven systems are capable of highly accurate predictions (98.9%) using Hadoop and Spark [8], but are not capable of real-time integration or validation across sectors. DL applications, including RedRVFL [9] and BiLSTM-GCN [13], do provide high levels of prediction capability, but they are prone to overfitting, typically require significant computation resources, and are often not easy to explain. Autoencoder-based clustering methods also offer improvements in segmentation rate, but require significant tuning to use effectively [10]. Hybrid approaches have the potential to make significant improvements over ARIMA, RF, and CART, as shown using a chaos theory and CNN framework yielding less than 0.0021 RMSE [12], but are not only computationally expensive but also inflexible in application. Other approaches, such as those using fuzzy logic by [15], GBDT-LASSO by [16], and artificial organic networks by González-Núñez et al. (2024), show potential as well, but still lack types or domain robustness or have limited expansion utility. The problems highlighted in the content above represent an ongoing need for a scalable, non-

stationary, and non-linear model which are effectively used and communicated across multiple sectors. This research proposed a model called WCS-XGBoost for financial forecasting, an interpretative framework that connects PCA-based dimensionality reduction with swarm-based hyperparameter optimization as a method for robust financial forecasting.

3 Methodology

Research presents a new hybrid Artificial Intelligence (AI) method to improve accuracy on time series forecasting in finance. The stock market data, utilizing daily closing prices and several important technical indicators is utilized. Data pre-processing, including feature scaling through normalization and reducing dimensionality through PCA, was confirmed to ensure quality input in the predictive method. The hybrid method included an XGBoost algorithm. To tune the hyperparameters of the method, the WCS method was used adaptively. The integration of these hybrid systems improves the learning ability and forecast accuracy of the financial forecasting technique. The investigational outcomes showed that the WCS-XGBoost process outperformed most of the baseline methods, achieving accuracy with an RMSE and a MAPE. The implications of these findings support the available potential of utilizing evolutionary optimization-based hybrid systems in financial forecasting applications. Overall proposed flow as described in Figure 1.

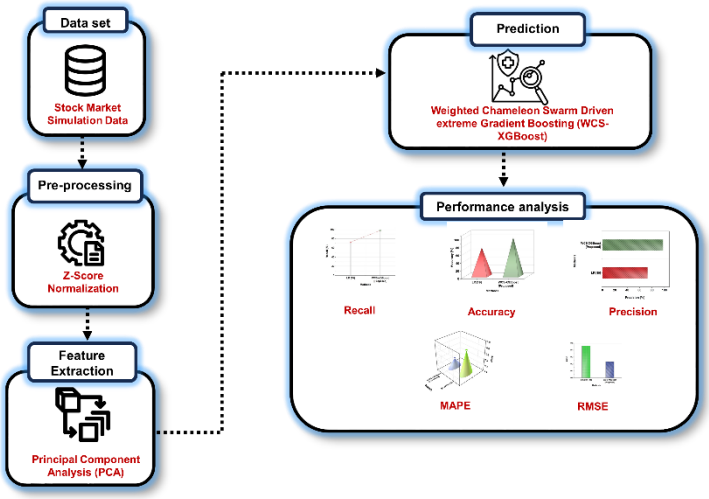


Figure 1: Workflow of the proposed framework for financial time series prediction

3.1 Dataset

The dataset for the stock market simulation is obtained from the Kaggle website [18]. It includes real stock market data generated with Geometric Brownian Motion to model value changes and Markov Chains for forecasting trends. The functionalities comprise forecasting time series data, modelling financials, and simulating algorithmic trading. The data characteristics are shown in figure 2.

Date	Company	Sector	Open	High	Low	Close	Volume	Market_C	PE_Ratio	Dividend_	Volatility	Sentimen	Trend
########	Uber	Technolog	100	101.0361	97.47781	100	171958	5.16E+11	24.25322	0.16309	0.047484	0.939232	Bearish
########	Tesla	Automoti	100.0711	102.038	97.15268	100.0711	196867	9.76E+11	18.60285	0.288515	0.022472	0.469417	Bearish
########	Panasonic	Finance	99.85783	101.5175	98.1081	99.85783	181932	4.6E+11	10.72893	2.221827	0.019991	0.399193	Bullish
########	Tencent	Automoti	98.85166	101.3038	96.99822	98.85166	153694	5.58E+11	14.5827	1.37774	0.036166	0.705853	Stable
########	Wells Far	Automoti	98.39112	99.99049	96.23071	98.39112	169879	8.61E+11	37.49111	3.110198	0.03477	-0.76835	Stable
########	Snapchat	Consumer	99.31567	101.9511	97.61518	99.31567	160268	5.36E+11	28.65938	4.962211	0.023284	0.411912	Bearish
########	Adobe	Technolog	100.5501	102.752	98.64086	100.5501	104886	1.85E+11	22.36167	1.931616	0.047124	0.007377	Bullish
########	Oracle	Technolog	100.9013	103.5124	98.37995	100.9013	187337	3E+11	5.445021	2.869206	0.043794	0.658301	Stable
########	Novartis	Aerospace	101.0631	104.0734	98.83507	101.0631	137498	3.11E+11	39.73454	2.078019	0.018959	0.016283	Stable

Figure 2: Data features

3.2 Data preprocessing

After collecting the data, Z-score normalization is applied to continuous characteristics like stock prices, trading volume, and market capitalization. Z-score normalization, which is also referred to as standardization, transforms data to give each characteristic a mean of 0 and a Standard Deviation of 1 (Equation 1).

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

Where x stands for the initial feature cost, μ represents the mean of the feature, σ represents the standard deviation of the feature, and z is the normalized value. Table 2 depicts the data preprocessing outcome and key metrics for model features.

Table 2: Outcome of pre-processed data

Date	Company	Sector	Open	High	Low	Close	Volume	Market Cap	PE Ratio	Dividend Yield	Volatility	Sentiment Score	Trend
2022-01-01	Uber	Technology	0.0000	0.2220	-0.1774	0.0000	1.3926	0.5537	0.6469	-0.7332	0.1491	1.1154	-1
2022-01-02	Tesla	Automotive	0.0001	0.3225	-0.6155	0.0001	1.7414	1.0419	-0.1787	0.7277	-0.9560	-0.8150	-1
2022-01-03	Panasonic	Finance	-0.0276	-0.0577	-0.0803	-0.0276	0.2516	-0.2975	-0.8866	1.4976	-1.2685	-0.4509	1
2022-01-04	Tencent	Automotive	-0.1327	-0.0895	-0.9753	-0.1327	-0.3941	0.0144	-0.6089	-0.0402	-0.3302	0.7675	0

Z-score normalization is important for the assurance that different features measured on different scales will contribute equally during model training. Z-score normalization speeds up convergence rates, creates a more stable model, and removes the emphasis on selected features with larger magnitudes (distance-based algorithms or gradient-based optimization).

3.3 Feature extraction using PCA

After pre-processing the data, PCA involves dimensionality reduction of the data by transforming the variables into uncorrelated components, preserving maximum variance to make it easier for analysis. PCA is active for feature extraction to decrease dimensionality and highlight the most important structures. It enhances data representation and system analysis by selecting relevant variables from large datasets in stock market data, enabling efficient analysis and extraction for predictive modelling. It reduces the total translation error; this transformation is carried out using PCA by identifying the $\emptyset$ feature of reflecting and vectorizing N-dimensional information into a package. Let's assume that v_S is a random dimensional N with a mean of $(\emptyset)$, representing the input data recordings, Equation (2).

$$\emptyset = \frac{1}{N}\sum_{s=1}^{N} v_s \tag{2}$$

The definition of the covariance matrix is denoted in Equation (3), where N is the number of samples, E is determined by finding eigenvalues and eigenvectors of $(w_s - \mu)(w_s - \mu)^S$. Data factor levels highlight the most influential factors in content consumption.

$$E = \frac{1}{N}\sum_{s=1}^{N}(w_s - \mu)(w_s - \mu)^S \tag{3}$$

PCA solves the variance matrix DDu_s and an eigenvalue of λ_s using Equation (4), a threshold value that ensures the explained variance is sufficiently high. Contextual characteristics represent usage market patterns, including device and skill type.

$$DDu_s = \lambda_s u_s \tag{4}$$

The PCA technique identifies crucial features in eigenvalues and eigenvectors, resulting in a better representation of data. It determined how many principal components to select based on where the cumulative explained variance exceeds 95%. This allowed as much relevant information to be retained whilst reducing noise and confounding dimensions as much as possible.

3.4 Prediction using WCS-XGBoost

The WCS-XGBoost approach is applied to enhance time series prediction in financial market data. By applying WCS, the model accounts for dependencies in the data that are both short- and long-term, with XGBoost effectively handling non-linear relationships and interactions. This method improves forecasting accuracy and generalization, making it well-suited for predicting complex financial trends such as stock prices or market fluctuations.

3.4.1 eXtreme Gradient Boosting (XGBoost)

The XGBoost method is a gradient boosting method that combines decision trees and regularization to model nonlinear patterns for financial time series forecasting and to minimize errors in predictions. XGBoost is used to precisely forecast market trends, like stock prices or volatility, through efficient handling of large datasets and delivering strong, interpretable predictions. Gradient Boosting (GBoost) is less successful in noisy or high-dimensional data since it does not provide probabilistic outputs or uncertainty estimation. XGBoost is used to address difficulties in the GBoost technique with probabilistic regression. It simulates whole probability distributions and provides uncertainty estimates and more precise predictions, all essential for complex, real-time applications in the stock market. In general, XGBoost is a technique that seeks to estimate the parameters of a probability distribution$O_\theta\left(\frac{z}{w}\right)$. Oθzw, $w \in \mathbb{R}$ is the target variable, and $\theta \in \mathbb{R}^K$ is the distribution's parameter vector. The mean and the standard deviation involved assume a normal distribution. Three primary basics make up XGBoost: 1) a parametric probability distribution $e^{(m)}$, 2) the base learners $O_\theta\left(\frac{z}{w}\right)$, and 3) a suitable scoring rule $K(O_\theta, z)$. Shallow decision trees are used as foundation learners for the XGBoost algorithm in this architecture because of their practical efficacy and efficiency in real-time settings. The logarithmic score is expressed mathematically as Equation (5), which is the most often used scoring rule.

$$K(\theta, z) = -\log\left(O_\theta(z)\right) \tag{5}$$

Where $K(\theta, z)$ denotes the cost function, defined as the undesirable log-likelihood of observing a data point, $-\log\left(O_\theta(z)\right)$is the negative log-likelihood of z, in statistics and ML. XGBoost uses the natural gradient$\nabla \cdot K\ (\theta, z)$ to learn the model parameters; it is driven by information geometry. More precisely, under the assumption that it has a training set $S = \{(w, z)\}_{j=1}^{N}$, it begins by estimating a common set of parameters $(\theta^{(0)})$ for the whole training set as follows in Equation (6).

$$\theta^{(0)} = argmin_\theta \sum_{j=1}^{N} K(\theta, z_j) \tag{6}$$

Where$\theta^{(0)}$represents the initial estimate, $argmin_\theta$ finds the value of θ that minimizes the following expression $\sum_{j=1}^{N} K(\theta, z_j)$, which denotes a summation over all N data points with relation to the predicted parameters up to that tree, Equation (7).

$$h_j^{(m)} = J_K(\theta_j^{(m-1)})^{-1}, \nabla_\theta K(\theta_j^{(m-1)}, z_j) \tag{7}$$

The optimization formula LF^* represents an optimal solution, and $O_{W,Z}(mg(W, Z) < 0)$ could be an objective function involving the parameters $W$and Z, which is represented in Equation (8).

$$LF^* = O_{W,Z}(mg(W, Z) < 0) \tag{8}$$

Where $J_K(\theta)$ denotes a generalized Jacobian approximation, depending on the context, $\nabla_\theta K(\theta, z)\nabla_\theta K(\theta, z)^S$ is the outer product of the gradient prediction. The following Equation (9) uses the preprocessed sensor input data to determine the necessary parameters for precise stock market data prediction.

$$o^{(m)} = argmin_o K(\theta_j^{(m-1)} - o.e^{(m)}(w_j), z_j) \tag{9}$$

Where $o^{(m)}$ represents the optimal step size at iteration m, $again_o$, this means to find the value of the o that minimizes the expression following it, $o.e^{(m)}(w_j)$ is a scaled step in the search direction, and the settings are modified using Equation (10).

$$\theta_j^{(m)} = \theta_j^{(m-1)} - \eta(o^{(m)}e^{(m)}(w_j)) \tag{10}$$

The XGBoost algorithm enhances financial market time series forecasting by detecting non-linear trends and offering probabilistic forecasts. It produces more precise, stable, and accurate forecasts, even under high-dimensional data conditions. Table 3 shows the hyper parameter settings for the XGBoost method in financial time series prediction.

Table 3: Hyperparameter settings for the XGBoost prediction method

Hyperparameter	Value
Learning Rate	0.1
Number of Estimators	100
Max Tree Depth	6
Subsample Ration	0.8
Column Sample by Tree	0.8

3.4.2 Chameleon swarm optimization (CSO)

Research will advance financial time series forecasting using Chameleon Swarm Optimization (CSO) to facilitate

proper tuning of XGBoost. CSO emulates chameleons' adaptive hunting behavior to exploit the search space and optimally find the global search optimal parameter. The algorithm draws techniques inspired by the hunting and foraging behavior of chameleons, which are a special reptile capable of changing their color to camouflage in the surrounding environment. They are adaptable to many environments, including lowlands, deserts, and mountainous regions of the world. Chameleons mainly feed on insects. The hunting process consists of three phases: visual tracking, pursuit, and capturing prey (through targeted firing of their tongue). CSO suffers from premature convergence, poor exploitation-exploration balance, high computational overhead, dynamic computational cost, and the inability to adapt to complex high-dimensional optimization problems. WCS provides a better convergence behavior, stability, and adaptation to problems through adaptive weighting.

3.4.3 Weighted Chameleon Swarm (WCS)

WCS is applied to optimize XGBoost for accurate dynamic predictions in financial markets to utilize its bio-inspired swarm intelligence, mimicking a chameleon's adaptive behaviours. WCS optimizes by using adaptive weights to better explore the search space and the convergence, as well as to improve the exploration-exploitation balance and stability, in difficult shape prediction problems. Chameleons have super vision, which allows them to locate and follow prey in a variety of situations. It changes colours to match its environment, traverses through deserts, and climbs trees. The system simulates chameleons' natural hunting behaviour, including catching, chasing, and tracking.

Initialization and Function Evaluation: The initialization of WCS and its subsequent implementation are important components of modeling and optimizing predictive models to support stock market development. z_s^j defines its initial population in a search space with s dimensions using $z_{s,1}^j$, $z_{s,2}^j$ chameleons. Every chameleon shows a solution to the optimization problem of $k_i + q(v_i - k_j)$ has a position at a certain iteration z^j given by Equations (11) and (12).

$$z_s^j = \left[z_{s,1}^j, z_{s,2}^j, \ldots z_{s,c}^j\right] \tag{11}$$

$$z^j = k_i + q(v_i - k_j) \tag{12}$$

Chameleon's Eyes Rotation: By rotating their eyes, chameleons determine where prey is located to optimize time series stock market data. The ability to turn aids in 360-degree prey detection; the subsequent actions take place as follows: The original position of the chameleon is the focus of severity (i.e., the start); the rotation matrix is used to detect the location of the prey; The chameleon's position is updated using a rotation matrix at the gravitational midpoint, and then returned to its initial position to refine market stage optimization. $z_{s+1}^{j,i}$updates the position at the $s + 1$ iteration and $z_s^{j,i} + O_1\left(O_s^{j,i} - H_s^i\right)$ is global solution. $z_s^{j,i} + \mu(v^i - k^i)q_3 + k_a^i$denotes the positive control parameters, which are presented in Equation (13).

$$z_{s+1}^{j,i} = \begin{cases} z_s^{j,i} + O_1\left(O_s^{j,i} - H_s^i\right)q_2 + O_2\left(H_s^i - z_s^{j,i}\right)q_1 \\ z_s^{j,i} + \mu(v^i - k^i)q_3 + k_a^i dgn(rand - 0.5\,)q_1 < O_o \end{cases} q_1 \geq O_o \tag{13}$$

Hunting of prey: The WCS algorithm's prey behaviour hunting is essential for reducing the distribution of resource deployment for financial market building construction. It optimizes the model parameters, efficiently handling complex market data. More informed and effective decision-making in financial strategies can be enabled using Equation (14).

$$u_{s+1}^{j,i} = xu_s^{j,i} + d_1\left(H_s^i - z_s^{j,i}\right) + d_2\left(O_s^{j,i} - z_s^{j,i}\right)q_2 \tag{14}$$

Where, $u_{s+1}^{j,i}$ denotes the updated position of the agent at step $s + 1$in the swarm that represents the agent's new state after movement. d_1 *and* d_2 are weight factors that control the influence of the prey hunting behaviour. WCS algorithm modifies the location of agents within the swarm, optimizing their waves towards more favorable locations (prey or optimum points), allowing for better resource allocation in financial market strategies. Table 4 illustrates the hyperparameter scenarios for the WCS optimization method. Pseudocode 1 shows the proposed WCS-XGBoost method's working procedure.

Table 4: Hyperparameter scenarios for the optimization method

Hyperparameter	**Value**
Population Size	30
Max Iterations	100
Prey Behavior Weights (d1, d2)	0.1 – 0.9
Step Size Learning Rate (η)	0.05

Pseudocode 1: Weighted Chameleon Swarm-driven eXtreme Gradient Boosting (WCS-XGBoost)

Start

Step 1: *Load and preprocess time series data*

Input: *Financial time series dataset* $D = \{(w_j, z_j)\}$ *for* $j = 1$ *to* N

Apply $Z - score$ *normalization*

Apply PCA for dimensionality reduction

Step 3: *Train the final XGBoost model using optimized* θ_best

For each data point j:

Calculate the scoring function using Equation (5)

Step 5: *After optimization, extract the best parameters* θ *best* using Equation (6)

Update the gradient direction using Equation (7)

Solve optimization using Equation (8)

Compute the optimal step size using Equation (9)

Update parameter θ *using* Equation (10)

Step 4: *Begin optimization loop*

For iteration $s = 1$*to max_iter*:

For each agent j:

Evaluate current fitness: $k_j = -MSE(XGBoost_model(z^j))$

Eye rotation mechanism

Step 2: *Initialize WCS parameters*

Initialize swarm population $z_s^j \in \mathbb{R}s$ *for each chameleon agent j using* Equation (11)

For each agent j in the swarm:

Initialize position using Equation (12)

Initialize velocity v_i *randomly*

Evaluate fitness $k_j = -MSE(XGBoost_model(z^j))$

If rand $< q_1$:

Update position using Equation (13)

Else:

Prey hunting behavior

Update velocity using Equation (14)

Update position

$z^j = z^j + z_{s+1}^{j,i}$

Evaluate new fitness

$new_fitness = -MSE(XGBoost_model(z^j))$

If new fitness $> f_j$:

Accept the new position

Else:

Revert to the previous best position

End function

WCS-XGBoost utilizes a swarm intelligence optimization method to advance the accuracy of predictions for complex time series scenarios, making the model more suitable, robust, and well-performing in difficult financial forecasting tasks. WCS-XGBoost differs from other hybrid models combining boosting and metaheuristic--it types bio-inspired adaptation and bio-inspired adaptation with PCA dimensionality reduction--to develop an optimized, full hierarchical time series forecasting pipeline designed specifically for nonlinear financial data. The WCS-XGBoost method, which combines the XGBoost method with the WCS optimization strategy, is a powerful combination. This combination greatly improves the methods hyperparameter optimization process which allows it to escape local optimum and achieve better generalization. WCS's adaptive and exploratory exploitation strategy allows for easier, faster, and more stable convergence during training. As a result of WCS-XGBoost's great predictive accuracy by being generally better than methods that are conventional and other hybrid methods, this is especially true for highly complex nonlinear financial time series data. Furthermore, WCS-XGBoost is robust, consistent, and capable of handling various data distributions allowing practical forecasting situations to be undertaken by the model.

4 Result and discussion

The aim is to develop an effective time series prediction model for financial market data to improve stock data forecasting accuracy and support strategic decision-making under uncertainty. Table 5 shows the experimental setup, and Table 6 displays the ablation study on the impact of each component in the WCS-XGBoost framework for financial time series prediction.

Table 5: Experimental setup

Components	Specification
Operating System	Windows 10
Graphics Card (GPU)	NVIDIA GeForce RTX 3060
Processor (CPU)	Intel Core i7
Software	Python 3.x, pandas, NLTK, TensorFlow/Keras, Flask
Storage	500 GB SSD
RAM	16 GB DDR4

Table 6: Ablation study on the impact of each component in the WCS-XGBoost framework for financial time series prediction

Method Variant	Accuracy (%)	Precision (%)	Recall (%)
Data + Z-Score Normalization + PCA	88	87	86
Data + Z-Score Normalization + PCA + XGBoost	90	89	88
Data + Z-Score Normalization + PCA + XGBoost + CSO	93	92	91
Data + Z-Score Normalization + PCA + XGBoost + WCS	95	94	93
Data + Z-Score Normalization + PCA + XGBoost + WCS + Feature Tuning	**98.69**	**98**	**98**

ROC: The ROC curve graphically illustrates how well a classification model performs; it evaluates the discriminating model, and the TPR and FPR are compared across various thresholds. Figure 3 displays the outcome of the ROC curve.

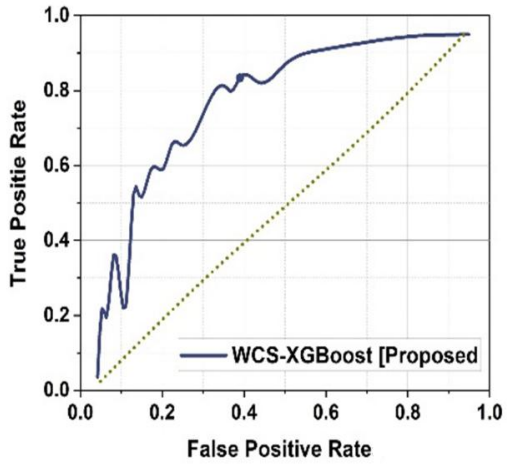


Figure 3: Graphical representation of ROC

The steady increase in method accuracy with more training epochs indicates that WCS-XGBoost effectively learns from the data and progressively improves its performance.

4.1 Stock market prediction

WCS-XGBoost predicts stocks into predefined categories using learned representations from historical data. It incorporates technical indicators, such as sentiment analysis, and fundamental metrics to enhance performance on financial time series. Figure 4 shows the Stock Market Prediction. This model's close tracking of the actual stock price indicates significant forecasts for stock trends, although slight deviations show the very nature of stock price predictions.

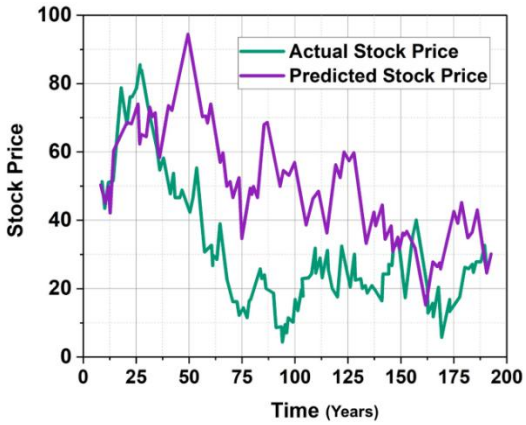


Figure 4: Stock market prediction

Comparative analysis : The evaluation of the effectiveness of the WCS-XGBoost method is associated with existing methods, including BiLSTM [20] and LR [19]. The comparative analysis uses multiple performance metrics, recall, accuracy, precision, RMSE, and MAPE to provide a comprehensive assessment of model efficacy. In this research, parameter choices for the WCS-XGBoost method were based on maximizing predictive performance across critical classification performance metrics (accuracy, precision, and recall) that are inherent to reliable stock market forecasting. The result is summarized in Table 7.

Accuracy is the degree to which a measurement or computation closely matches the real value of observations and corresponds to an accurate value.

Precision is the degree of precision in calculating the potential between measurements of the same element. **Recall** reflects a method's ability to correctly identify all relevant instances within a given dataset. Figure 5 illustrates the result of accuracy.

Table 7: Numerical outcome of metrics

Methods	Accuracy (%)	Precision (%)	Recall (%)
LR [19]	72.7	72.9	72.7
WCS-XGBoost [Proposed]	**98.69**	**98.02**	**98.00**

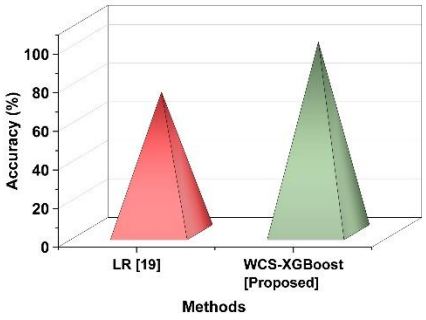


Figure 5: Result of accuracy

The proposed WCS-XGBoost method achieved an accuracy of 98.69%, significantly outperforming the existing LR approach, which attained only 72.7%. This substantial improvement underscores the efficiency and reliability of the proposed strategy. Figure 6 presents the comparative precision outcomes, further highlighting the method's predictive advantage.

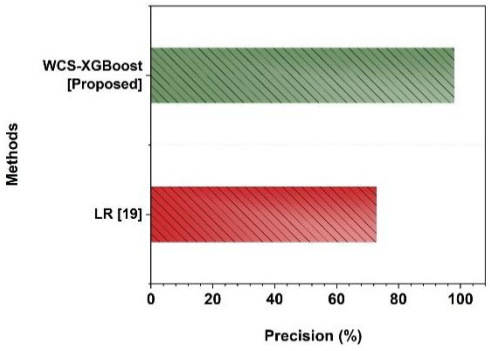


Figure 6: Precision comparison between the proposed method and the LR method [19]

Compared to the existing method, WCS-XGBoost achieved a precision 0f 98.02%, notably surpassing LR, which is 72.9%. This indicates that the proposed methods enhanced effectiveness in producing accurate predictions. Figure 7 illustrates the recall performance, further confirming its superiority.

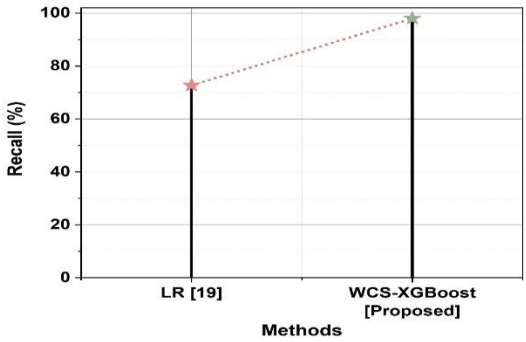


Figure 7: Performance outcome of recall

WCS-XGBoost achieved a significantly higher recall of 98.00% compared to 72.7% with LR, highlighting its superior ability to accurately identify relevant financial patterns and outperform conventional models.

RMSE calculates the mean magnitude of prediction errors in a model, applying a severe penalty for greater errors due to the squaring of the modification between predicted outcomes and true observations. **MAPE** refers to the different percentage deviation between forecasted results and observed values, offering relative indicators of forecast accuracy. Table 8 shows the numerical results of error metrics. Figure 8 shows the evaluation of (a) RMSE and (b) MAPE.

Table 8: Numerical results of error metrics

Methods	RMSE	MAPE
BiLSTM [20]	0.4605	0.795
WCS-XGBoost [Proposed]	**0.2312**	**0.321**

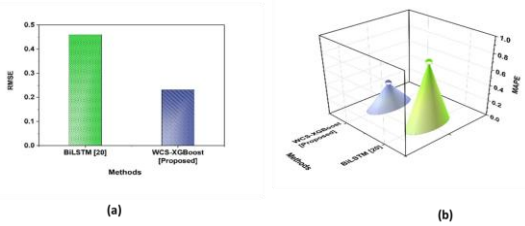


Figure 8: Evaluation of (a) RMSE and (b) MAPE

The BiLSTM performed moderately with an RMSE of 0.4605 and an MAPE of 0.795. The WCS-XGBoost model outperformed both, with much lower RMSE (0.2312) and MAPE (0.321). These findings demonstrate that WCS-XGBoost outperforms existing approaches as a robust and

adaptable framework for analyzing market trends and guiding financial decision-making.

4.2 Dataset comparison

In this research, the proposed WCS-XGBoost model is trained in both financial dataset such as Historical Stocks Price Dataset [21] and Stock Market Simulation Dataset. The goal of this comparative evaluation was to investigate the generalization capability of the proposed model in both synthetic and real market environments. The method's presentation was estimated using standard classification evaluation metrics: recall, accuracy, and precision. As shown in Table 9, the WCS-XGBoost model performed consistently well against both datasets, albeit with slightly better performance on the simulation dataset. This small, though meaningful difference, was a strong indication that the model is robust and sufficiently adaptable to learn complex financial patterns, irrespective of the complexity of the underlying data generation process. All of these results help validate the model's utility prospectively for real-world application, in financial forecasting, where a small performance improvement can have a considerable economic impact.

Table 9: Performance comparison of the proposed WCS-XGBoost model in simulation and real-world stock market datasets

Metrices	Stock Market Simulation Dataset [Proposed]	Stocks Historical Price Dataset [21]
Accuracy	98.69	98.62
Precision	98.02	97.58
Recall	98.00	97.93

The proposed WCS-XGBoost model has shown a better predictive performance when compared with the Stocks Historical Price Dataset [21], especially with the metrics used for evaluation. For example, on the stock market simulation dataset evaluated, accuracy was 98.69%, which is greater than the accuracy of the historical stock price model (98.62%). Likewise, precision (98.02%) and recall (98.00%) were compared with 97.58% and 97.93% respectively, from the model in [21]. Although the differences in accuracy, precision, and recall were only small in percentage, they are important for financial forecasting based on stock market predictions, since minor improvements can lead to more accurate decision-making. These results indicate the proposed method generalizes well between different datasets and consistently performs well in estimating stock market behavior. The use of PCA for dimensionality reduction and CSO for dynamic hyperparameter tuning enhanced the robustness and accuracy of the proposed framework when associated with the previous method [21] for treating and interpreting financial time series data with its nonlinear complexities.

4.3 Cross validation

Cross-validation is a type of model evaluation process that separates parts of the data into different subsets to examine the model's performance stability, generalization, and overfitting avoidance. This research conducted a 5-fold cross-validation and applied cross-validation to the results of WCS-XGBoost, ensuring that WCS-XGBoost produced similar consistent performance and was a reliable model across subsets of the training and unseen stock market dataset. Table 10 illustrates the 5-fold cross-validation results for WCS-XGBoost.

Table 10: Performance metrics of the WCS-XGBoost model using 5-fold cross-validation in financial time series data

Fold	Accuracy (%)	Precision (%)	Recall (%)	RMSE	MAPE
1	98.55	97.90	97.80	0.235	0.325
2	98.70	98.10	98.00	0.232	0.320
3	98.60	97.85	97.90	0.234	0.324
4	98.75	98.20	98.10	0.229	0.318
5	98.85	98.05	98.10	0.226	0.316
Average	**98.69**	**98.02**	**98.00**	**0.2312**	**0.321**

The performance robustness and generalization ability of the WCS-XGBoost model introduced in this research were evaluated using 5-fold cross-validation on the stock market simulation dataset. For each fold, the dataset was separated into training and validation subsets to mitigate overfitting and make fair estimates of performance. The dataset was separated into 5-folds, and evaluation metrics such as recall, accuracy, precision, RMSE, and MAPE were calculated. The WCS-XGBoost method achieved high performance across all folds. All metrics are presented in Table 10. WCS-XGBoost had an average accuracy of 98.69%, precision of 98.02%, and recall of 98.00% across all folds. All error metrics were very low for the lifespan of the prediction across folds, with RMSE of 0.2312 and MAPE of 0.321 providing a consistency

throughout different splits of the data, showing the WCS-XGBoost model is reliable and stable.

The research developed an effective time series prediction model for financial market data to enhance stock price forecasting. There were four problems with the study: it did not have integration with real-time data sources, there was limited applicability by context through industry, there was reduced explainability, and applied validation efforts across enterprise scales and contexts were minimal [8]. The model overfitted on very volatile datasets, had limited flexibility to react to changing market conditions, and lacked generalizability over a broad slice of financial regimes [9]. Clustering methods struggled with financial data dimensionality, lacked precision for temporal dependencies, and produced less interpretable clusters [10]. Had problems validating across different financial sectors, real-time dexterity, interpretability in black-box models, and large institutional scale (Bu). There was a lack of generalizability to non-chain markets, reduced interpretability, a lack of sufficient multi-source data integration, and increased computational requirements for dynamic updates (Jio). The WCS-XGBoost system was compared with conventional systems such as LR [19] and BiLSTM [20]. LR [19] has complexity in managing high-dimensional data and was susceptible to kernel regression assortment in the stock market. The systems were prone to over-fitting and needed considerable training data, limiting their relevance in complex market analysis methods. BiLSTM [20] overfits because of its complexity, particularly when trained on restricted, highly detailed real-world market data. In contrast, the WCS-XGBoost technique efficiently overcomes these restraints by capturing nonlinear connections, directing feature significance, and making more robust and precise stock market forecasts.

5 Conclusion

Financial time series prediction is vital to facilitate thoughtful investment decision-making, risk management, and economic forecasting. This research outlined a new hybrid model that synthesizes WCS with XGBoost to provide improved forecasting for complex and nonlinear financial data. The dataset consists of a variety of financial variables, including foreign exchange rates, stock indices, and prices on commodities. Data preprocessing involved Z-score normalization as a standard input feature normalization technique and PCA as a data dimensionality reduction technique to clean up and improve our input data. The use of WCS provided intelligent hyperparameter tuning for XGBoost through adaptive weights on the TD learning; in addition, an intelligent exploration-exploitation balance will lead to faster convergence. The WCS-XGBoost model outperformed the baseline model by a substantial margin, achieving an accuracy of 98.69%, a recall of 98.00%, a precision of 98.02%, MAPE (0.321, and RMSE (0.2312, suggesting that the model is robust across the financial domains. The ablation study also demonstrated the utility of all components, namely PCA and WCS, which work to improve model performance. The research did not consider real-time data handling, estimation of uncertainty, and integration of multiple data sources, and this restricts adjustment in highly volatile financial circumstances. In future research, it will incorporate real-time data streaming, uncertainty estimation, and data fusion from multiple different sources to further enhance the robustness of the prediction. The proposed system is expected to support a reliable, scalable, and data-driven approach to financial forecasting and offer some practical value to financial analysts, traders, and policymakers working in fast-moving financial markets. The WCS-XGBoost method will be validated with real-world financial datasets obtained for its robustness and generalizability beyond those that were achieved in simulated datasets for the purposes of this research.

References

[1] Zheng H, Wu J, Song R, Guo L, &Xu Z (2024). Predicting financial enterprise stocks and economic data trends using machine learning time series analysis. *Applied and Computational Engineering*, 87, 26-32. https://doi.org/10.54254/2755-2721/87/20241562

[2] Sharma R, & Mehta K (2024). Stock market predictions using deep learning: developments and future research directions. *Deep Learning Tools for Predicting Stock Market Movements*, 89-121. https://doi.org/10.1002/9781394214334.ch4

[3] Gong H (2024). An Enhanced Hybrid Model for financial market and economic analysis: a case study of the Nasdaq Index. *International Journal of System Assurance Engineering and Management*, 15(7), 3406-3423. https://doi.org/10.1007/s13198-024-02349-0

[4] Zareeihemat P, Mohamadi S, Valipour J, &Moravvej S.V (2025). Forecasting stock market volatility using housing market indicators: A reinforcement learning-based feature selection approach. *IEEE Access*. https://doi.org/10.1109/ACCESS.2025.3554224

[5] Yang C.H, Molefyane T, Lee B, Hsueh T, & Lin Y.D (2025). Improving Investment Forecasting: A Comparative Analysis of Machine Learning Models as Key GDP Indicators. *IEEE Access*. https://doi.org/10.1109/ACCESS.2025.3554641

[6] Amiri B, Haddadi A, &Mojdehi K.F. (2025). A Novel Hybrid GCN-LSTM Algorithm for Energy Stock Price Prediction: Leveraging Temporal Dynamics and Inter-Stock Relationships. *IEEE Access*. https://doi.org/10.1109/ACCESS.2025.3536889

[7] Pan H, Tang Y, & Wang G (2024). A Stock Index Futures Price Prediction Approach Based on the MULTI-GARCH-LSTM Mixed Model.

Mathematics, 12(11), 1677. https://doi.org/10.3390/math12111677

[8] Zhang S (2025). A Big Data-Driven Approach to Financial Analysis and Decision Support System Design. Informatica, 49(11). https://doi.org/10.31449/inf.v49i11.7065

[9] Bhambu A, Gao R, &Suganthan P.N. (2024). Recurrent ensemble random vector functional link neural network for financial time series forecasting. *Applied Soft Computing*, 161, 111759. https://doi.org/10.1016/j.asoc.2024.111759

[10] Cortés D.G, Onieva E, López I.P, Trinchera L, & Wu J (2024). Autoencoder-Enhanced Clustering: A Dimensionality Reduction Approach to Financial Time Series. *IEEE Access*, 12, 16999-17009. https://doi.org/10.1109/ACCESS.2024.3359413

[11] Durairaj D.M., & Mohan B.K. (2022). A convolutional neural network-based approach to financial time series prediction. *Neural Computing and Applications*, *34*(16), pp.13319-13337. https://doi.org/10.1007/s00521-022-07143-2

[12] Li Y, Chen L, Sun C, Liu G, Chen C, & Zhang Y (2024). Accurate stock price forecasting based on deep learning and hierarchical frequency decomposition. *IEEE Access*. https://doi.org/10.1109/ACCESS.2024.3384430

[13] Lazcano A, Herrera P.J, &Monge M (2023). A combined model based on recurrent neural networks and graph convolutional networks for financial time series forecasting. *Mathematics,* 11(1), 224. https://doi.org/10.3390/math11010224

[14] Albahli S, Irtaza A, Nazir T, Mehmood A, Alkhalifah A, &Albattah W (2022). A machine learning method for the prediction of the stock market using real-time Twitter data. *Electronics*, 11(20), 3414. https://doi.org/10.3390/electronics11203414

[15] Bu Y (2024). Fuzzy Decision Support System for Financial Planning and Management. Informatica, 48(21). https://doi.org/10.31449/inf.v48i21.6718

[16] Jiao Z (2024). Dynamic Financial Distress Prediction Using Combined LASSO and GBDT Algorithms. Informatica, 48(17), pp.139-152. https://doi.org/10.31449/inf.v48il7.6493

[17] González-Núñez E, Trejo LA, and Kampouridis M (2024). A comparative study of stock market forecasts based on a new machine learning model. Big Data and Cognitive Computing, 8(4), p.34. https://doi.org/10.3390/bdcc8040034

[18] Dataset: https://www.kaggle.com/datasets/samayashar/stock-market-simulation-dataset

[19] Mokhtari S, Yen K.K, & Liu J (2021). Effectiveness of artificial intelligence in stock market prediction based on machine learning. arXiv preprint arXiv:2107.01031. https://doi.org/10.5120/ijca2021921347

[20] Yang M, & Wang J (2022). Adaptability of financial time series prediction based on BiLSTM. *Procedia Computer Science*, 199, 18-25. https://doi.org/10.1016/j.procs.2022.01.003

[21] Dataset: https://www.kaggle.com/datasets/prodzar/stocks-historical-price-data

Acronyms	**Description**
TN	True Negative
TP	True Positive
FP	False Positive
FN	False Negative
LSTM	Long-term and short-term memory
RedRVFL	Recurrent Ensemble Deep Random Vector Functional Link
DL	Deep learning
CNN	Convolutional Neural Networks
GCN	Graph Convolutional Networks
BiLSTM	Bidirectional Long Short-Term Memory
ELM	Extreme Learning Machine
LLMs	large language models
TPR	true positive rate
FPR	false positive rate
ROC	Receiver Operating Characteristic

Optimization of Neural Network Architectures for Dynamic System Prediction Using LSTM-Based NODE and Water Wave Optimization

Jiating Fang
School of Mathematics and Statistics, Baicheng Normal University, Baicheng, Jilin, 137000, China
E-mail: fang2358@126.com

Keywords: Dynamic system, Variational Differential Equation (VDE), time-series data, network architecture, Long Short Term Neural Ordinary Differential Water Wave Optimization (LST-NODE-WWO).

Scientific computing faces challenges in accurately predicting complex dynamic systems due to initial conditions and non-stationary behavior, with rational neural networks often requiring human tuning. To overcome these limitations, this research proposes a Long Short-Term Neural Ordinary Differential Water Wave Optimization (LST-NODE-WWO) for dynamic system prediction. The goal is to automatically discover the optimal network architecture that adapts to the dynamics of each system. The proposed methodology includes five steps: collection of time-series data (e.g., position and velocity), preprocessing with z-score normalization for temporal consistency, and multi-resolution feature extraction using the discrete wavelet transform (DWT). The hybrid model integrates LSTM for capturing temporal dependencies, NODE as a variational differential equation (VDE) for continuous-time modeling, and WWO as a metaheuristic strategy for structure and hyperparameter optimization. This fusion enables the system to learn both physical consistency and long-range temporal patterns. Experimental results on a coupled damped harmonic oscillator dataset show that the LST-NODE-WWO outperforms baseline models. Experimental evaluations using a coupled damped harmonic oscillator dataset exhibit that the suggested model outperforms baseline approaches. The LST-NODE-WWO achieves a test MAE of 0.0914 and RMSE of 0.1298 for Oscillator 1, and a test MAE of 0.0213 and RMSE of 0.0301 for Oscillator 2, significantly improving prediction accuracy. Hence, LST-NODE-WWO offers a robust solution for modeling nonlinear, time-varying dynamic systems with minimal manual intervention.

Povzetek: Raziskava predstavi hibridni model LST-NODE-WWO, ki združuje LSTM za učenje časovnih odvisnosti, Neural ODE za neprekinjeno modeliranje dinamike in Water Wave Optimization za samodejno izbiro arhitekture. Na sklopu dušene sklopljene nihalne dvojice doseže najnižje MAE/RMSE in znatno preseže QLSTM ter QGRU pri napovedovanju nelinearnih dinamičnih sistemov.

1 Introduction

Dynamical systems form the basis for understanding and modeling the physical universe since they describe how systems evolve with respect to internal and external factors [1]. Modeling and predicting these systems, i.e., determining their state at a future time or times, is significant in all aspects of life: traffic control, finance (financial markets), ecology, and biological systems [2]. These systems often exhibit nonlinear complex behaviors that make accurate prediction difficult, particularly when one doesn't know the governing equation(s) or only partially understands them. Time series modeling, or forecasting future behavior from timely observed data, has emerged as a technique to model such systems [3]. In physics and engineering, time series data occur from systems driven by DEs. However, due to the complexity of the system, hidden interactions, or measurement capabilities, it is frequently impossible to know the exact governing equations [4].

In consequence, data-driven methods have shifted into the limelight, allowing researchers to model system behavior directly from observations without using the true system equations [5]. Various data-driven approaches have been proposed; specifically, NN and DL-based models have demonstrated an impressive capability to capture temporal patterns and linear/nonlinear relationships in time series data. RNNs, LSTM networks, and derivatives are often used for time-series forecasting, as they appropriately preserve historical dependencies [6, 7]. Requirements, however, necessitate careful network design adjustment, which can significantly affect generalization and prediction performance [8].

To approach the structural selection and model-building problem, some research has recently introduced some novel mathematical approaches by including ODEs and optimization in the dynamic neural network-based approach [9]. One developing area involves incorporating the use of ODE for neural networks, which allows them to learn dynamics for continuous time and optimize their internal structure with good theoretical reasoning [10].

This allows for a more realistic portrayal of a system's behavior, and even improves stability with long-term predictions. In addition to common real-world model issues like incomplete and noisy observations across dynamic systems [11], wireless communications present unique challenges that further complicate inference due to issues of signal interference and/or congestion that can lead to observation inconsistencies or entirely missing information. In conjunction with greater dimensionality and more interacting variables, the ability to forecast pure dynamics reliably is almost impossible using standard forms of modeling [12]. NN models in conjunction with very adaptively trained variational optimization provide an opportunity to deal with the missing piece, and to also account for more complicated dependencies and entropies [13].

The purpose of this research is to create a highly adaptable and automated neural network model to predict complex dynamic systems accurately. By using a combination of the LSTM network, NODE, and structurally optimizing the model with WWO, this research aims to provide an alternative to conventional neural networks that typically require tuning by a human operator and rarely work for non-stationary data or noisy data. The LST-NODE-WWO model learns temporal patterns, captures the continuous-time behavior in the system, and can adjust its structure dynamically, resulting in a robust, generalizable methodology that satisfies a push-button solution for real-world applications for predicting dynamic systems.

The organization of the research is as follows: Part 2 gives the related works, Part 3 explains the methodology, Part 4 gives the results and discussion, and Part 5 concludes the research.

2 Related works

Recent advancements in neural networks have significantly improved the modeling and prediction of nonlinear dynamical systems. The research [14] compared RNN variants for predicting pore-water pressure in slope stability analysis, with LSTM and GRU showing higher robustness. To address uncertainty in runoff forecasting, a hybrid SVM-based model was introduced [15]. The research [16] connected dynamical system theory with neural training via KOT. The analysis [17] investigated adaptive behavior and predictive information in agent-environment systems. In physical forecasting, a DL-physics hybrid called PhICNet was proposed [18]. A transformer-based approach using continuous attention for spatiotemporal prediction was developed in [19], while SDPA-Net with novel dynamic attention mechanisms for sales forecasting was presented in [20]. LSTM-based data assimilation was improved in [21], and forecasting models were compared under pandemic conditions in [22]. Additionally, research [23] introduced a fuzzy recurrent neural model optimized by swarm intelligence, and improved learning was demonstrated by injecting partial physics knowledge into DNNs [24]. Table 1 illustrates the Existing studies with key methods and findings.

Table 1: Summary of related works with key methods and results

Ref	Model	Application Domain	Evaluation Metrics & Results	Key Contribution
Wei et al., [14]	RNN, LSTM, GRU vs MLP	PWP (Slope Stability)	LSTM & GRU achieved better robustness; MLP performed acceptably; standard RNN struggled with delay	LSTM & GRU gave more stable forecasts under lagged inputs
Feng et al., [15]	Hybrid SVM	Monthly Runoff Forecasting	Lower RMSE vs. conventional methods; exact values not stated	Handles nonstationary signals via decomposition
Hu et al., [16]	Non-deterministic WSO+ BST	Dynamic Economic Dispatch	RMSE (Clear): 2.68% vs 5.43% (LS-SVM)	Enhanced wind power forecasting accuracy across weather conditions and reduced scheduling time using a hybrid evolutionary optimization approach
Candadai and Izquierdo [17]	Random Dynamical System Model	Predictive Information in Agents	Conceptual; no metrics given	Shows that the prediction size varies with environmental dynamics

Saha et al., [18]	PhICNet (Conv-RNN + Physics)	Spatiotemporal Forecasting w/ Hidden Source	Accurate long-term prediction in spatiotemporal fields; quantitative metrics not specified	Integrates physics & deep learning for source identification
You et al., [19]	STNN (Spatiotemporal Transformer)	Short-Term Time Series Forecasting	Outperforms baselines; reconstructs phase space of systems; detailed metrics not given	Uses continuous attention for accurate forecasting
Li et al., [20]	SDPA-Net	Sales Forecasting	↑46% CORR, ↓39.5% RRSE, ↓41.5% RMAE over baseline	Combines change-sensitive loss, hierarchical attention, and regression
Cheng and Qiu [21]	LSTM for Covariance Learning	Data Assimilation in Dynamics	Better assimilation accuracy & lower computational cost than existing methods	Learns observational covariance directly from data
Oukhouya et al., [22]	ARDL, LSTM, XGBOOST	MASI (Financial Index Forecasting)	ARDL: MAPE = 26.7%, Processing time = 1 sec	ARDL captured trend-seasonality well in the pandemic context
Nasiri and Ebadzadeh [23]	MFRFNN + PSO	Chaotic Time Series + Real Datasets	RMSE significantly reduced vs. conventional NN	Combines fuzzy logic and recurrent feedback for precision
Robinson et al., [24]	Physics-Guided DNN	Nonlinear Dynamic Systems (Lorenz, etc.)	Enhanced convergence, reduced uncertainty, ↑ and accuracy with partial physics input	Improves learning by integrating physical priors

1.1 Problem statement

Dynamic systems, characterized by nonlinear, time-dependent behavior, are fundamental across domains such as hydrology, climate modeling, structural analysis, and financial forecasting. Traditional machine learning approaches often fail to capture temporal dependencies and nonlinearities effectively, prompting the rise of neural network-based methods. It has explored various architectures, including LSTM, GRU, transformer models, and hybrid physics-informed frameworks. For instance, LSTM and GRU have demonstrated robustness in slope stability analysis [14], while hybrid SVM models addressed uncertainty in runoff prediction [15]. Others, like PhICNet [18] and physics-guided DNNs [24], blend physical laws with deep learning for improved accuracy and interpretability. To overcome these challenges, the proposed LST-NODE-WWO model integrates LSTM's temporal learning, NODE's continuous-time dynamics, and WWO's global optimization. This hybrid approach adaptively discovers optimal architectures, enhances long-term prediction accuracy, and embeds dynamic system behavior, leading to improved generalization, interpretability, and robustness in complex nonlinear dynamic system forecasting.

3 Methodology

This methodology combines time-series data acquisition, z-score normalization, and DWT for effective preprocessing and feature extraction. It integrates LSTM for capturing temporal dependencies, Neural-ODEs for continuous dynamic modeling, and WWO for tuning architecture and parameters. Together, these components form a hybrid LST-NODE-WWO model for accurate nonlinear dynamic system prediction. Figure 1 gives the overall flow of the methodology.

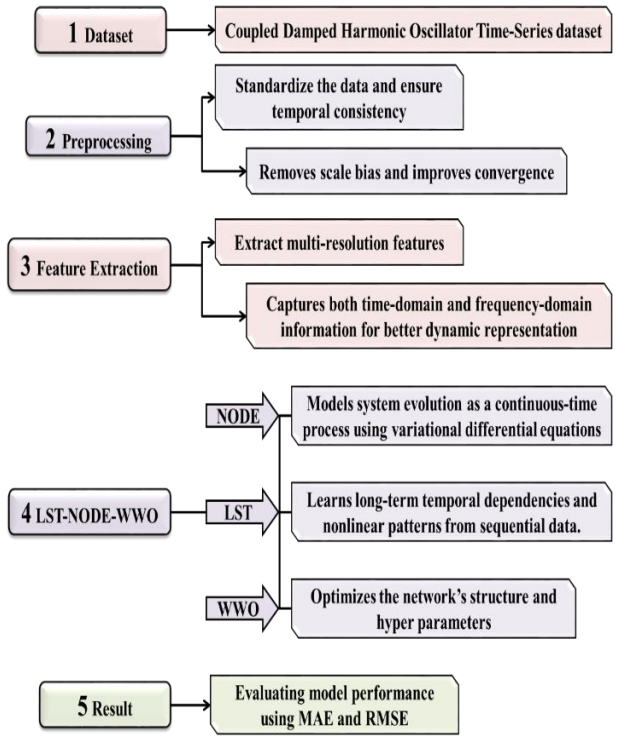


Figure 1: Overall flow of methodology

1.2 Dataset

The Coupled Damped Harmonic Oscillator Time-Series dataset represents the multivariate dynamics of two interacting masses connected through springs and dampers. Captured over 2763-time steps at uniform intervals of 0.01 seconds, the dataset includes detailed measurements of position (x1, x2), velocity (v1, v2), and acceleration (a1, a2) for both masses, offering a comprehensive view of their coupled motion. Each row corresponds to a specific point in time, and the system_id column distinguishes between different configurations or system simulations. This dataset is ideal for studying classical mechanical systems, analyzing second-order dynamic behavior, and applying multivariate time-series forecasting or physics-informed machine learning models. Source:
https://www.kaggle.com/datasets/programmer3/coupled-damped-harmonic-oscillator-time-series

1.3 Pre-processing using z-score normalization

The normalization is applied to standardize time series input data, ensuring consistency across dynamic systems and improving model adaptability. It transforms feature values using their mean μ and standard deviation σ (Equation 1). This maps the data to a zero-centered scale, reducing the impact of outliers and improving the strength and accuracy of the proposed model during training and optimization.

$$u' = \frac{u-\mu}{\sigma} \quad (1)$$

Ensures that each feature has a mean of 0 and a standard deviation of 1, improving convergence during training. This normalization is particularly effective in handling the varying magnitudes of physical quantities (e.g., velocity vs. acceleration) and reducing model sensitivity to scale disparities.

1.4 Feature extraction using DWT

DWT is used to extract multi-resolution features from time-series data, offering better time-frequency localization than the STFT. In contrast to STFT, DWT offers high-frequency resolution for low frequencies and high time resolution for high frequencies, which is comparable to human hearing. The DWT uses a pair of filters; $h[]$ presents the high-pass filter to extract detail components. $g[]$ denotes the low-pass filter to extract approximation components. The decomposition is performed using Equations (2 and 3).

$$x_{high}[l] = \sum_m y[m]h[2l-m] \quad (2)$$

$$x_{low}[l] = \sum_m y[m]g[2l-m] \quad (3)$$

Where$x_{high}[l]$, and$x_{low}[l]$ are the outputs of high-pass (h) and low-pass (g) filters, respectively. $y[m]$ denotes the original time series input signal. $l$representsoutput sample index after down-sampling. m presents the input sample index. The process decreases dimensionality while retaining important temporal and frequency characteristics. DWT provides the LST-NODE-WWO model with the advantage of better learning dynamic behavior since it can capture short transients and longer trends.

1.5 Long short term neural ordinary differential water wave optimization (LST-NODE-WWO) for dynamic system prediction

The LST-NODE-WWO framework that was proposed combines the memory-oriented temporal learning of LSTMs with continuous modeling of physical consistency using NODE and the metaheuristics of WWO. The hybrid model provided adaptable architecture and hyperparameter selection with dynamically captured complex functions of time-varying behavior. As a result, it made accurate and generalizable predictions for nonlinear dynamic systems without user tuning, linking data-driven learning with the understanding of physical systems. WWO-LSTM-NODE Integration Strategy, detailing how WWO optimizes the LSTM-NODE model. After

constructing the initial architecture, WWO explores hyperparameter configurations such as LSTM units, NODE depth, learning rate, and batch size. Using Propagation, Breaking, and Refraction, WWO generates candidate solutions, evaluates them by training the model, and selects the best based on validation performance. The final model is retrained using the optimal configuration, ensuring fully automated structure and training parameter optimization without manual tuning.

1.5.1 NODE

NODE is a type of DL model that represents system dynamics as a continuous-time process. NODE is similar to using traditional deep networks with distinct layers, but instead, NODE describes the transition of state in the system through DEs. The basic formulation is in Equation (4).

$$\dot{y}(t) = F\,(y(t), s) \tag{4}$$

Where $y(t)$ is the system state at time t; $\dot{y}(t)$ is the time derivative of $y(t)$; F is a neural network parameterized by time-dependent weights; θ represents the learnable parameters of the model; t is continuous time. Given an initial condition$y\,(0) = y_0$, an ODE solver integrates the equation to predict future system states y (t_k). This framework uses variational DEs to approximate the underlying physical laws of dynamic systems. Incorporating variational principles helps neural-ODEs generate smooth, accurate, and physics-consistent trajectories, making them especially suitable for modeling nonlinear, time-varying dynamic systems.

1.5.2 LSTM

LSTM networks improve upon RNNs as a sophisticated architecture that is capable of capturing information from data that is sequential with long-range temporal dependencies. The results of LSTMs can be used to predict within a dynamic system, capture periodic time-based behavior, and retain relevant information about the past over time. This makes LSTM highly capable in scenarios where standard RNNs often fall short, as follows in Equations (5 to 9).

$$f_s = \sigma(V_f \cdot [h_{s-1}, y_s] + a_f \tag{5}$$

$$j_s = \sigma(V_j \cdot [h_{s-1}, y_s] + a_j (6)$$

$$b_s = \tanh\,(V_b \cdot [h_{s-1}, y_s] + a_b \tag{7}$$

$$o_s = \sigma(V_o \cdot [h_{s-1}, y_s] + a_o \tag{8}$$

$$h_s = o_s \cdot \tanh\,(b_s) \tag{9}$$

Forget gate f_s determines what information from the previous state to discard. j_s input gate determines what new information to store. Candidate state b_s proposed new content to add to the cell state. The output gate o_s determines what part of the memory cell to output. Final hidden state h_s output of the LSTM for the current time step.

1.5.3 WWO

In the proposed LST-NODE-WWO model, WWO serves as a metaheuristic algorithm for automatically optimizing neural network structure and hyperparameters, eliminating the need for manual tuning. Inspired by the propagation of water waves, WWO represents each solution as a wave, where its position in the search space reflects a candidate solution, and its fitness indicates prediction accuracy.

Propagation adjusts each dimension c of the solution Y by a random displacement to explore the search space. The updated solution is given by Equation (10).

$$Y_c' = Y_c + rand\,(-1, 1) \cdot \lambda M_c \tag{10}$$

Where λ is the wave's **wavelength,**M_c is the range of the cth dimension, and rand(-1,1) is a uniform random number. Y_c: The value of the solution Y in the **dimension** (i.e., one specific parameter or component). Y_c': The **new (updated) value** after the operation is applied.

Breaking introduces local search behavior around promising solutions by generating variations using Gaussian perturbation (Equation 11). Where Gaussian (0,1) is a normally distributed random number (mean 0, standard deviation 1), and β is the breaking coefficient.

$$Y_c' = Y_c + Gaussian\,(0,1) \cdot \beta M_c \tag{11}$$

Refraction avoids stagnation by repositioning poor solutions based on the current best (Equation 12). Where $Y_{best,c}$ is the best solution's value in dimension c.

$$Y_c' = Gaussian\left(\frac{Y_{best,c} + Y_c}{2}, \frac{Y_{best,c} - Y_c}{2}\right) \tag{12}$$

After each operation, the wavelength is updated to balance exploration and exploitation (Equation 13).

$$\lambda' = \lambda \frac{f(Y)}{f(Y')} \tag{13}$$

The WWO algorithm is clearly defined and integrated with the LSTM-NODE model to automate hyperparameter tuning. Each solution Y is treated as a wave representing hyperparameters such as LSTM units, NODE depth, and learning rate. Propagation adjusts the dimension c as $Y_c' =$

$Y_c + rand\ (-1,1)\ \cdot \lambda M_c$, where λ is the wavelength and M_c is the range of dimensions c. Breaking uses Gaussian noise: $Y_c' = Y_c + Gaussian\ (0,1) \cdot \beta M_c$, with β as the breaking coefficient. Refraction updates poor solutions via $Y_c' = Gaussian\ \left(\frac{Y_{best,c} + Y_c}{2}, \frac{Y_{best,c} - Y_c}{2}\right)$, where $Y_{best,c}$ is the best solution. Wavelength is updated with $\lambda' = \lambda \frac{f(Y)}{f(Y')}$, ensuring adaptive balance between exploration and exploitation in the optimization process.

Here, $f(Y)$ is the current solution fitness, and ϵ is the small constant to avoid division by zero. By embedding WWO into the LST-NODE-WWO framework, the model dynamically learns the most effective neural configurations, improving generalization and prediction accuracy for time-dependent, nonlinear dynamic systems. Algorithm 1 gives the pseudocode of the proposed LST-NODE-WWO method. Table 2 presents the hyperparameters of the proposed method.

Algorithm 1: LST-NODE-WWO

Input: Time-series dataset D (e.g., x1, x2, v1, v2, a1, a2)

Output: Optimized LST-NODE model for dynamic system prediction

1. Build Initial LST-NODE Model:

a. Define an LSTM layer to capture long-range dependencies

b. Integrate Neural ODE layer for continuous-time state evolution

c. Combine into a single LST-NODE architecture

2. Train the Initial Model:

a. Use time-series data to train the LST-NODE model

b. Evaluate performance using MAE/RMSE on validation data

3. Optimize Model with Water Wave Optimization (WWO):

a. Initialize a population of waves (each representing different hyperparameters)

b. For each wave Y:

i. Propagation: Y' = Y + rand(-1,1) * λ * Mc

ii. Breaking (if promising): Y' = Y + Gaussian(0,1) * β * Mc

iii. Refraction (if stagnant): Y' = Gaussian((Y_best + Y)/2, (Y_best - Y)/2)

iv. Update wavelength: λ' = λ * (f(Y) / f(Y'))

v. Evaluate fitness of each wave (i.e., retrain + validate)

c. Select best-performing wave configuration Y_best

4. Retrain Final Model:

a. Use the best parameters from Y_best to reconfigure LST-NODE

b. Retrain using full training data

5. Prediction:

a. Use the final trained model to forecast future dynamic states

Return: Final trained LST-NODE model with optimized structure

Table 2: The proposed LST-NODE-WWO model hyperparameters

Component	Parameter	Value
LSTM	Number of units	64
	Activation	tanh
	Optimizer	Adam
	Learning rate	0.001
NODE	ODE solver	Runge-Kutta (RK45)
	Time steps	0.01
	Tolerance	1e-5
WWO	Population size	30
	Number of iterations	100
	Breaking coefficient (β)	0.3
	Wavelength λ range	[0.1, 0.5]
	Fitness function	MAE on validation set

2 Result and discussion

The output behavior of the proposed LST-NODE-WWO model, including velocity, position, and acceleration plots, followed by quantitative performance comparisons, was presented in this section. Results demonstrate the model's superior ability to capture and predict nonlinear dynamics with high accuracy and stability. Table 3 shows the experimental setup.

Table 3: Experimental setup

Component	**Specification**
Software	
Programming Language	Python 3.9
Deep Learning Framework	TensorFlow 2.11
Libraries Used	NumPy, SciPy, PyWavelets, Matplotlib, Scikit-learn
Hardware	
CPU	Intel Core i7-10750H
RAM	16 GB
GPU	NVIDIA GTX 1660 Ti (6 GB)
Operating System	Windows 10 / Ubuntu 22.0

2.1 Output phase

The velocities v1 and v2 of the two coupled masses are shown in Figure 2. Each signal has damped oscillatory motion, slowly lessening in amplitude due to loss of energy. The curves show phase separation that indicates the coupled interaction and the effectiveness of the proposed model to learn dynamic patterns.

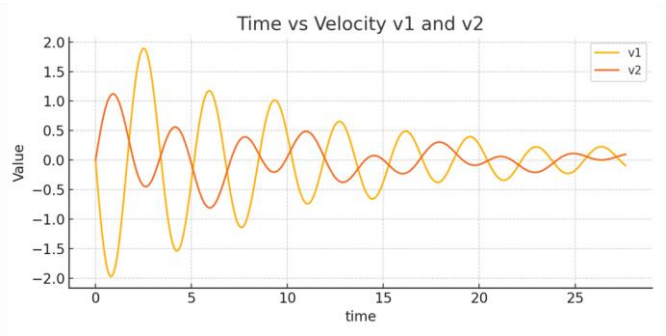


Figure 2:Velocity-time plot of oscillators v1 and v2, showing phase-shifted damped motion

Figure 3displays the position data x1and x2over time for both oscillators. The motion starts with high amplitude and converges as damping increases. The proposed model accurately captures the out-of-phase yet coupled behavior of the masses, demonstrating the system's energy dissipation and synchronization over time.

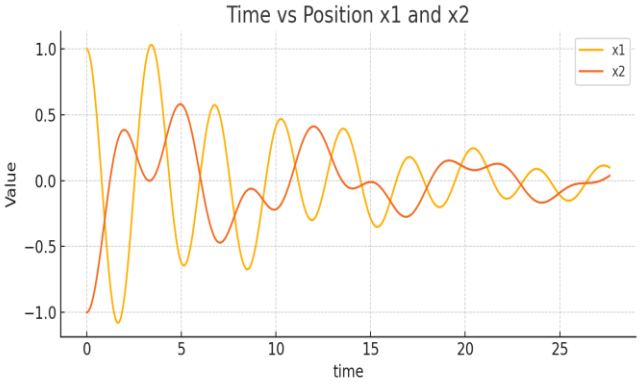


Figure 3: Position-time plot for x1 and x2, illustrating coupled damped oscillations.

Figures 4 and 5 show acceleration values a1 and a2, respectively. High-frequency oscillations are observed initially, gradually decaying as the system stabilizes. These results verify the model's capability to capture second-order dynamics and transient responses of the coupled system under damping and spring force interactions.

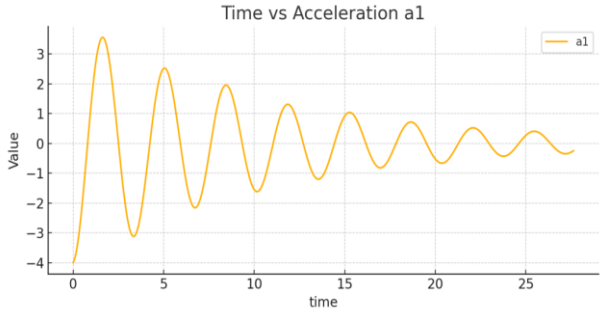


Figure 4: Acceleration-time plots for (a) a1, highlighting damped transient behavior.

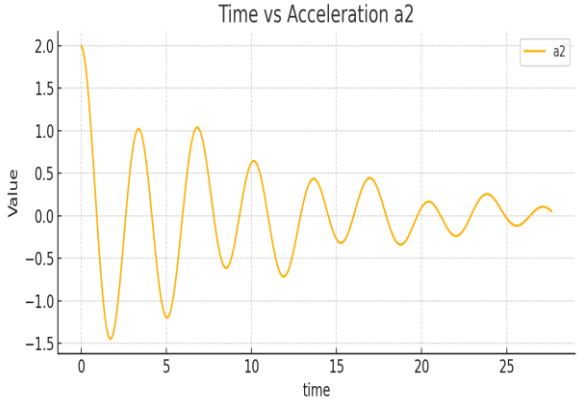


Figure 5: Acceleration-time plots for a2, highlighting damped transient behavior.

2.2 Comparison phase

The Two Coupled Damped Harmonic Oscillators system is a classical dynamic model used to simulate the interaction between two masses connected by springs and subjected to damping forces. It is widely used to test the effectiveness of time-series prediction models due to its nonlinear, oscillatory, and time-dependent behavior.

To assess the predictive accuracy of the proposed LST-NODE-WWO model, it was compared against two quantum-inspired baselines, QLSTM and QGRU [25], using MAE and RMSE on both training and testing sets.

Table 4: Error comparisons between Oscillator 1's train and test

Model	MAE		RMSE	
	Train	**Test**	**Train**	**Test**
QLSTM [25]	0.5284	0.4763	0.6987	0.5374
QGRU [25]	0.2585	0.2411	0.3587	0.2701
LST-NODE-WWO [Proposed]	0.1123	0.0914	0.1539	0.1298

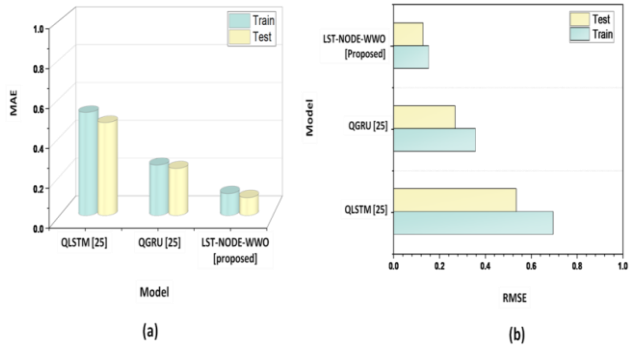


Figure 6: Performance comparison of models with (a) MAE and (b) RMSE

Table 5: Error comparisons between Oscillator 2's train and test

Model	MAE		RMSE	
	Train	**Test**	**Train**	**Test**
QLSTM [25]	0.1244	0.2591	0.1607	0.2885
QGRU [25]	0.0794	0.0482	0.0949	0.0602
LST-NODE-WWO [Proposed]	0.0276	0.0213	0.0365	0.0301

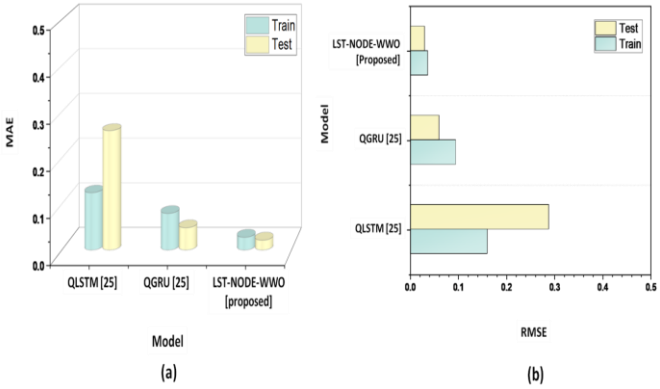


Figure 7: Performance comparison of models with (a) MAE and (b) RMSE

As shown in Tables 4 and 5, the proposed LST-NODE-WWO model outperforms both QLSTM and QGRU across all metrics for both oscillators. For Oscillator 1 (Figure 6 (a, b)), the LST-NODE-WWO achieves the lowest MAE and RMSE values on both training and testing sets, indicating high learning efficiency and generalization. Similarly, when looking at Oscillator 2 (Figure 7 (a, b)), the proposed model provides a significant reduction in prediction error, especially for the test phase, demonstrating its generalizability when forecasting unseen dynamic patterns.

Computational efficiency

Table 6 compares QLSTM, QGRU, and the proposed LST-NODE-WWO in terms of computational efficiency. LST-NODE-WWO requires the highest training time (42s), inference time (7.5ms), memory (3.2GB), and model size (4.7MB), reflecting its complex hybrid design. In contrast, QGRU is the most efficient. The trade-off suggests LST-NODE-WWO prioritizes accuracy over lightweight performance.

Table 6: Comparison of computational metrics across baseline and proposed prediction models.

Metric	**QLSTM [25]**	**QGRU [25]**	**LST-NODE-WWO (Proposed)**
Training Time	29 s	24 s	42 s
Total Training Time	24 min	20 min	35 min
Inference Time	5.1 ms	4.3 ms	7.5 ms
Memory Usage	2.1 GB	1.8 GB	3.2 GB
Model Size	3.2 MB	2.8 MB	4.7 MB

Existing methods, such as QLSTM [25] and QGRU [25] struggle to capture continuous dynamics and often require tedious manual tuning of hyperparameters, which limits their generalizability and accuracy for complex dynamical systems. The suggested LST-NODE-WWO is a hybrid approach that employs continuous time modelling, increases the time efficiency, and removes the need for explicit hyperparameter tuning, allowing for better adaptations to the wide range of possible nonlinear, time-variant behaviours that are naturally endowed to this domain, along with resulting in higher performance on dynamic system forecasting challenges. Overall, the findings support the efficacy of employing LSTM for temporal learning, NODE for continuous time modelling, and WWO for structural optimization. The proposed model functions quite well through combinations of these frameworks for accurate and stability-based forecasting of complex dynamic systems, demonstrating the plausibility of using a generalizable model to help quickly generate solutions to real-world time-series forecasting tasks.

3 Conclusion

The hybrid neural architecture called LST-NODE-WWO introduced a framework for dynamic system prediction that incorporates LSTM, NODE, and WWO. The LST-NODE-WWO model integrates both the temporal dependencies and continuous-time dynamics of the system while optimizing its architecture automatically. Applying LST-NODE-WWO to a coupled damped harmonic oscillator resulted in both smaller MAE (0.0213) and RMSE (0.0301) for Oscillator 2 relative to standard modelling approaches, demonstrating LST-NODE-WWO can be both accurate and robust. The model does not require any manual tuning and can adapt to the non-linear, time-varying behavior of real-world systems. The LST-NODE-WWO architecture, with its generalizability, could enable a variety of dynamic prediction problems across many fields in engineering and science. Furthermore, we clarified the **interaction mechanism** between LSTM, NODE, and WWO, showcasing how the model combines data-driven learning, differential modeling, and adaptive optimization. This comprehensive integration makes the proposed architecture suitable for modeling **nonlinear, time-varying systems** and opens avenues for applications in **real-time or high-dimensional dynamic environments**. Future work will focus on expanding the framework to handle more complex systems, leveraging hardware-accelerated solvers, and improving real-time deployment scalability.

Appendix

Abbreviation	Full form
NODE	Neural Ordinary Differential Equations
VDE	variational differential equation
WWO	Water Wave Optimization
DL	deep learning
DEs	differential equation
NN	Neural networks
RNN	recurrent NN
DNN	Deep neural networks
PWP	pore-water pressure
LSTM	long short-term memory
ANN	artificial NN
MLP	multi-layer perceptron
GRU	gated recurrent unit
KOT	Koopman Operator Theory
STNN	spatiotemporal transformer NN
SDPA	Spatiotemporal Dynamic Pattern Acquisition Mechanism
MASI	Moroccan All Shares Index
MAPE	Mean Absolute Percentage Error
MFRFNN	multi-functional recurrent fuzzy NN
MAE	Mean Absolute Error
RMSE	Root Mean Squared Error
QLSTM	Quantum LSTM
QGRU	Quantum GRU
PhICNet	Physics-Incorporated Convolutional Recurrent Neural Networks
ARDL	AutoRegressive Distributed Lag
XGBOOST	Extreme Gradient Boosting
STFT	short-time Fourier Transform
DWT	discrete wavelet transform
WSO	Whale Swarm Optimization
BST	Bee Species Transition

References

[1] PanahiS,& Lai YC (2024). Adaptable reservoir computing: A paradigm for model-free data-driven prediction of critical transitions in nonlinear dynamical systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science, 34*(5). https://doi.org/10.1063/5.0200898

[2] Yang, C., & Li, B. (2025). SDN-DRLTE Algorithm Based on DRL in Computer Network Traffic Control. *Informatica*, *49*(13). https://doi.org/10.31449/inf.v49i13.7576

[3] Uribarri G &Mindlin GB (2022). Dynamical time series embeddings in recurrent neural networks. *Chaos, Solitons& Fractals, 154*, 111612. https://doi.org/10.1016/j.chaos.2021.111612

[4] Chen N,&Majda AJ (2020). Predicting observed and hidden extreme events in complex nonlinear dynamical systems with partial observations and short training time series. *Chaos: An Interdisciplinary Journal of Nonlinear Science, 30*(3). https://doi.org/10.1063/1.5122199

[5] Bertozzi O, Chamorro HR, Gomez-Diaz EO, Chong MS, & Ahmed S (2024). Application of data-driven methods in power systems analysis and control. *IET Energy Systems Integration, 6*(3), 197–212. https://doi.org/10.1049/esi2.12122

[6] Song X, Liu Y, Xue L, Wang J, Zhang J, Wang J, Jiang L, & Cheng Z (2020). Time-series well performance prediction based on a Long Short-Term Memory (LSTM) neural network model. *Journal of Petroleum Science and Engineering, 186*, 106682. https://doi.org/10.1016/j.petrol.2019.106682

[7] Islam BU,& Ahmed SF (2022). Short-term electrical load demand forecasting based on LSTM and RNN deep neural networks. *Mathematical Problems in Engineering, 2022*(1), 2316474. https://doi.org/10.1155/2022/2316474

[8] Bandara K, Bergmeir C, &Smyl S (2020). Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *Expert Systems with Applications, 140*, 112896. https://doi.org/10.1016/j.eswa.2019.112896

[9] Yang, Y., & Li, H. (2025). Neural ordinary differential equations for robust parameter estimation in dynamic systems with physical priors. *Applied Soft Computing, 169*, 112649. https://doi.org/10.1016/j.asoc.2024.112649

[10] ZimmeringB,&Niggemann O (2024). Integrating continuous-time neural networks in engineering: Bridging machine learning and dynamical system modeling. *Machine Learning for Cyber Physical Systems*, 55.. DOI:10.24405/15313

[11] Liu, C., Quilliot, A., Toussaint, H., & Feillet, D. (2025). Dynamic Routing for Large-Scale Mobility On-Demand Transportation Systems. *Informatica*, *49*(1). https://doi.org/10.59615/ijie.3.1.55

[12] Cheng Y, Li G, Zhou X, & Ye S (2025). Research on time series forecasting models based on hybrid attention mechanism and graph neural networks. *Informatica, 49*(21). https://doi.org/10.31449/inf.v49i21.7580

[13] Legaard C, Schranz T, Schweiger G, Drgoňa J, Falay B, Gomes C, Iosifidis A, Abkar, M, & Larsen P (2023). Constructing neural network-based models for simulating dynamical systems. *ACM Computing Surveys, 55*(11), 1–34. https://doi.org/10.1145/3567591

[14] Wei X, Zhang L, Yang HQ, Zhang L, & Yao YP (2021). Machine learning for pore-water pressure time-series prediction: Application of recurrent neural networks. *Geoscience Frontiers, 12*(1), 453–467. https://doi.org/10.1016/j.gsf.2020.04.011

[15] Feng Z K, Niu W J, Tang Z Y, Jiang ZQ, Xu Y, Liu Y, & Zhang HR (2020). Monthly runoff time series prediction by variational mode decomposition and support vector machine based on quantum-behaved particle swarm optimization. *Journal of Hydrology, 583*, 124627. https://doi.org/10.1016/j.jhydrol.2020.124627

[16] Hu, Y., Yang, X., Chen, B., Gu, G., & Pan, L. (2025). Wind Power Prediction and Dynamic Economic Dispatch Strategy Optimization Based on BST-RGOA and NDO-WOA. *Informatica*, *49*(6). https://doi.org/10.31449/inf.v49i6.6940

[17] CandadaiM,&Izquierdo E J (2020). Sources of predictive information in dynamical neural networks. *Scientific Reports, 10*(1), 16901. https://doi.org/10.1038/s41598-020-73380-x

[18] Saha P, Dash S, &Mukhopadhyay S (2021). Physics-incorporated convolutional recurrent neural networks for source identification and forecasting of dynamical systems. *Neural Networks, 144*, 359–371. http://dx.doi.org/10.48550/arXiv.2004.06243

[19] You, Y, Zhang L, Tao P, Liu S, & Chen, L (2022). Spatiotemporal transformer neural network for time-series forecasting. *Entropy, 24*(11), 1651. https://doi.org/10.3390/e24111651

[20] Li D, Lin K, Li X, Liao J, Du R, Chen D, & Madden A (2022). Improved sales time series predictions using deep neural networks with spatiotemporal dynamic pattern acquisition mechanism. *Information Processing & Management, 59*(4), 102987. https://doi.org/10.1016/j.ipm.2022.102987

[21] Cheng S,&Qiu M (2022). Observation error covariance specification in dynamical systems for data assimilation using recurrent neural networks. *Neural Computing and Applications, 34*(16), 13149–13167. https://doi.org/10.1007/s00521-021-06739-4

[22] Oukhouya MH, Angour N, Aboutabit N, &Hafidi I (2025). Comparative analysis of ARDL, LSTM, and XGBoost models for forecasting the Moroccan stock market during the COVID-19 pandemic. *Informatica, 49*(14). https://doi.org/10.31449/inf.v49i14.5751

[23] NasiriH,&Ebadzadeh MM. (2022). MFRFNN: Multi-functional recurrent fuzzy neural network for chaotic time series prediction. *Neurocomputing, 507*, 292–310. https://doi.org/10.1016/j.neucom.2022.08.032

[24] Robinson H, Pawar S, Rasheed A, & San O (2022). Physics guided neural networks for modelling of non-linear dynamics. *Neural*

Networks, *154*, 333–345. https://doi.org/10.1016/j.neunet.2022.07.023

[25] Chen Y,&Khaliq A (2024). Quantum recurrent neural networks: Predicting the dynamics of oscillatory and chaotic systems. *Algorithms*, *17*(4), 163. https://doi.org/10.3390/a17040163

Application of Cycle Generative Adversarial Networks for Unpaired Image Style Transfer in Product Packaging Design

Jie Liu
Communication University of Shanxi, Jinzhong,030619, Shanxi, China
E-mail: jiel88580@gmail.com

Keywords: cycle generative adversarial network (CycleGAN), industrial design, visual communication, product packing design direction

Received: July 8, 2025

The rapid advancement and extensive application of AI technology in the past few years have also spurred rapid evolution in deep learning. Image recoloring, detection, recognition, and creative style transfer are some of the most common applications of deep learning technology. Visual art styles are also frequently transferred using methods based on deep learning. The goal of this paper is to develop an algorithm for rapid visual art style migration based on building confrontation networks. The CelebA-HQ celebrity faces dataset, collected from multiple sources of artistic and product design images, was used to assess the effectiveness of style transfer. The preprocessing stage involves resizing all images to a fixed resolution and Min-max normalizing, ensuring consistent input and stable GAN training for effective visual style migration. A feature extraction technique, such as Histogram of Oriented Gradients (HOG), is employed to evaluate structural consistency before and after style migration. This paper introduces a novel method for automatically generating image-quality art using the Cycle Generative Adversarial Network (CycleGAN) and a migration algorithm. The CycleGAN produced high reconstruction quality with a PSNR of 23.84 and strong structural consistency with an SSIM of 0.81, confirming its effectiveness in artistic style migration for packaging design. To enhance the effect of image creative style migrations, the image is identified using a multiscale discriminator. The results of the trial show that this strategy is very valuable and applicable for advertising and use.

Povzetek: Obravnav je problem neparnega prenosa umetniških slogov v embalažno oblikovanje. Predlagana je metoda CycleGAN z večlestvenim diskriminatorjem in HOG-analizo za ohranjanje strukture. Model doseže visoko kakovost prenosa ter učinkovito in koherentno ustvarja nove slogovne vizualizacije za industrijsko in komunikacijsko oblikovanje.

1 Introduction

Images are the first form of creative endeavor and an integral part of humanity's culture since they reflect their authors' complex and unusual ideas and feelings. A great piece of drawing reflects the individual imaginative flair of the creator. Creators significantly enhance their creative abilities by studying the distinctiveness of the artistic style. In this context, when computer technology develops and is used more quickly, more and more people are becoming interested in machine vision and image processing in addition to conventional instruction in art theory [1]. By carefully employing computer vision technology, artists may learn how to express their distinctive, imaginative formation style by studying real-world settings or images and applying the correct painting methods. In recent years, specialists and academics in artificial intelligence have begun to pay more attention to the generative adversarial network (GAN), a crucial subfield of deep learning. The produced information and the actual example are separated using the discriminator, and the generator settings are changed to make the generated data seem more accurate [2]. The generator creates information using noise that is near the original sample. The network completes the objective of the position by producing data that is as close to the actual sample as possible through the generator's and discriminator's adversarial training, which simultaneously improves the accuracy of the discriminator's categorization and the generator's forecast. Additionally, as research has advanced, many GAN variations have been used, and the GAN's image applications have grown significantly [3]. Many academics now use GAN-based style migration algorithms for works of art. Painting techniques were transferred to photos utilizing a synthesis of the underlying image's texture characteristics, while the pictures' lexical importance was ignored. Using a convolutional neural network (CNN), it is feasible to extract significant information from images and modify the style to produce more realistic images [4].

A generalized style migration technique that migrates any number of image styles to a single model by directly matching the feature covariance of content and style images through whitening and coloring modifications. A CycleGAN was used to perform unpaired visual transformation between images of product packaging and artistic style domains. GAN-based image style migration techniques are being employed more and more frequently in creating different forms of paintings as art, and successful product packaging design direction outcomes have been attained [5]. The majority of the artworks created utilizing style migration techniques may keep the majority of the Outline and general shade data of sincere photographs, provided the training data are sufficient. However, according to a recent study, they still need to be artwork-level flawless. For instance, it can be challenging to match the content structure of particular photographs with important subject aspects and more complicated semantic information through style migration [6]. Consequently, some produced photos exhibit local features that must be added or blurred, local color distortion, and emphasized image topics. The researchers provide a method for migration of the image-to-artwork style based on Cycle-GAN to more effectively accomplish style migration from the artworks from the viewpoint of visual study. To create artwork pictures in a three-dimensional creative class, the artistic color, shape, texture, and dimension are combined with artistic qualities [7].

1.1 Problem statement

In many traditional visual design processes, the process of integrating artistic styles into industrial applications tends to lack efficiency and flexibility. Existing style transfer methods encounter challenges in the conservation of structural coherency and crucial content across domains. Thus, the need for an automated method allowing efficient and predictable relocation of artistic styles exists. The present methods cannot produce visually acceptable results for packaging and communication design. Therefore, for our approach, we have proposed a CycleGAN method with multiscale discrimination and HOG feature analysis to overcome these limitations.

The research aims to develop a CycleGAN-based algorithm that allows quick and coherent visual art style transfer within an industrial, packaging, and communication design context. Structural consistency is maintained through the use of Histogram of Oriented Gradients (HOG) as an evaluation of the features, and a multiscale discriminator is introduced to allow for the transfer of style in an accurate and detailed manner. This methodology provides a streamlined creative process, which enables the combination of artistic elements into product design to enhance productivity and engagement in visual communication.

1.2 Contributions of the research

- The research aims to advance a CycleGAN-based framework for rapid visual art style migration, enhancing creative design processes across industrial and communication domains.
- CelebA-HQ celebrity faces data was collected from diverse visual artworks and product design samples, permitting training and evaluation across multiple artistic styles.
- The preprocessing stage involved resizing all images to a fixed resolution and applying Min-Max normalization to ensure consistent input dimensions and stable CycleGAN training.
- Features were extracted using Histogram of Oriented Gradients (HOG) to assess and maintain structural consistency before and after style migration.
- The adopted method employed a CycleGAN architecture enhanced with a multiscale discriminator to preserve fine-grained style details during unpaired image-to-image translation.
- The model effectively enabled seamless style transfer, generating visually coherent designs suitable for product development, packaging, and visual storytelling applications.
- The results confirmed that the CycleGAN-based strategy facilitated stylistic adaptability in design outputs, with strong applicability in creative industries, particularly in advertising and visual communication design.

The rest of the paper is structured as follows: A summary of related works is provided in Part 2, a more detailed description of the methodologies is given in Part 3, the results and the discussion are covered in Part 4, and the conclusion is provided with future scope in Part 5.

2 Related works

The research creates a colour images as colored pencil sketches using cluster-based segmentation, two-tone mapping, line integral convection is used collectively for texture generation the contours formerly from a neon transform [8]. Blending the textured layers with the contours produces a stylized sketch effect. The results showed that the method allows for an extremely fast and efficient option to transform images with attractive results mimicking pencil drawings. However, implementations of the method exhibited challenges for accurately capturing the deep subtle tonal transitions and erratic stroke motion inherent in handmade/hand drawn artwork. The research shows improved visual quality and structure through object-specific style transfer by utilizing Wavelet Transform (WT) and Visual Geometry Group Network (VGGNet19) [9]. The result of our research exceeds the

current ways of creating images in realism, and the correlation of intrinsic features in images. But the method is computationally intensive and, in making them perform well requires constant tuning of wavelet parameters and a comprehensive understanding of the content class being converted. The research developed Deep Reinforcement Learning (DRL) to enhance autonomous driving control in dynamic traffic scenarios [10]. The implementation leverages AirSim as a simulation environment, via Deep Deterministic Policy Gradient (DDPG) and Recurrent DDPG (RDPG), and a Convolutional Neural Network (CNN) to process images of each timestep in real-time. Additionally, developed an original reward system to help to improve convergence times and control accuracy. These results confirm the potential of DRL-based methods for enhancing autonomous vehicle control performance. The limitation included to how can manage rare edge cases and derive generalization of conclusions across myriad driving scenarios. To improve artistic image style transfer, a modified CycleGAN model incorporated into the generator architecture the use of an attention mechanism was created [11]. The investigation was to maintain image detail, blur the boundaries of the target, and improve stylization. The usability findings for our experiments show better visual quality and user satisfaction compared to the standard CycleGAN. The model performs with increased clarity and application value in artistic image processing. Research introduced Edge feature and Self-Attention (ESA-CycleGAN), a style transfer model which uses edge features and self-attention [12]. The architecture used a generator, discriminator, and an edge feature extraction network. Self-attention is used to catch global features in an image and edge features used to preserve fine details. Experimental results showed ESA-CycleGAN has a higher Inception Score (IS) and Fréchet Inception Distance (FID) which showed the model has better image quality and fine detail preservation. The limitation of the model is its difficulty dealing with very complex textures along with the amount of computational resource it requires for training. A related works analysis, a literature review, or a systematic review is discussed regarding image migration based on CycleGAN. It requires a comprehensive examination and evaluation of the research body, shown in Table 1.

Table 1: Related works

Ref	Objective	Findings	Limitation
Wang [13]	Digital art classification and aesthetics evaluation using hierarchical deep learning	The model employs Dense Convolutional Network (DenseNet), Convolutional Next (ConvNeXt), and Vision Transformer (ViT) models to more accurately acquire and classify complex artistic features.	limited to classification tasks, does not accommodate for real-time artistic style migration or throughout other modes of image generation transformation
Jin [14]	Design a cross-modal attention-based generative adversarial network (CMAGAN) for converting text into high-quality artistic design images.	Proposed a three stage Generative Adversarial Network (GAN) with Recurrent Neural Network (RNN) encoding and Attention-on-Attention (AoA) to generate semantically aligned high-fidelity artistic images using the low-dimensional text embedding.	Concentrated on text-to-image generation; does not address image-to-image style transfer or multi-domain image migration.
Li and Guo [15]	Develop a brain visual cognition-based image classification model integrating deep convolutional neural networks (DCNN).	Integrated DCNN with Long Short-Term Memory (LSTM) and attention mechanisms to simulate brain visual processing; achieved high accuracy across visual regions and reduced loss	The model is focused on EEG-based image classification. This model is not able to do any real-time image generation, artistic style transfer.
Liu and Zhou [16]	Investigated automated layout generation in graphic design with the generative adversarial network for layouts (LayoutGAN)	Showed that the wireframe rendering discriminator provides better layout quality than a relation-based discriminator, allowing for fully automated room floor plan generation.	The model reliant on real-time designer input because intelligent layout continues to be semi-automated as a result of black-box constraints that occur in statistical learning.
Lee et al. [17]	Create a soft computing neural network model to	Focused on altered facial landmarks using only RGB image data, while	It only applies to facial forgery detection; it cannot account for

	detect fake facial images generated by Handcrafted Facial Manipulation (HFM)	deliberately avoiding use of any metadata; achieved improved detection performance compared to state-of-the-art classifiers.	any other form of image manipulation
Wang et al. [18]	Enhance ink painting style transfer by addressing domain asymmetry	Asymmetric CycleGAN with Salient Edge and Feature-wise Cycle Loss	Improved brushstroke detail, ink diffusion, and efficiency over existing methods.
Jiangsha et al. [19]	NDT image simulation via extra-supervised CycleGAN	Used x-ray welds with label-guided CycleGAN	Technical; not focused on artistic transfer or real-time adaptability
Suwannik [20]	Use CycleGAN with randomly generated data to transfer traditional art style to silhouette images	Achieved better style transfer results on black-and-white silhouettes compared to previous methods.	Issues remain with artifacts and spikes in the output images.

2.1 Research gap

Earlier research exposed several limitations in AI-driven style transfer applications. Fine art generation using CycleGAN lacked contextual style depth and remained confined to sketch-based rendering [8]. Chinese painting synthesis relied on domain-specific deep learning, with limited adaptability beyond classical styles or low-resource environments [9]. Style transfer for animated costumes using FCN and CycleGAN achieved task-specific coherence, but produced low style fidelity and faced inconsistencies due to dataset limitations [10]. The present technique overcomes these limitations by integrating multiscale discriminators and HOG-based structural consistency checks to preserve detail and context across diverse visual domains. CycleGAN is adapted for generalized, high-fidelity style migration, enabling broader applicability beyond domain-specific constraints.

3 Methodology

The research aims to use a CycleGAN-based approach for visual art style migration. Preprocessing involves resizing and min-max normalizing images to guarantee consistent input. Histogram of Oriented Gradients (HOG) is applied as a feature extraction method to assess structural consistency both before and after a style translation. A multiscale discriminator improves style fidelity and detail preservation. The technique is effective in generating visually coherent outputs for industrial, communication, and packaging design, as shown in Figure 1.

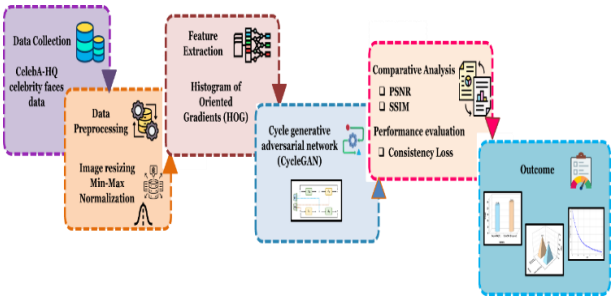


Figure 1: Semantic diagram of the CycleGAN approach

3.1 Data collection

The CelebA-HQ celebrity faces data was collected from the source Kaggle. A dataset consisting of 30,000 high-quality celebrity faces resampled to 256px. NVIDIA used this dataset in the research paper Progressive Growing of GANs for Improved Quality, Stability, and Variation. This dataset is meant for generative models: only images with no attributes/labels.

Source:
https://www.kaggle.com/datasets/badasstechie/celebahq-resized-256x256/data

3.2 Preprocessing

Preprocessing is the first step in deep learning (DL). It consists of preparing the raw inputs for model training. This typically consists of bringing the raw images into a consistent format and quality. Preprocessing typically resizes images to the same resolution and normalizes pixel values to lessen input discrepancies and isolate variability to help in obtaining stable training results. These steps help the model embed meaningful pattern information and increase residual variability in the input.

❖ **Resizing**

It allows to ensure all input images are the same size before they are passed into the CycleGAN model. This consistent size enhances training stability and quality of the style transfer. Using fixed-size inputs better manages computational complexity and memory usage. During the resize process, the input images keep their potential salient visual items while ultimately transforming non-salient visual feature variation. This means the model is able to learn style patterns in a more effective manner across the two domains.

❖ **Min-max Normalization**

The pre-processing phase begins with min-max normalization so that the pixel values are on the same scale for all input images. This linear transformation of pixel values (min-max normalization) produces a consistent value distribution, which allows CycleGAN to use consistent values during training. The normalization method is defined in equation (1):

$$W_{new} = \frac{W-\min(w)}{\max(w)-\min(w)} \quad (1)$$

Where W_{new} is the normalized pixel value, W is the original pixel value, max(w) and min(w) represent the minimum and maximum values in the image dataset. The Min-max normalization ensures that pixel values are standardized across images for uniform input. This results in a more stable training environment for the CycleGAN model and allows for more accurate style transfer. This allows the model to create aesthetically stable and structurally consistent results in art style migration.

3.3 Feature extraction using histogram of oriented gradients (HOG)

HOG advance a rapid visual art style migration algorithm using CycleGAN, enhancing image translation for industrial and visual communication design. To support rapid and coherent visual art style migration, feature extraction using HOG is combined to evaluate structural consistency across original and transformed images. HOG captures essential shape, edge, and orientation information, guaranteeing that key visual features, especially those critical to product design and visual communication, remain intact after style translation using CycleGAN. This strengthens the fidelity of creative style migration by quantifying preservation of structural details during the generative process. Block-wise gradient vectors are normalized to reduce lighting sensitivity, using the equation (2)

$$e = \frac{u}{\sqrt{||u||_2^2 + f^2}} \quad (2)$$

where u is the unnormalized gradient vector and f is a small constant. This ensures robustness in the detection of stylistic preservation. This allows for accurate measurement of creative transitions across visual domains. HOG ensures that essential structural elements are preserved during style migration. This enhances the reliability of CycleGAN-generated outputs by maintaining edge clarity and design integrity across visual transformations.

3.4 CycleGAN

CycleGAN was invented to overcome the unpaired image-to-image translation problem in computer vision. The essence of CycleGAN is to enforce cycle consistency to overcome mode collapse, which is common in other GANs. In particular, CycleGAN contains two generators $S_\Theta: x \to y$ and $L_\Upsilon: y \to x$, two discriminators φ_Φ and ψ_Ξ, which are implemented using weight-parameterized neural networks $\Theta, \Upsilon, \Phi, \Xi$, for the picture conversion issue between two domains y and x, as illustrated in Figure 2.

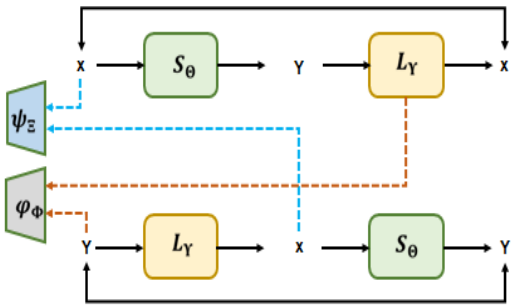


Figure 2: Standard CycleGAN architecture

Figure 2 shows the standard architecture of CycleGAN, adapted for unpaired style transfer between original product images (domain x) and an artistic style (domain y). This structure enables visual art style migration by allowing for consistent translation between artistic styles and existing design projects while maintaining core visual characteristics. The following min-max problem is then solved to estimate the unknown generator parameters Θ and Υ as presented in equation (3)

$$\min_{\Theta,\Upsilon\Phi,\Xi} max f_{CycleGAN}(\Theta, \Upsilon\Phi, \Xi) \quad (3)$$

Where Θ, Υ are the generator parameters optimized to minimize the CycleGAN loss by translating images between two visual domains. Φ, Ξ are the discriminator parameters optimized to maximize the same loss by distinguishing real from generated images in each domain. This optimization enables successful visual art style migration through learned bidirectional mappings that retain structured features while transferring artistic styles of interest from source designs and target visual domains. When loss is indicated by equation (4)

$$f_{CycleGAN}(\Theta, \Upsilon\Phi, \Xi) = f_{Cycle}(\Theta, \Upsilon) + f_{Disc}(\Theta, \Upsilon\Phi, \Xi) \quad (4)$$

Where $f_{CycleGAN}(\Theta, \Upsilon)$ is the cycle consistency loss, ensuring that an image translated from one domain to another and then back remains close to the original, preserving core content. Θ, Υ, and discriminators Φ, Ξ perform in creating realistic images and detecting fakes. It enables effective visual art style migration by maintaining key design features while transferring stylistic attributes. The cycle-consistency term offered in this context is$f_{Cycle}(\Theta, \Upsilon)$, which is denoted by equation (5)

$$f_{Cycle}(\Theta, \Upsilon) \coloneqq \int_y \left\| y - S_\Theta\left(L_\Upsilon(y)\right)\right\| t\mu(y) + \int_x \left\| x - L_\Theta\left(S_\Theta(x)\right)\right\| tc(x), \quad (5)$$

Here, S_Θ represents the Generator that maps from the domain x. L_Υ is the generator that maps from domain y back to domain x, $y - S_\Theta\left(L_\Upsilon(y)\right)$ measures how much the image in domain y, after being mapped to domain x and back, deviates from its original form, enforcing backward consistency. $\mu(y), c(x)$ are the probability distributions over images in domains y and x, respectively. t typically represents the norm type (often L_1or L_2) used for measuring reconstruction error. The discriminator loss is$f_{Disc}(\Theta, \mathcal{H}; \Phi, \Xi)$ is described as equation (6)

$$f_{Disc}(\Theta, \mathcal{H}; \Phi, \Xi) = \int_y log(\varphi_\Phi(\mathrm{y}))t\mu(y) + \int_x log(1 - \varphi_\Phi(S_\Theta(y)))tc(x) + \int_x log(\psi\Xi(x))tc(x) + \int_x log(1 - \psi_\Xi(L_\Upsilon(y)))t\mu(x), \quad (6)$$

Where μ and c are, y and x probabilities, respectively. CycleGAN $\varphi_\Phi$attempts to differentiate between the real $y \in Y$ and the false created by$S_\Theta(x)$, and ψ_Ξ is the discriminator that determines the dissimilarity among the real $x \in X$ and the one generated by $L_\Upsilon(y)$.

The algorithm 1 employs artificial intelligence as a method to seamlessly embed creative visual styles into industrial ideas, opening up previously unimaginable opportunities for creativity. The CycleGAN also enables the generation of visually coherent depictions of style-based designs—based on the pre-established structure of meaning. Thus, aiding practical applications for packaging, product aesthetics, and visual narratives within the industry.

Algorithm 1: CycleGAN

Step 1: BEGIN

Step 2: FOR training iterations, ACTIVATE:

Step 3: Take n data samples from the W field$\{W^{(1)}, .. W^{(m)}\}$

Step 4: Take m data samples from the Y field $\{Z^{(1)}, .Z^{(n)}\}$

$$K_{real}^{(D)} = \left(\frac{1}{m}\right)\sum_{j=1}^{M}(C_w\left(W^{(j)}\right) - 1)^2 + \left(\frac{1}{m}\right)\sum_{j=1}^{M}(C_Z\left(Z^{(j)}\right) - 1)^2$$

Step 5: Calculate the produced data's discriminator loss

$$K_{real}^{(D)} = \left(\frac{1}{m}\right)\sum_{j=1}^{M}(C_w\left(H(W^{(j)})\right) - 1)^2 + \left(\frac{1}{m}\right)\sum_{j=1}^{M}(C_Z\left(E(Z^{(j)})\right) - 1)^2$$

Step 6: Adjust the discriminator network's settings.

Step 7: Determine the generator's loss by $Z \longrightarrow W$

$$K^{(E)} = \left(\frac{1}{M}\right)\sum_{j=1}^{n}\left(C_w\left(E_y{}^{(j)}\right) - 1\right))^2 + K_{Cycle}^{Z \to W \to Z}$$

Step 8: Calculate the generator loss of $Z \longrightarrow W$

$$K^{(H)} = \left(\frac{1}{M}\right)\sum_{j=1}^{n}\left(C_w\left(E_y{}^{(j)}\right) - 1\right))^2 + K_{Cycle}^{Z \to W \to Z}$$

Step 8: Optimize generator network parameters.

Step 9: END FOR.

Step 10: END

4 Result and discussion

This section describes the results of the model implementation, which include performance assessment and comparison analysis. The CycleGAN-based method for unpaired image style transfer in product packaging design can be applied to capture various art forms while preserving the necessary structure. The results show the effectiveness of the model at preserving visual consistency, allowing for artistic adaptation, and improving against baseline models.

4.1 Experimental setup

The implementation of the CycleGAN-based visual style transfer model was developed using Python 3.11.4. The experiments were run on a high-performance desktop equipped with 64 GB RAM and an AMD Ryzen 5900X processor under Windows 11. This configuration ensured efficient training and evaluation of the model across multiple unpaired image domains, enabling consistent testing of visual style adaptation for both artistic and product design applications.

4.2 Parameters setup

The hypermeters for the CycleGAN method are described in Table 2.

Table 2: Parameter's setup

Hyperparameter	**Selected Values**
Hidden units per dense layer	256, 512
Epochs	100
Dropout rate	0.5
Optimizer	CycleGAN
Batch size	64
Learning rate	0.0002
Number of filters	64
Activation function	ReLU, Leaky ReLU
Number of convolutional layers	3
Pooling size	2×2

4.3 Performance evaluation

In the performance assessment, Consistency Loss is used to evaluate how well the model retains structural information as it undergoes style transfer. A downward trend in the observed loss in training epochs indicates a more effective representation of possible image reconstructions, along with stable translations and a consistent model, which means that the model retained important elements from the original packaging design during several transfers using artistic style.

❖ **Consistency loss**

It calculates the difference between the original image and the image reconstituted after performing one forward and one backward translation. It guarantees that critical structural characteristics of product packaging do not change as a result of style transfer. The lower the amount of consistency loss components, the greater the content that remains unchanged during artistic migration. Figure 3 displays the diminishing consistency loss trend during CycleGAN training for package style adaptation.

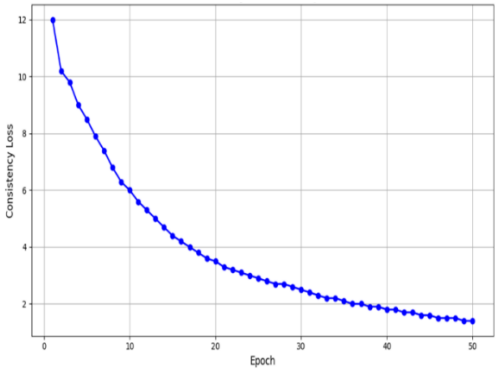


Figure 3: Consistency loss trend in the style transfer process

The graph shows a drastic decline in Cycle Consistency Loss from 12.0 at epoch 1 down to approximately 1.4 at epoch 50, which indicates an improving quality of reconstruction. This decreasing trend substantiates the model's ability to maintain key packaging forms while transferring the artistic style from Picasso to Warhol. The consistently falling loss indicates stable training of the CycleGAN and strong structural retention.

4.4 Comparison phase

In the comparison phase, the research explores the visual translation performance of CycleGAN in comparison with Exposition-Generative Adversarial (Expo-GAN) [21], with Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) as evaluation metrics. Expo-GAN emphasizes expressive style control, while CycleGAN enforces cycle consistency and multiscale discrimination. The evaluation stresses reconstruction accuracy and structural similarity between generated and target images. Table 3 shows the comparative evaluation results.

Table 3: Performance comparison of CycleGAN and Expo-GAN

Models	**PSNR**	**SSIM**
Expo-GAN [21]	21.39	0.76
CycleGAN [Proposed]	**23.84**	**0.81**

Table 3 presents the comparison of the image translation quality between Expo-GAN and the proposed CycleGAN. The CycleGAN has a higher PSNR of 23.84; its image reconstruction fidelity is better than that of Expo-GAN, which scored 21.39. For SSIM, CycleGAN acquired a somewhat higher value of 0.81, indicating better structural similarity as compared to 0.76 for Expo-GAN. This further shows that CycleGAN is improved at maintaining visual detail and style consistency.

❖ **Peak Signal-to-Noise Ratio (PSNR)**

PSNR measures the quality of image reconstruction by comparing the maximum power allowable of a signal to the power of the noise that could corrupt the signal. A larger PSNR value means that the translated image retains more of the original image detail and has lower noise distortion, indicating better visual quality in style-transferred images. Figure 4 indicates the PSNR improvement of CycleGAN compared to Expo-GAN for style-transferred package images.

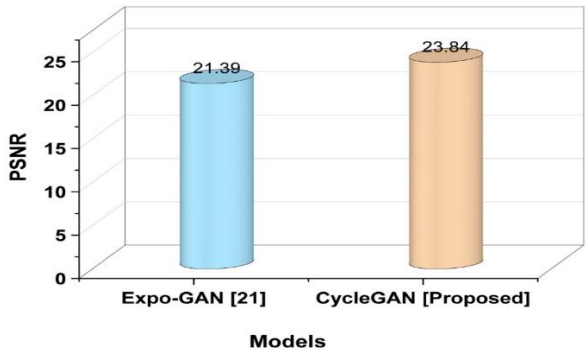


Figure 4: PSNR comparison between Expo-GAN and CycleGAN models

The CycleGAN achieved a PSNR of 23.84, indicating higher fidelity in preserving image quality compared to Expo-GAN, which recorded a PSNR of 21.39. This improvement reflects reduced distortion and better structural consistency in the translated outputs. The comparison highlights CycleGAN's advantage in maintaining visual clarity during style transfer.

❖ **Structural similarity index measure (SSIM)**

SSIM analyzes image quality as a function of its luminance, contrast, and structural components. It provides a measure of the similarity between the transformed image and the reference, with a focus on preserving perceptual consistency. High SSIM values indicate accurate preservation of visual qualities in the successively transferred style to facilitate extended transfer of artistic features. Figure 5 demonstrates the SSIM value of CycleGAN compared to Expo-GAN, demonstrating better structural preservation.

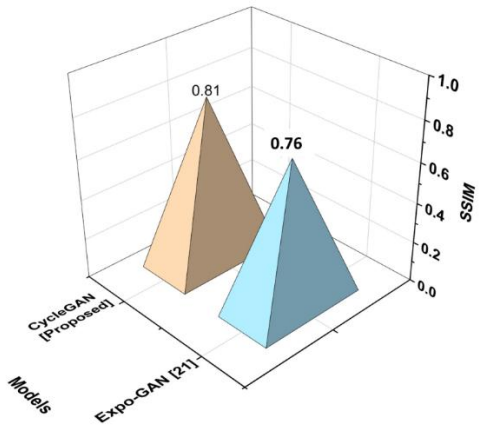


Figure 5: Structural similarity comparison between proposed models

The SSIM value of 0.81 for CycleGAN suggests, compared to 0.76 for Expo-GAN, superior structural similarity with the target image. This indicates CycleGAN's superior ability to maintain texture, edges, and perceptual detail during style transfer. The higher score indicates that CycleGAN has superior visual coherence.

4.5 Data comparison

The comparison illustrates the differences in focus between the datasets. CelebA-HQ is focused on explicit, structured facial features and is better for examining identity preservation and realism in style transfer tasks. Meanwhile, the Art Style Migration Performance dataset [22] focuses on artistic abstraction to examine adaptability in creative space. The Art Style Migration Performance data was collected from the source Kaggle. This dataset comes from actual-time visual transformations of different design and art images. It preserves significant image features, transformation patterns, and stylistic changes during migration. The data is flattened to a structured format that is ready for assessment, comparison, and investigation in creative design, digital art, and visual media use cases. It covers various content and style domains as they apply to contemporary visual communication and design processes. A comparison of CycleGAN's performance on two datasets is displayed in Table 4.

Table 4: Comparison of CycleGAN performance across datasets

Metrics	**CelebA-HQ**	**Art Style Migration [22]**
PSNR	23.84	21.97
SSIM	0.81	0.67

The PSNR score for CelebA-HQ reached 23.84, which means that the image quality after style migration was comparatively better than 21.97 for the Art Style Migration dataset. The SSIM values also indicate greater structural similarity in CelebA-HQ (0.81) compared to the Art Style Migration set (0.67), which signifies a greater preservation of features. These results show that CycleGAN was much more efficient on structured facial data compared to the Art dataset because the lower metric scores in the Art dataset indicate more variety and complexity that are more difficult to process as diverse artistic textures.

4.6 Discussion

The research aims to create a CycleGAN-based model for unpaired image style transfer in product packaging design, maintaining product identity while emphasizing various artistic styles. Expo-GAN [21] cannot imitate the

structural intricacy of the layout or contain distinctive visual details, often resulting in incorrect or visually distorted outputs. Therefore, there was a lack of brand identity that maintained the brand after producing package layouts, in addition to never receiving fidelity styles with complex layouts. The recommended CycleGAN model addresses these limits by employing structural feature alignment and multiscale discrimination, producing visually consistent, brand-aligned packaging designs that preserve essential content while adapting creative visual characteristics.

5 Conclusion

The research aims to develop effective approaches for artistic style migration using CycleGAN for packaging design. Data were gathered from the CelebA-HQ dataset and from several artistic and product design sources to create visual diversity. Image data was pre-processed, including resizing and min-max normalization, to develop common inputs for stable GAN training. To extract structural features, HOG (Histogram of Oriented Gradients) was used to evaluate content preservation across style translations. The method described here uses CycleGAN, having a multiscale discriminator to extract detail and to generalize the varying visual styles to packaging, while retaining important product features. Results demonstrated strong reconstruction quality, resulting in a PSNR of 23.84 and SSIM of 0.81, showcasing the high-level visual fidelity and structural consistency of outputs. A limiting component is that performance is weaker when subjected to highly abstract or non-uniform artistic styles. Future work may consider adaptive attention methods or domain-aware training strategies for generalization across wider artistic domains and better creative flexibility in the potential packaging transformations.

Declaration

Ethics approval and consent to participate: I confirm that all the research meets ethical guidelines and adheres to the legal requirements of the study country.

Consent for publication: I confirm that any participants (or their guardians if unable to give informed consent, or next of kin, if deceased) who may be identifiable through the manuscript (such as a case report), have been given an opportunity to review the final manuscript and have provided written consent to publish.

Availability of data and materials: The data used to support the findings of this study are available from the corresponding author upon request.

Competing interests: here are no have no conflicts of interest to declare.

Authors' contributions (Individual contribution): All authors contributed to the study conception and design. All authors read and approved the final manuscript

References

[1] Caporusso N (2023). Generative artificial intelligence and the emergence of creative displacement anxiety. *Research in Psychology and Behavior*, 3(1). https://doi.org/10.53520/rdpb2023.10795.

[2] Han X, Wu Y, & Wan R (2023). A method for style transfer from artistic images based on depth extraction generative adversarial network. *Applied Sciences*, 13(2), 867. https://doi.org/10.3390/app13020867.

[3] Batziou E, Ioannidis K, Patras I, Vrochidis S, & Kompatsiaris I (2023). Artistic neural style transfer using CycleGAN and FABEMD by adaptive information selection. *Pattern Recognition Letters*, 165, 55–62. https://doi.org/10.1016/j.patrec.2022.11.026.

[4] Wei Y, Ji M, Lv J, & Zhang X (2022). Research on image to illustration translation method based on CycleGAN. *International Journal of Computer Applications in Technology*, 69(3), 244–252. https://doi.org/10.1080/14626268.2023.2174557.

[5] Vartiainen H, & Tedre M (2023). Using artificial intelligence in craft education: crafting with text-to-image generative models. *Digital Creativity*, 34(1), 1–21. https://doi.org/10.1080/14626268.2023.2174557.

[6] Peng X, Peng S, Hu Q, Peng J, Wang J, Liu X, & Fan J (2022). Contour-enhanced CycleGAN framework for style transfer from scenery photos to Chinese landscape paintings. *Neural Computing and Applications*, 34(20), 18075–18096. https://doi.org/10.1007/s00521-022-07432-w.

[7] Liao Z, & Lin G (2023). Research on sketching face headshot generation based on improved CycleGAN. *Frontiers in Computational Intelligence Systems*, 4(1), 60–62. https://doi.org/10.1007/s11042-023-17746-z.

[8] Cui Z (2022). Application of line integral convolution techniques in the generation of fine art painting effects. *Mobile Information Systems*, 2022, 6872214. https://doi.org/10.1155/2022/6872214.

[9] Raghatwan JS, & Arora S (2024). Novel entropy-based style transfer of the object in the content image using deep learning. *Bulletin of Electrical Engineering and Informatics*, 13(5), 3418–3429. https://doi.org/10.11591/eei.v13i5.7659.

[10] Chang CC, Tsai J, Lin JH, & Ooi YM (2021). Autonomous driving control using the DDPG and RDPG algorithms. *Applied Sciences*, 11(22), 10659. https://doi.org/10.3390/app112210659.

[11] Wei Y (2024). Artistic image style transfer based on CycleGAN network model. *International Journal of Image and Graphics*, 24(4), 2450049. https://doi.org/10.1142/S0219467824500499.

[12] Wang L, Wang L, & Chen S (2022). ESA-CycleGAN: Edge feature and self-attention-based cycle-consistent generative adversarial network for style transfer. *IET Image Processing*, 16(1), 176–190. DOI:10.1049/ipr2.12342

[13] Wang Y (2025). Hierarchical multi-stream feature network for digital art aesthetic quality and style classification through intelligent systems. *Informatica*, 49(22). https://doi.org/10.31449/inf.v49i22.8065.

[14] Jin H (2025). Cross-modal attention GAN for text-to-artistic image generation. *Informatica*, 49(20). https://doi.org/10.31449/inf.v49i20.8303.

[15] Li X, & Guo H (2025). Fusion of deep convolutional neural networks and brain visual cognition for enhanced image classification. *Informatica*, 49(16). https://doi.org/10.31449/inf.v49i16.7787.

[16] Liu S, & Zhou L (2025). LayoutGAN for automated layout design in graphic design: An application of generative adversarial networks. *Informatica*, 49(10). https://doi.org/10.31449/inf.v49i10.7099.

[17] Lee S, Tariq S, Shin Y, & Woo SS (2021). Detecting handcrafted facial image manipulations and GAN-generated facial images using Shallow-FakeFaceNet. *Applied Soft Computing*, 105, 107256. https://doi.org/10.1016/j.asoc.2021.107256.

[18] Wang W, Li Y, Ye H, Ye F, & Xu X (2023). Ink painting style transfer using asymmetric cycle-consistent GAN. *Engineering Applications of Artificial Intelligence*, 126, 107067. https://doi.org/10.1016/j.engappai.2023.107067.

[19] Jiangsha A, Tian L, Bai L, & Zhang J (2022). Data augmentation by a CycleGAN-based extra-supervised model for nondestructive testing. *Measurement Science and Technology*, 33(4), 045017. https://doi.org/10.3390/su142114304

[20] Suwannik W (2022). Style transfer of black and white silhouette images using CycleGAN and a randomly generated dataset. *arXiv preprint*, arXiv:2208.04140. DOI:10.48550/arXiv.2208.04140

[21] Xie Q, & Yu R (2025). Expo-GAN: A style transfer generative adversarial network for exhibition hall design based on optimized cyclic and neural architecture search. *Computers, Materials & Continua*, 83(3). https://doi.org/10.32604/cmc.2025.063345

Continuous Voltage Regulation in Active Distribution Networks Using Twin Delayed Deep Deterministic Policy Gradient

Jiangyan Chen[1,2], Long Kou[2,3,*], Yuhua Wang[1], Yilin Liu[1]
[1]School of Artificial Intelligence and Electrical Engineering, Guangzhou College of Applied Science and Technology, Guangzhou, Guangdong, 511370, China
[2]International College, National Institute of Development Administration, Bangkok, 10240, Thailand
[3]School of Management, Guangzhou Huashang College, Guangzhou, Guangdong, 511300, China
E-mail: 18011832010@163.com
[*]Corresponding author

Keywords: active distribution networks (ADNs), electric vehicles (EVs), renewable energy sources (RESs), twin delayed deep deterministic policy gradient (TD3), voltage regulation

Received: August 31, 2025

Active Distribution Networks (ADNs), characterized by high levels of penetration by Electric Vehicles (EVs) and renewable energy sources (RES), lead to a high degree of uncertainty and control challenges for many system operators. Most previous studies examined methods for controlling voltage, but the Finite Action Space (AS) set a limit on the control granularity and scalability of the methods investigated. With this in mind, the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm is presented, a Continuous-Action (CA) actor-critic reinforcement learning algorithm in which the same system model, reward structure, and EV participant constraints are kept for a direct comparison. The TD3-based controller provided real-valued control actions for distribution system resources, fully approving of the additional flexibility provided by continuous ASs. The issue of value overestimation was overcome by combining TD3 with twin Critic Networks (CNs), while the features of target smoothing and delayed policy updates are also introduced to strengthen the stability and convergence of the learning algorithm. The simulations with the IEEE 33-bus (IEEE-33) and IEEE 123-bus (IEEE-123) systems highlighted improvements in voltage control granularity, convergence speed, and scalability across uncertain State Spaces (SSs).

Povzetek: Abalizirano je uravnavanja napetosti v aktivnih distribucijskih omrežjih z visoko penetracijo EV-jev in obnovljivih virov. Predlaga metodo TD3, ki z zveznimi dejanji, dvojnima kritikoma ter zakasnjenimi posodobitvami politike zmanjšuje precenjevanje Q-vrednosti in izboljša stabilnost. Simulacije (IEEE-33, IEEE-123) pokažejo boljšo točnost, hitrejšo konvergenco in večjo razširljivost kot DQN/DDQN.

1 Introduction

The increasing installation of Distributed Energy Resources (DERs) like Photovoltaics (PVs), along with the surge in EV deployments, is exacerbating the voltage stability challenges of the ADNs [1]. This new situation of uncertainties through bidirectional power flow and dynamic load behavior will require adaptive Voltage Regulation (VR) strategies that are faster and more adaptable than model-based or rule-based control systems. Deep reinforcement learning (DRL) has shown promise for this type of VR; however, discrete-action formulations such as the Averaged Weighted Double Deep Q-Network (AWDDQN) have limits with respect to coarse control resolution and scalability bottlenecks [2]. Therefore, the goal of this study is to provide an alternative to these limitations and offer a CA voltage control technique using the TD3 algorithm.

Conventional VR methods manage EV charging/discharging profiles or utilize multiple grid-side resources to improve voltage profiles. For instance, the model-based approaches leverage convex optimization and heuristic approaches to manage the EV charging actions for cost reduction and voltage support [3], [4]. However, the temporal uncertainty of access to EVs and spatial diversity of the charging needs introduce stochastic schedulable capacity constraints that complicate the control model [5]. To simplify this complexity, hierarchical and distributed frameworks are suggested. Some works model EV aggregators (EVAs) to reduce the complexity of optimizations and support scalable control [5]. Other works applied predictive and Monte Carlo methods to predict EV loads and optimize demand side voltage with multistage or heuristic solvers [6], [7], [8]. Other works use game-theoretical mechanisms [9] and distributed model predictive control (MPC), which is necessary for peer-to-peer coordination [10] to handle decentralized VR with EVA bidding or reactive power optimization.

Despite their effective structures, many of these strategies rely on correct predictions or centralized processing, restricting their versatility when the situations evolve rapidly. To circumnavigate these limitations, model-free reinforcement learning (RL) approaches are being studied for voltage control, as well. RL methods, such as Q-learning, were used to study the action of tap changer and capacitor switching [11], [12]. However, Q-learning has not been practical in ADNs due to the "curse of dimensionality," whereby a large number of controllable nodes and dynamic states have made traditional Q-learning impossible. Therefore, DRL methods such as Deep Q-Networks (DQN) were proposed. Continuous Q-learning and mapping to DQN facilitate the mapping of the state-AS in this new setting using neural networks to act as an interface. DQN has been mainly utilized for coordinating the active and reactive power outputs of energy storage and inverter-based resources for control of voltage [13], [14], [15]. A few studies have used DQN in combination with other DQN models to enable multi-timescale control or DQN as part of an approach with deterministic policy gradients to shape EV charging profiles for stabilizing voltages [16], [17].

research for more resilient and scalable VR methods, especially with high levels of DER and EV penetration. Zhang et al. [18] Recently, we have leveraged DRL via a physics-informed multi-agent approach where each inverter agent learns its control locally, with knowledge of the grid structure. This leads to disturbance-resistant and scalable decentralized VR using PV inverters. Golgol and Pal [19] propose a DRL-based voltage controller that learns solely from feeder-head state estimation to make quick control decisions with minimal communication infrastructure. Shi et al. [20] develop a data-driven affine control method that describes the relationship between each inverter's reactive power adjustment and its active power. In doing this they aim to learn new solutions to deal with solar uncertainty from real time distributed learning and consensus algorithms. Hierarchical schemes have also been further developed in the VR domain. Dutta et al. [21] present a receding horizon control method that operates in three stages: validating smart inverter models, jointly optimizing slow and fast voltage devices, whilst integrating an EV schedule as a grid supporting service. Similarly, Ma et al. [22] develop a hierarchical method using distributed energy storage systems, where the components have been solved using centralized model predictive control (MPC) with a full knowledge of the grid and provides set points on a global scale, whilst the decentralized agents follow certain heuristics to alter their instantaneous output, with limited communication. Zhang et al. [23] represented their work as multi-energy coordination, a strategy for integration through the use of a Power-to-Hydrogen system as a controllable load, absorbing excess generation from PV systems to help support voltage control. Despite advances in voltage control strategies in ADNs using DRL and optimization-based methods, there are two major challenges:

1. Control resolution and scale: Discrete-action DRL implementations such as DQN, DDQN, and AWDDQN require AS discretization, creating exponentially growing as more devices are adopted, resulting in high, prohibitive computational expense, and not a desirable control resolution in systems with continuous dynamics.

2. Learning instability and bias under uncertainty: The tendency to overestimate or underestimate action values is particularly problematic in stochastic settings, like many ADNs with high EV and renewable penetration. While AWDDQN improves DDQN using historical value averaging to some extent, it still relies on discrete action estimators and is sensitive to added noise and variability.

To address these difficulties, this paper offers a TD3-based voltage control framework, where the two major contributions are:

1. Continuous and scalable voltage control: The TD3 controller takes action as a real-valued output for each grid resource, avoiding discretization, and allowing scalable coordination in very high-dimensional ADNs in continuous, precise space.

2. Stability and bias-resistance to learning: Dual critics and target-action delays promote stable policies and reduced estimation bias, so that reliable training can take place in uncertain environments with more variability due to EVs and renewables.

The subsequent sections of this paper are structured in the following manner. Section II elucidates the theoretical basis of the TD3 algorithm, including its architecture and update rules, and its appropriateness for power systems control within the uninterrupted action area. Section III gives the TD3-based voltage control framework, including the system model, definition of the states and actions, reward design, and the hierarchical coordination framework for the EVA. Section IV details the simulation setup and examines the effectiveness of the suggested technique using altered versions of the IEEE-33 and IEEE-123 test systems, in terms of simulation training behavior, convergence behavior, and real-time VR ability. In conclusion, Section V recaps the key results to bring the study to a close and offers recommendations for subsequent research.

2 Principle of TD3 algorithm

2.1 DDPG algorithm

TD3 uses the Deep Deterministic Policy Gradient (DDPG) algorithm - as an actor-critic method for continuous ASs - as its baseline. The agent in RL is presented a state s, chooses an action a according to its policy, receives a reward r, and transitions to a next state s'. The process can be outlined as a Markov Decision Process (MDP) with a SS S, AS A, reward function $r(s,a)$, transition probability $P(s' \mid s,a)$, and discount factor γ. The objective of the agent is to determine a superior action-value function $Q(s,a)$ that fulfills the Bellman optimality equation.

$$Q^*(s,a) = \mathbb{E}_{s'}\left[r(s,a) + \gamma \max_{a'} Q^*(s',a')\right] \quad (1)$$

Where the expectation is taken over the next state s' In DDPG (which was created for continuous actions), the maximization over a' is not handled by brute force search,

but instead by a differentiable policy (actor) network $\mu(s)$. DDPG trains both a CN $Q(s,a \mid \theta^Q)$ which is to approximate the action-value function, and an Actor Network (AN) $\mu(s \mid \theta^\mu)$ which is to approximate the optimal policy. The critic takes in a state-action combination and forecasts the Q-value, whereas the actor converts states into continuous actions. Furthermore, DDPG uses experience replay and Target Networks (TNs), like DQN, to make training more stable. DDPG uses slowly moving target parameters θ^Q_{targ} and θ^μ_{targ} which tracks the learned parameters θ^Q, θ^μ (that are updated via polyak averaging) in order to give consistency to the targets. With the actor μ being used for the next action, the target value from the critic can be written:

$$y_i^{DDPG} = r_i + \gamma Q\left(s'_i, \mu\left(s'_i \mid \theta^\mu_{\text{targ}}\right) \mid \theta^Q_{\text{targ}}\right), \quad (2)$$

Which indicates the anticipated return from state s'. The CN is trained to minimize the mean-squared Bellman error between this target. The critic loses at iteration i is:

$$L\left(\theta_i^Q\right) = \mathbb{E}_{(s,a,r,s')}\left[Q\left(s,a \mid \theta_i^Q\right) - y_i^{DDPG}\right]^2, \quad (3)$$

The parameters θ^Q of the critic are then updated by using gradient descent to reduce the critic loss while the AN will seek to maximize Q as estimated by the critic. When optimizing the actor policy, θ^μ is adjusted in the direction that improves $Q\left(s,\mu(s \mid \theta^\mu)\right)$. In practice, the gradient of the actor is computed using deterministic policy gradient theory, which provides it.

$$\nabla_{\theta^\mu} J \approx \mathbb{E}_{s\sim\mathcal{D}}\left[\nabla_a Q(s,a \mid \theta^Q)|_{a=\mu(s|\theta^\mu)} \nabla_{\theta^\mu}\mu(s \mid \theta^\mu)\right] \quad (4)$$

Shifting the policy toward actions that have higher Q values. While DDPG will continually update the critic at each time step, it will also continue to update both the AN and TN. An unstable behavior can sometimes be observed in a naive implementation of DDPG, with a failure mode often occurring early during training, as the overestimation of values by the critic Q function begins, which incentivizes the actor to exploit this incorrect Q value, resulting in poor performance.

2.2 Overestimation and double-Q learning

Overestimation bias occurs because of the nature of function approximation and the max operator, which allows Q-values to be overestimated in a manner that is time invariant. In the DDPG context, the actor is always trying to maximize the output of the critic, so it is crucial that the overestimation of the Q value does not lead to the actor learning a divergent or suboptimal policy. A known solution (see Double Q-learning for discrete domains) is to use separate CNs, yielding two estimates $Q_1(s,a)$ and $Q_2(s,a)$. Each critic is separately initialized and trained. To combat overestimation, the minimum of the 2 critic values is selected as the target estimate, such that the clipped Double-Q technique restricts the impact of an overestimated Q-value on the target. Formally, let Q_1 and Q_2 be defined as the two critics with corresponding TN, and the target is then modified as follows:

$$y_i^{Double\text{-}Q} = r_i + \gamma \min_{j=1,2} Q_j\left(s'_i, \mu\left(s'_i \mid \theta^\mu_{targ}\right) \mid \theta^{Q_j}_{targ}\right). \quad (5)$$

By using the minimum of Q_1, Q_2 for the bootstrap target, this method encourages underestimation (if a network overestimates, the minimum still predicts an estimate closer to the truth) and therefore avoids the compounding of bias. Each CN is trained using regression to the shared target y above, which minimizes $\left(Q_j(s,a \mid \theta^{Q_j}) - y\right)^2$. This dual-'critic' approach is similar to the DDQN approach in discrete RL, but instead is designed for continuous actor-critic methods. It has been observed that simply applying Double DQN to an actor-critic architecture is ineffective, as the slow-changing policy maintains the bias correlation. While having two independent critics (similar to Double-Q-learning) works better, even unbiased estimates with high variance will destabilize training. Thus, additional techniques are needed to stabilize the training process further.

2.3 TD3 algorithm improvements

The Twin Delayed DDPG (TD3) algorithm provides three main advancements to DDPG to combat overestimation and instability. The first advancement is the "Twin" critics with clipped double-Q as described above. Using 2 critics ensures TD3 reduces overestimation bias because it takes the smallest Q-value into account when computing the target value. The second advancement is the "Delayed" policy update. TD3 keeps the actor (policy network) update consistent with the critic delay. Although TD3 updates the AN (and TN) after every two critic updates (first the target critic optimizations followed by the Target Actor (TA) updates or just one normal critic update), this delay allows TD3 to update the actor only after allowing the critic optimization steps to converge closer to the correct Q-value function. This helps keep the actor from immediately chasing noisy Q-value estimates provided directly by a critic. The third and final improvement is the smoothing of the targeted policy. TD3 incorporates a minor degree of random noise to actions chosen by the target policy (AN) before being queried at the target CNs to find the target action value. In simpler mechanics, TD3 instead of getting from the TA its output $a' = \mu\left(s' \mid \theta^\mu_{\text{targ}}\right)$ directly, it gets a sampled action:

$$a'(s') = clip\left(\mu\left(s' \mid \theta^\mu_{targ}\right) + \epsilon, a_{Low}, a_{High}\right), \epsilon \sim \mathcal{N}(0,\sigma), \epsilon \textit{ clipped to } [-c,c], \quad (6)$$

Where a_{Low} and a_{High} are the action bounds. This noisy action $a'(s')$ is then used in the target Q computation:

$$y_i^{TD3} = r_i + \gamma \min_{j=1,2} Q_j\left(s'_i, a'(s') \mid \theta^{Q_j}_{targ}\right). \quad (7)$$

The added noise (with small variance σ) averages the metric Q-values over a neighborhood of actions, and therefore prevents the policy from exploiting sharp peaks or inconsistencies in the Q-function approximator over the policy. This target policy smoothing is a regularizer that makes the critic less vulnerable to erroneous high-value spikes. The policy will not tend to a small number of actions that may be overestimated, as the target value will be based on slightly perturbed actions, as well.

3 Voltage control method based on TD3 algorithm

3.1 Voltage control model for ADNs

The intention of VR in the ADN remains to minimize the deviation of the node voltages from their nominal values over a scheduling horizon. The optimization challenge can be stated as follows:

$$min \sum_{t=1}^{T} \left(\frac{1}{N} \sum_{n=1}^{N} ts \cdot \left(U_{n,t} - U_{base} \right)^2 \right) \quad (8)$$

The system must satisfy operational constraints on voltage magnitude and resource capacities:

$$\begin{aligned} U_{min} \le U_{n,t} \le U_{max} \\ Q_{j,min} \le Q_{j,t} \le Q_{j,max} \\ P_{j,t,min} \le P_{j,t} \le P_{j,t,max} \end{aligned} \quad (9)$$

Where $P_{j,t}$ and $Q_{j,t}$ denotes the active and reactive power of the j-th controllable resource at time t, subject to upper and lower bounds.

3.2 Schedulable capacity of EVAs

Every Electric Vehicle Aggregator (EVA) is inherently a controllable active power source with time-varying characteristics based on the schedulable charging and discharging power capabilities of its fleet of EVs. For each EV, the schedulable charging capability (SCC), discharging capacity (SDC), and power bounds (SCP, SDP) are calculated as follows:

$$\begin{aligned} \mathrm{SCC}_{d,t} = C_d \left(\mathrm{SOC}_{d,s} - \mathrm{SOC}_{d,t} \right) \\ + \eta_c \left(t + ts - t_{d,s} \right) P_{d,c,\max} \end{aligned} \quad (10)$$

$$\mathrm{SCP}_{d,t} = \min \left(\frac{\mathrm{SCC}_{d,t}}{ts \cdot \eta_c}, P_{d,c,\max} \right) \quad (11)$$

$$\begin{aligned} \mathrm{SDC}_{d,t} = C_d \left(\mathrm{SOC}_{d,t} - \mathrm{SOC}_{d,\min} \right) \\ + \frac{\left(t_{d,e} - t - ts \right) P_{d,d,\max}}{\eta_d} \end{aligned} \quad (12)$$

$$\mathrm{SDP}_{d,t} = \min \left(\frac{\mathrm{SDC}_{d,t} \cdot \eta_d}{ts}, P_{d,d,\max} \right) \quad (13)$$

Aggregating over all EVs d in EVA l, the total capacity metrics are:

$$SCP_{l,t} = \sum_{d=1}^{N_{l,t}} SCP_{d,t}, SDP_{l,t} = \sum_{d=1}^{N_{l,t}} SDP_{d,t} \quad (14)$$

3.3 MDP formulation for TD3

The RL formulation follows an MDP framework with components defined as:

- SS $\mathcal{S}$:

$$s_t = \left\{ U_{i,t}, SCP_{l,t}, SDP_{l,t}, P_{j,t}, Q_{j,t} \right\} \quad (15)$$

This includes voltage magnitudes, schedulable capacities, and current output of resources. Total state dimension is $N + 2L + 2J$.

- AS $\mathcal{A}$:

In contrast to AWDDQN, the TD3 algorithm outputs continuous-valued control actions for each device directly:

$$a_t = \left\{ \hat{P}_{j,t}, \hat{Q}_{j,t} \right\}, j = 1, \ldots, J \quad (16)$$

These actions are constrained within their operational bounds:

$$\hat{P}_{j,t} \in \left[P_{j,t,min}, P_{j,t,max} \right], \hat{Q}_{j,t} \in \left[Q_{j,min}, Q_{j,max} \right] \quad (17)$$

3.4 Reward function design

The reward function within the TD3-based voltage control model reflects the objective of minimizing voltage deviations from the base value while applying soft penalties for violations. It originated from the objective function (1) in the first part of the paper, and was rewritten as:

$$r_t = -\frac{1}{N} \sum_{i=1}^{N} \lambda_i \cdot ts \cdot \left(U_{i,t+1} - U_{base} \right)^2 \quad (18)$$

Where λ_i is a penalty factor that is determined dynamically based on the size of the deviation at node i, and ts is the time step. The penalty factor is selected as:

$$\lambda_i = \begin{cases} 1 & 0 < \left| U_{i,t+1} - U_{base} \right| \le 0.01 \\ 5 & 0.01 < \left| U_{i,t+1} - U_{base} \right| \le 0.03 \\ 10 & 0.03 < \left| U_{i,t+1} - U_{base} \right| \le 0.05 \\ 50 & \left| U_{i,t+1} - U_{base} \right| > 0.05 \end{cases} \quad (19)$$

In this way, the reward structure will allow the TD3 agent to reduce large voltage violations, while somewhat it will train on the best way to allocate resources towards the most important nodes within constraints, and account for operational priorities and tolerance in VR.

3.5 TD3 network architecture for voltage control

The TD3 algorithm deploys an actor-critic architecture comprising:

- AN $\mu(s \mid \theta^{\mu})$: produces continuous-valued actions (i.e., active and reactive setpoints for each device) given the system state.
- Two CNs $Q_1(s, a \mid \theta^{Q_1})$, $Q_2(s, a \mid \theta^{Q_2})$: estimate the expected cumulative reward from a given state-action pair. To lessen the tendency to overestimate, the lower of the two Q-values is utilized when updating policies.

The critics are trained by lessening the subsequent deficit:

$$L(\theta^{Q_i}) = \mathbb{E}_{s,a,r,s'}[(Q_i(s,a) - y)^2], i = 1,2 \quad (20)$$

Where the target Q-value, represented by y, is calculated as follows:

$$\begin{aligned} y \\ = r + \gamma \\ \cdot min \left(Q_1' \left(s', \mu'(s') \right), Q_2' \left(s', \mu'(s') \right) \right) \end{aligned} \quad (21)$$

Here, μ' and Q_i' are the TA and CNs, and γ is the discount factor. The actor is upgraded utilizing the deterministic policy gradient:

$$\begin{aligned} \nabla_{\theta^{\mu}} J \approx \mathbb{E}_s \left[\nabla_a Q_1(s, a \mid \theta^{Q_1}) \right|_{a=\mu(s)} \\ \cdot \nabla_{\theta^{\mu}} \mu(s \mid \theta^{\mu}) \left] \right. \end{aligned} \quad (22)$$

To stabilize training and prevent premature convergence, TD3 incorporates:

- Delayed policy updates: the AN and TN are updated less frequently than the critics.

- Target policy smoothing: small noise $\mathcal{N}(0,\sigma^2)$ is added to the target action to regularize Q -values:

$$a' = \mu'(s') + clip\ (\mathcal{N}(0,\ \sigma^2), -c, c) \quad (23)$$

This strategy improves robustness against overfitting and variance from stochastic environments.

3.6 Integration of TD3 into the voltage control loop

The trained TD3 agent is implemented into a hierarchical voltage controller framework while preserving the system architecture from the AWDDQN model. Given the absolute number of decision steps t, the agent observes the complete system state s_t (refer to Equation 15), and returns a vector of continuous might control actions (where a_t is a vector of dimension 2 m – i.e., two actions for every m controllable device):

$$a_t = \mu(s_t \mid \theta^{\mu}) = \left[P_{1,t}, Q_{1,t}, \dots, P_{J,t}, Q_{J,t}\right] \quad (24)$$

These outputs are directly interpreted as active and reactive power setpoints for every controllable device, although these actions must be clipped to ensure bounds of Equations (9)-(11) are not violated. The control signals to the vector a_t are saved to either the ADN simulator or the advanced real-time controller, which calculates the next state s_{t+1} based upon the UCT. The start of the next cycle is completed by the agent observing the reward r_t (refer to Equation 18), and storing the experience (s_t, a_t, r_t, s_{t+1}) in the replay buffer, while updating the online network and TN per the TD3 rules - refer to Equations (20)-(23).

3.7 Offline training and online deployment

The TD3-learned voltage control system has two key phases:

1. Offline Training phase: The agent explores the environment using a Gaussian noise process over actions and learns the optimal voltage control policies, collecting them through experience replay until convergence of the average episode reward and decreasing the frequency of voltage violations.
2. Online Control phase: After training is complete, the TD3 agent can determine actions in real time without exploration noise. The AN outputs deterministic control actions that are determined only from observations of the system state. The outputs are continuous and bounded, so they can be used immediately with no extra decoding or quantization required for grid/backup-unstable devices.
3.

3.8 Real-time feasibility and constraint enforcement

To maintain operational safety, control outputs from the TD3 actor are bounded at runtime by device-specific constraints:

$$a_{j,t} \in \left[P_{j,t}^{min}, P_{j,t}^{max}\right] or \left[Q_{j,t}^{min}, Q_{j,t}^{max}\right] \quad (25)$$

These bounds reflect:

- Schedulable EV capacities as defined by Equations (10)-(14),
- Static VAR limits of reactive units.

These constraints are enforced by clipping the outputs during both action selection and training updates. Additionally, by using a design with penalty-based reward, agents learn to give voltage sensitive nodes priority adaptively.

3.9 Hierarchical EVA coordination

Charging for EVs is controlled in a two-level hierarchy:

- Upper Level: The TD3 agent figures out optimal total power for each EVA (aggregator) by taking into consideration the system state and schedulable availability.
- Lower Level: Each EVA takes the total power and distributes the power to the individual EVs based on SOC, arrival/departure windows, and ability to charge, with a weighted distribution.

$$P_{d,t} = \begin{cases} \dfrac{O_{j,t}}{SCP_{P,t}} \cdot P_{j,t}^{max}, & if\ O_{j,t} \geq 0 \\ \dfrac{O_{j,t}}{SDP_{d,t}} \cdot P_{j,t}^{min}, & if\ O_{j,t} < 0 \end{cases} \quad (26)$$

This design allows fulfillment of the initial scheduling guarantees while still ensuring feasibility in implementation and benefiting from TD3's fine-grained continuous control.

4 Simulation and result analysis

4.1 Test system and parameter settings

To assess how well the novel TD3-based voltage control method works, simulations are performed with two modified benchmark distribution networks, namely the IEEE-33 and IEEE-123 benchmark systems. Both systems have DERs, EVA, and reactive compensation units. The revised connection points with power rating settings are shown in Table I. Relative to the baseline condition identified for each system, the assumption of PV and WT generator connected at buses 10, 26 (IEEE-33) and buses 17, 63 (IEEE-123) with total capacity of 2 MW for PV and 1.2 MW for WT, while the EVA units are established at buses 21, 29 (IEEE-33) and buses 55, 102 (IEEE-123) with reactive units at buses 14 and 31 in both systems. The reactive power resources can operate within limits of [-0.6 Mvar, 0.6 Mvar].

Updated output profiles for the PV and WT units with ±15% random variations to mimic uncertainty are shown in Figure 1(a). The base load values are 4 MW and 6 MW for the IEEE-33 and IEEE-123 systems, respectively, along with ±10% load perturbations, as illustrated in Figure 1(b). The EVSC profiles are given in Figure 1(c), accounting for 120 EVs per aggregator, with 45kWh batteries, and charging/discharging power limited to 11kW. Charging efficiencies are kept at 0.97, and it is assumed that excited about charging EVs typically charge during the afternoon period at a workplace and continue charging again at their residential points during the night if not fulfilled.

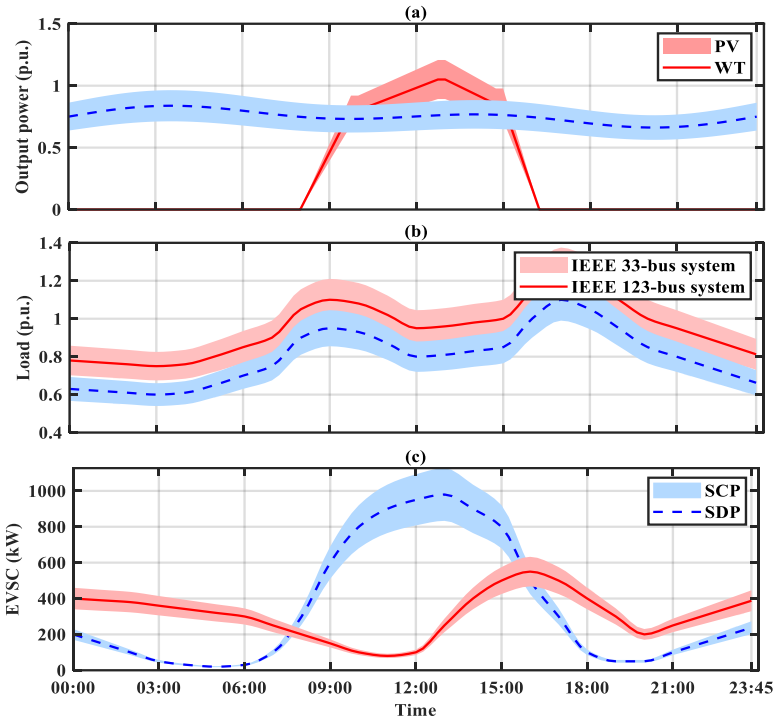


Figure 1: (a) Output profiles of PV and WT units. (b) load profiles with ±10% uncertainty. (c) EVSC schedules

Table 1: Settings of DERs and adjustable resources

Testing System	Connected Node	Resource Type	Power Setting
IEEE-33	10, 26	PV Generators	2.0 MW
	15	WT Units	1.2 MW
	21, 29	EV Aggregators (EVAs)	Shown in Figure 1(c)
	14, 31	Reactive Power Resources	[-0.6 Mvar, 0.6 Mvar]
IEEE-123	17, 63	PV Generators	2.0 MW
	30	WT Units	1.5 MW
	55, 102	EV Aggregators (EVAs)	Shown in Figure 1(c)
	45, 88	Reactive Power Resources	[-0.6 Mvar, 0.6 Mvar]

In the IEEE-33 system, the SS was defined as 33 node voltages plus 2 EVAs with 4 variables each (SCP, SDP, SCC, SDC), and 3 adjustable units, resulting in 44 input dimensions. The TD3 agent takes continuous ASs, which are bounded by the limits of the devices. However, a discretized mapping is kept for interpretability purposes to maintain consistent representations. Each actuator could be represented with 10 levels of granularity; in effect, 1000 equivalent action vectors can be visualized. With the same method, the IEEE-123 system is considered to consist of 123 voltage nodes, 4 EVAs, and 6 adjustable resources, which leads to a 151-dimensional state input to the neural networks.

4.2 Training process analysis

The TD3 agent was trained with synthetic data that was generated with Monte Carlo Sampling for 500 days (with 15% randomness applied to the DER and load values. Each episode was at a 15-minute granularity for a 24-hour time frame consisting of 96-time steps. The unseen data sets are important to validate generalization. The AC and twin CNs consist of {128, 128, 64} neurons for the IEEE-33 system and a deeper architecture of {256, 256, 128} neurons for the IEEE-123 system. Important TD3 hyperparameters for this training run were a policy update delay of 2 steps, target smoothing coefficient for $\tau = 0.005$, discounting factor for $\gamma = 0.98$, and action noise standard deviation of $\sigma = 0.1$ (refer to Table II).

Table 2: TD3 training hyperparameters for the simulation environment

Parameter	IEEE-33 system	IEEE-123 system
Activation function	ReLU	ReLU
Discount factor γ	0.98	0.98
Learning rate η	0.0005	0.0005
Policy updates delayed	2	2

Replay buffer size M	50,000	100,000
Mini-batch size B	256	256
Target smoothing factor τ	0.005	0.005
Number of episodes I	2500	7000
Exploration noise std. σ	0.1	0.1
Hidden layers (Actor/Critic)	{128, 128, 64}	{256, 256, 128}

The training process demonstrates the superior ability of the TD3 algorithm, compared to the DDPG and DQN-based baselines, to converge more consistently throughout training. Observed episode rewards started to consistently rise and stabilize at approximately 2500 episodes in the case of the IEEE-33 system and 7000 episodes in the IEEE-123 setup. Figure 2 depicts the convergence process, and it demonstrates how the TD3 agent received the highest cumulative rewards across nearly all episodes. In the early parts of training, there was a lot of variation in the rewards received by the TD3 agent. This is because of the exploratory noise that stimulates exploration, and this variation in rewards decreases throughout training as policy variation decreases. The double critic mechanism provided an additional layer of stabilisation to the learning, and the delayed updates of the actor were other contributing factors that aided learning stability and with alleviating the ability to overestimate value.

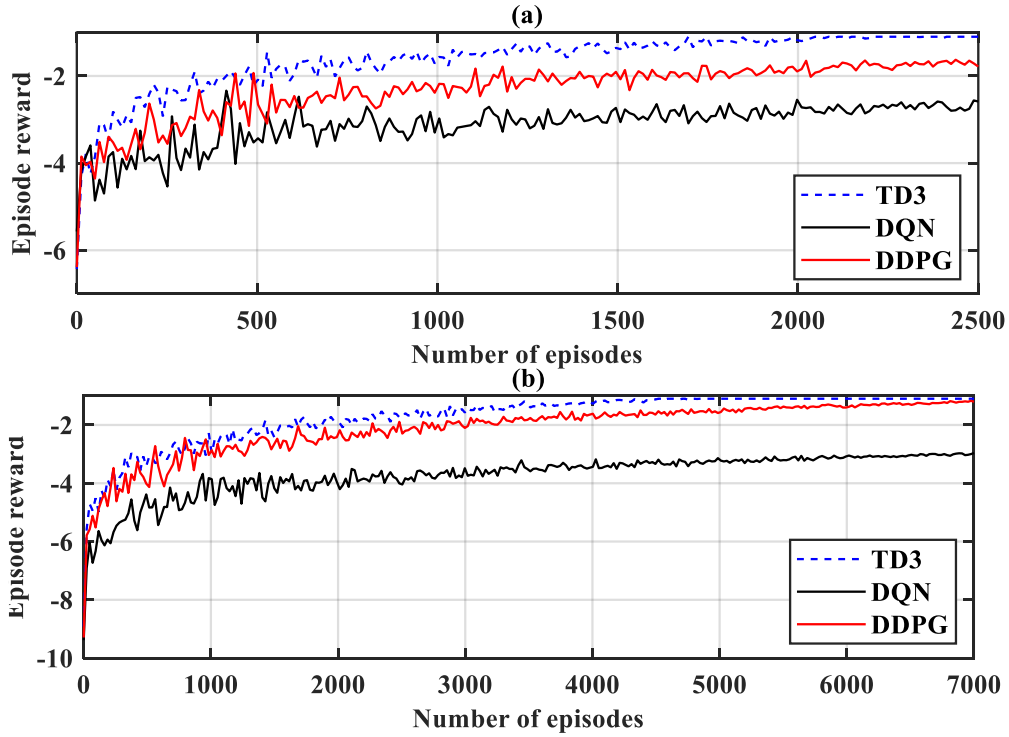


Figure 2: Episode reward progression for TD3, DDPG, and DQN-based agents. a) IEEE-33, b) IEEE-123

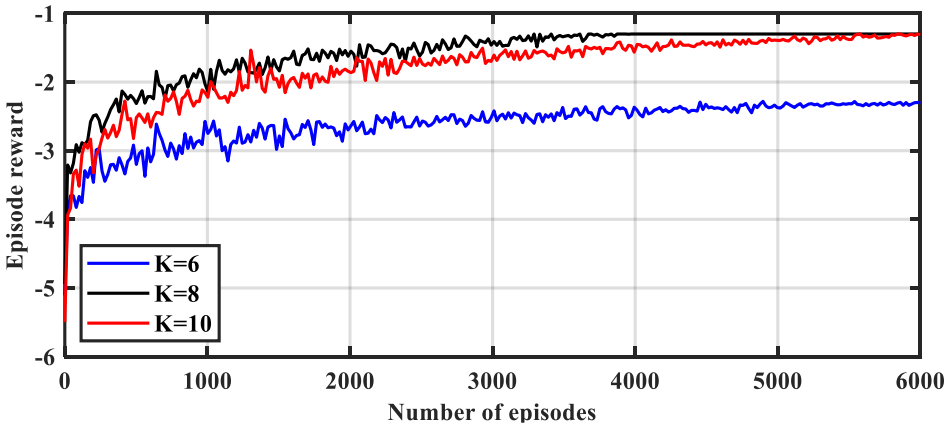


Figure 3: Training reward curves for TD3 under different action discretization resolutions

4.3 Effect of K value on training

While TD3 uses a continuous AS, the discretization granularity is evaluated to provide a comparative assessment and visualization. The parameter represented by K indicates the number of discrete control levels per actuator (or action resolution). For K = {6,8,10}, this translates to approximately action dimensions of 125, 1000, and 3375 for the IEEE-33 system. Figure 3 illustrates the training convergence for the different K's; smaller K values have higher reward convergence, suggested by quicker learning, which is a result of less action exploration complexity; however, smaller K also causes suboptimal final rewards as controls are too coarse. K = 8 allowed for convergence towards the end of the training period, where the time and the accuracy were balanced at the episode number of about 2500. Even though K = 10, when K increased, there was a better reward, but this was at the expense of learning speed and greater variance. K = 8 will be adopted for final deployment as this value represents the best tradeoff.

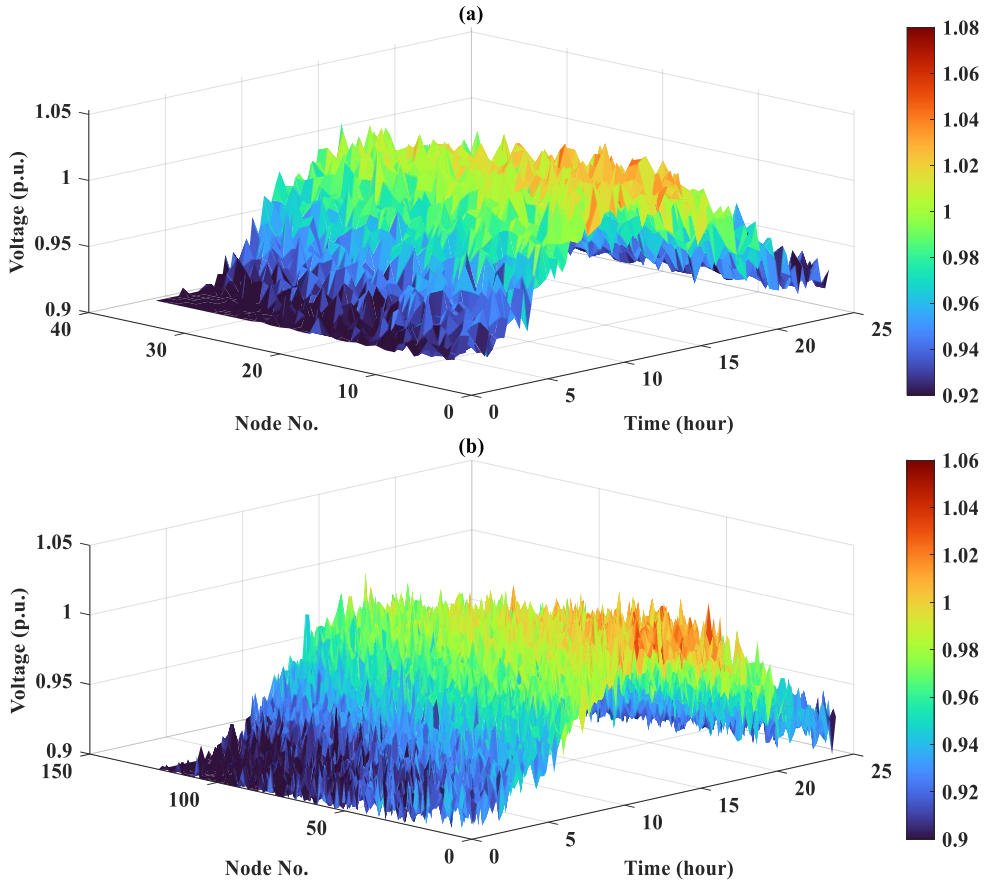


Figure 4: Uncontrolled voltage profiles across nodes over 24 hours in both test systems. a) IEEE-33, b) IEEE-123

4.4 Simulation results and discussions

Simulation results of the trained TD3 voltage control agent in control simulations show it is performing better in both test systems. Uncontrolled voltage profiles under the three test scenarios are provided in Figure 4. Clear signs of overvoltage during midday (with the peaks of PV) and undervoltage during the night (due to peak demand and lack of generation) are seen from the uncontrolled voltage profiles. For example, Node 26 in the IEEE-33 test system manages to exceed 1.07 p.u. at around midday. In the same test system, Node 33 demonstrates a loss of generation at considerably too low a voltage, with a measured voltage of around 0.93 p.u.

The controlled voltage results for traditional MINLP, DQN, and TD3 are presented in Figure 5. The TD3 agent accomplishes all nodes staying within [0.95, 1.05] p.u., in addition to achieving smoother voltage profiles. In the case of the IEEE-123 network, TD3 achieves a maximum deviation of ±0.025 p.u., which is outstanding, and a great improvement from DDQN.

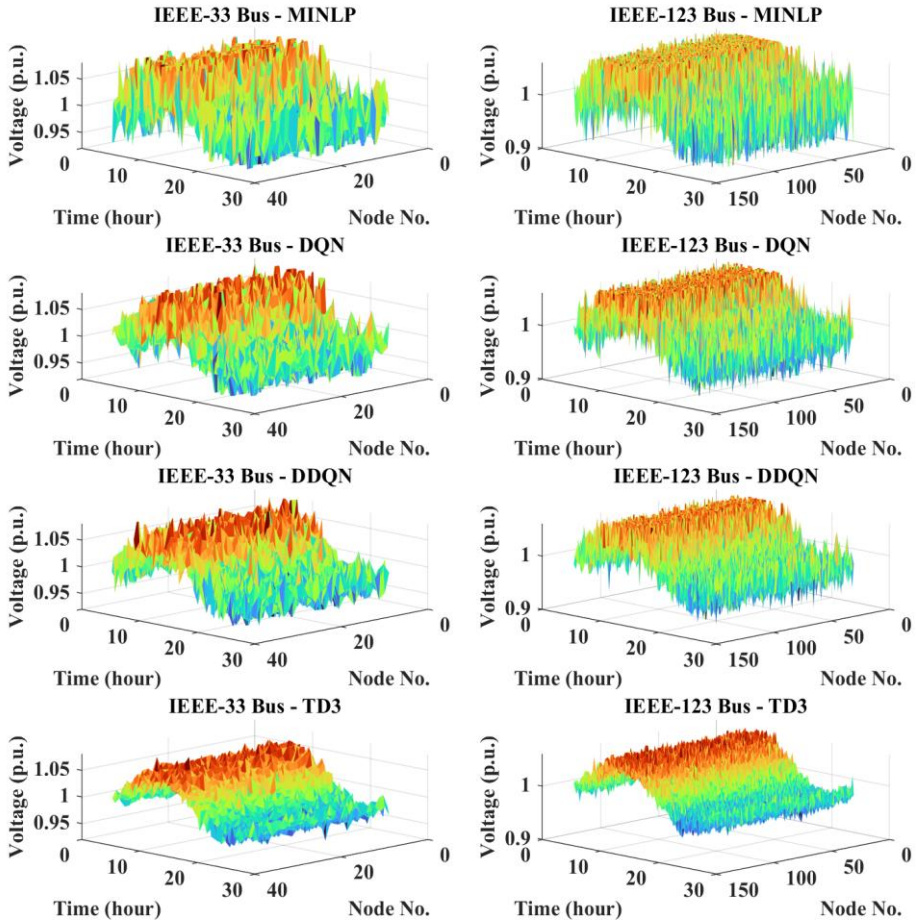


Figure 5: Controlled voltage profiles of IEEE-33 and IEEE-123 under MINLP, DQN, DDQN, and TD3 algorithms

In-depth actuator outputs (see Figure 6) depict that TD3 uses reactive power units at mid-day to achieve overvoltage mitigation effectively, while at night it employs appropriate control of EV discharging to facilitate recovery to a usable voltage. Table III summarizes the objective values, voltage ranges, and computation times across all control schemes. In both systems, TD3 yields the lowest voltage variance with only 0.1% of the computation times of MINLP, making it appropriate for real-time implementations.

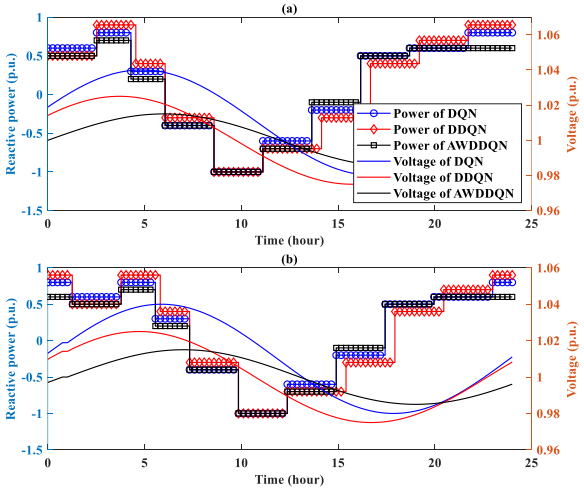


Figure 6: Real-time output actions of reactive units and voltage tracking under TD3 and baseline methods

Table 3: Voltage control performance comparison

Testing System	Case	Objective Function Value	Voltage Range (p.u.)	Calculation Time (s)
IEEE-33	Initial state	0.0256	[0.928, 1.071]	–
	MINLP	0.0089	[0.960, 1.050]	215.63
	DQN	0.0105	[0.947, 1.058]	0.18
	DDQN	0.0081	[0.950, 1.049]	0.24
	TD3 (Proposed)	0.0067	[0.957, 1.018]	0.26
IEEE-123	Initial state	0.0349	[0.920, 1.056]	–
	MINLP	0.0045	[0.964, 1.044]	1030.12
	DQN	0.0059	[0.945, 1.047]	0.60
	DDQN	0.0031	[0.973, 1.043]	0.94
	TD3 (Proposed)	0.0023	[0.976, 1.038]	0.99

The SOC curves for EVA 1 in Figure 7 demonstrate that the TD3 agent coordinated both system-level voltage requirements on the overarching transmission network and EV charging requirements (individually). In respect to the requirements of charging and discharging within the given time steps, nearly all EVs have met their SOC targets, indicating the successful coordinated control of the TD3 agent despite actively participating in grid level, active actions.

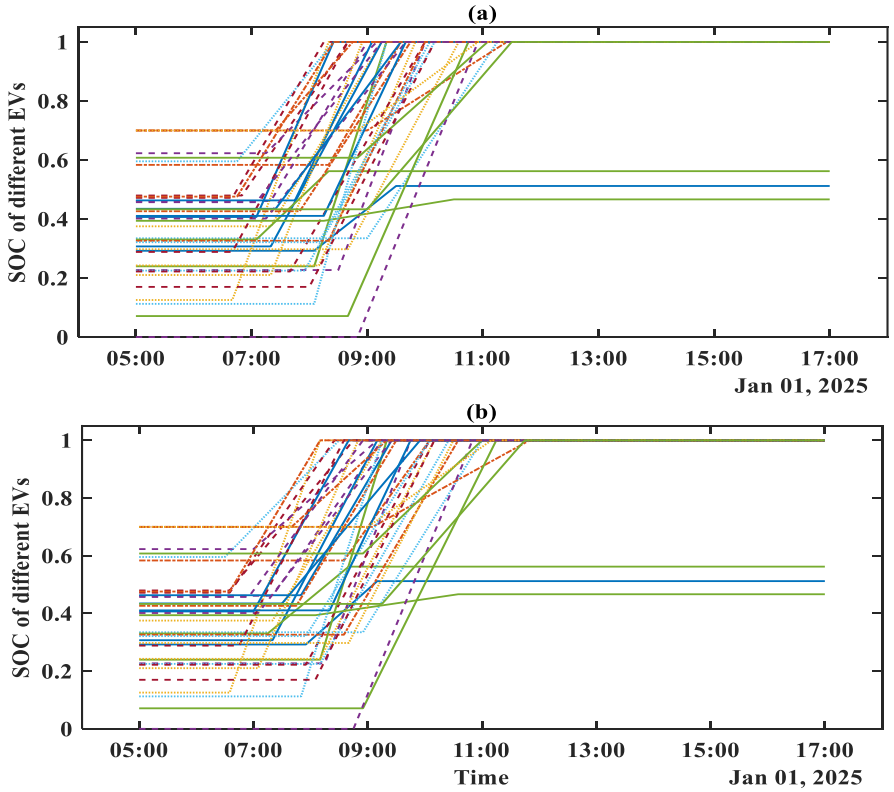


Figure 7: EV SOC trajectories post-dispatch for EVA 1, validating consumer satisfaction and grid compatibility

5 Conclusion

This research proposed a TD3-based smart voltage control framework for ADNs that included CA DRL coupled with system-aware constraints, like EV charging flexibility and renewable energy variability. By utilizing the TD3 algorithm, the proposed controller produces continuous output of power setpoints of adjustable resources, like EVAs and reactive power units. The controller is capable of smooth, fine-grained VR at nodes throughout the network since the voltage controller does not require discretized action sets. Simulations conducted on modified IEEE-33 and IEEE-123 test systems showed that TD3-based controllers were capable of maintaining node voltages within operational limits, despite variation in renewable generation and load. Convergence during training of the algorithm exhibited stability, and it was able to train at an accelerated manner, surpassing previously established RL model convergence rates. The algorithm was robust to random stochastic disturbances in the environment and resulted in a high level of control accuracy while representing a low computational effort. The TD3 algorithm also preserved a user-side EV constraint, leaving enough power capacity to meet EV charging demands while still supporting the distribution grid control needs. The method proposed fulfills the main

research aim of establishing a scalable, stable, and high-resolution voltage control strategy that is implementable in modern distributions with high DERs and EVs. It also exhibited congruence with recent developments of CA RL and confirmed that deterministic actor–critic approaches are relevant for real-time grid applications. The implementation and performance of the method are contingent on proper parameter tuning and good simulation data; and while the simulation results offered improvements, operational deployment must involve realistically promoting constraints of sensors that are noisy, communication that is delayed, and hardware interfaces that cause bottlenecks. Future investigations should look at extending this framework with adaptive exploration, decentralized multi-agent coordination, and hybrid model-based components to enhance explainability and safety.

Nomenclature

Symbol	Definition	Symbol	Definition
s_t	State vector at time t	$P_{d,c,\max}$	Maximum charging power of EV d
a_t	Action vector at time t	$P_{d,d,\max}$	Maximum discharging power of EV d
r_t	Reward at time t	η_c	Charging efficiency of EV d
$U_{n,t}$	Voltage at node n at time t	η_d	Discharging efficiency of EV d
U_{base}	Base/reference voltage (1.0 p.u.)	$\text{SCC}_{d,t}$	Schedulable charging capacity of EV d at time t
	Time step duration	$\text{SCP}_{d,t}$	Schedulable charging power of EV d at time t
T	Dispatch time horizon	$\text{SDC}_{d,t}$	Schedulable discharging capacity of EVd at time t
N	Total number of nodes	$\text{SDP}_{d,t}$	Schedulable discharging power of EV d at time t
$Q_{j,t}$	Reactive power of resource j at time t	$\text{SCP}_{l,t}$	Total schedulable charging power of EVA l at time t
$P_{j,t}$	Active power of resource j at time t	$\text{SDP}_{l,t}$	Total schedulable discharging power of EVA l at time t
$Q_{j,\min}$	Minimum reactive power limit of resource j	$U_{i,t}$	Voltage at node i at time t
$Q_{j,\max}$	Maximum reactive power limit of resource j	λ_i	Penalty factor for node i
$P_{j,t,\min}$	Minimum active power limit of resource j at time t	γ	Discount factor in TD3
$P_{j,t,\max}$	Maximum active power limit of resource j at time t	$\mu(s \mid \theta^{\mu})$	An actor gives an action
C_d	Battery capacity of EV d	$Q_i(s,a \mid \theta^{Q_i})$	Critic estimates the value of that action.
$\text{SOC}_{d,t}$	State of charge of EV d at time t	y	Target Q-value
$\text{SOC}_{d,s}$	Initial SOC of EV d upon arrival	$L(\theta^{Q_i})$	Critic loss function
$\text{SOC}_{d,\min}$	Minimum required SOC of EV d upon departure	θ^{μ}	Parameters of the AN
$t_{d,s}$	Expected arrival time of EV d	θ^{Q_i}	Parameters of the CN
$t_{d,e}$	Expected departure time of EV d		

Acknowledgements

Interim Research Results of the Electrical Control System Design Practice Base under the 2024 Teaching Quality and Teaching Reform Project of Guangzhou College of Applied Science and Technology (Project No.: 2024ZG009)

Interim Research Results of the 2024 Higher Education Research Project under the 14th Five-Year Plan of Guangdong Higher Education Society (Project No.: 24GYB96)

Development of New Energy-Metal Air Battery Emergency Power Supply for Zhaoqing City, Guangdong Province, Science and Technology Innovation Guidance Category Project (Project No. 24122519009103)

Authorship contribution statement

Long Kou: Supervision, Conceptualization, Project administration, Writing-Original draft preparation.
Jiangyan Chen: Software, Methodology.
Yuhua Wang: Validation.
Yilin Liu: Language review.

Data availability

Available upon request.

Conflicts of interest

The authors affirm that they have no competing interests related to the posting of this document.

Author statement

All authors have reviewed and endorsed the manuscript, confirming adherence to the stated authorship criteria. Each author also attests to the integrity and authenticity of the work presented.

Ethical approval

Each author was directly and significantly involved in the work that led to this paper and will publicly stand behind its content.

References

[1] S. Nematshahi, D. Shi, F. Wang, B. Yan, and A. Nair, "Deep reinforcement learning based voltage control revisited," *IET Generation, Transmission & Distribution*, 17(21): 4826–4835, 2023. https://doi.org/10.1049/gtd2.13001

[2] S. M. Abdelkader *et al.*, "Advancements in data-driven voltage control in active distribution networks: A Comprehensive review," *Results in Engineering*, 23: 102741, 2024. https://doi.org/10.1016/j.rineng.2024.102741

[3] Y. Zheng, Y. Song, D. J. Hill, and K. Meng, "Online distributed MPC-based optimal scheduling for EV charging stations in distribution systems," *IEEE Trans Industr Inform*, 15(2): 638–649, 2018. DOI:10.1109/TII.2018.2812755

[4] M. Mazumder and S. Debbarma, "EV charging stations with a provision of V2G and voltage support in a distribution network," *IEEE Syst J*, 15(1): 662–671, 2020. https://doi.org/10.1016/j.ecmx.2025.101138

[5] A. Ahmadian, B. Mohammadi-Ivatloo, and A. Elkamel, "A review on plug-in electric vehicles: Introduction, current status, and load modeling techniques," *Journal of Modern Power Systems and Clean Energy*, 8(3): 412–425, 2020. DOI:10.35833/MPCE.2018.000802

[6] H. Patil and V. N. Kalkhambkar, "Grid integration of electric vehicles for economic benefits: A review," *Journal of Modern Power Systems and Clean Energy*, 9(1): 13–26, 2020. DOI:10.35833/MPCE.2019.000326

[7] X. Sun and J. Qiu, "Hierarchical voltage control strategy in distribution networks considering customized charging navigation of electric vehicles," *IEEE Trans Smart Grid*, 12(6): 4752–4764, 2021. DOI:10.1109/TSG.2021.3094891

[8] Y. Wang, T. John, and B. Xiong, "A two-level coordinated voltage control scheme of electric vehicle chargers in low-voltage distribution networks," *Electric Power Systems Research*, 168: 218–227, 2019. https://doi.org/10.1016/j.epsr.2018.12.005

[9] Y. Liu and H. Liang, "A discounted stochastic multiplayer game approach for vehicle-to-grid voltage regulation," *IEEE Trans Veh Technol*, 68(10): 9647–9659, 2019. https://doi.org/10.1016/j.rineng.2025.106813

[10] J. Hu, C. Ye, Y. Ding, J. Tang, and S. Liu, "A distributed MPC to exploit reactive power V2G for real-time voltage regulation in distribution networks," *IEEE Trans Smart Grid*, 13(1): 576–588, 2021. DOI: 10.1109/TSG.2021.3109453

[11] D. Haoran, Y. Ming, C. Fang, and S. Guozhong, "Reactive power and voltage optimization control approach of the regional power grid based on reinforcement learning theory," *Transactions of China Electrotechnical Society*, 30(12): 408–414, 2015. https://doi.org/10.3390/en17246454

[12] D. Cao *et al.*, "Reinforcement learning and its applications in modern power and energy systems: A review," *Journal of modern power systems and clean energy*, 8(6): 1029–1042, 2020. DOI:10.35833/MPCE.2020.000552

[13] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, 518(7540): 529–533, 2015. DOI:10.1038/nature14236

[14] J. Shi, W. Zhou, N. Zhang, Q. Chen, J. Liu, and Z. Cao, "Deep reinforcement learning algorithm of voltage regulation in distribution network with energy storage system," *Electric Power Construction*, 41(03): 71–78, 2020. https://doi.org/10.1016/j.apenergy.2022.120510

[15] R. Diao, Z. Wang, D. Shi, Q. Chang, J. Duan, and X. Zhang, "Autonomous voltage control for grid operation using deep reinforcement learning," in *2019 IEEE power & energy society general meeting (PESGM)*, IEEE, 2019, 1–5. DOI:10.48550/arXiv.1904.10597

[16] Q. Yang, G. Wang, A. Sadeghi, G. B. Giannakis, and J. Sun, "Two-timescale voltage control in distribution grids using deep reinforcement learning," *IEEE Trans Smart Grid*, 11(3): 2313–2323, 2019. DOI:10.1109/TSG.2019.2951769

[17] X. Sun and J. Qiu, "A customized voltage control strategy for electric vehicles in distribution networks with reinforcement learning method," *IEEE Trans Industr Inform*, 17(10): 6852–6863, 2021. DOI: 10.1109/TII.2021.3050039

[18] B. Zhang, D. Cao, W. Hu, A. M. Y. M. Ghias, and Z. Chen, "Physics-Informed Multi-Agent deep reinforcement learning enabled distributed voltage control for active distribution network using PV inverters," *International Journal of Electrical Power & Energy Systems*, 155: 109641, 2024. https://doi.org/10.1016/j.ijepes.2023.109641

[19] M. Golgol and A. Pal, "High-speed voltage control in active distribution systems with smart inverter coordination and DRL," in *2024 IEEE Power & Energy Society General Meeting*

(PESGM), IEEE, 2024, 1–5. https://doi.org/10.1016/j.epsr.2024.110528

[20] N. Shi, R. Cheng, L. Liu, Z. Wang, Q. Zhang, and M. J. Reno, "Data-driven affinely adjustable robust Volt/VAr control," *IEEE Trans Smart Grid*, 15(1): 247–259, 2023. DOI: 10.1109/TSG.2023.3270112

[21] A. Dutta, S. Ganguly, and C. Kumar, "Three-stage receding horizon-based voltage control and electric vehicle charge scheduling of active distribution networks," *IET Renewable Power Generation*, 18, 4308–4317, 2024. https://doi.org/10.1049/rpg2.12977

[22] C. Ma, W. Xiong, Z. Tang, Z. Li, Y. Xiong, and Q. Wang, "A Hierarchical Voltage Control Strategy for Distribution Networks Using Distributed Energy Storage," *Electronics (Basel)*, 14(9): 1888, 2025. https://doi.org/10.3390/electronics14091888

[23] Y. Zhang, J. Chen, H. Zhao, W. Zhang, W. Jiao, and W. Dai, "Coordinated voltage control of active distribution networks with photovoltaic and power to hydrogen," *IET Energy Systems Integration*, 5(3): 245–260, 2023. https://doi.org/10.1049/esi2.12096

A Hybrid Optimization Approach for Pile Capacity Estimation Using Radial Basis Functions and Time-Dependent Variables

Shanwei Zhang
College of Design, Chongqing College of Finance and Economics, Yongchuan 402160, Chongqing, China
E-mail: 19936052713@163.com

Keywords: geotechnical engineering**,** predictive modeling**,** structural safety**,** soil properties, sensitivity analysis, time-dependent variables

Received: March 3, 2025

Since the pile and surrounding soil interact in complex, nonlinear ways and time-dependent geotechnical parameters play a significant role, accurately estimating the axial bearing capacity of driven piles is still a crucial but difficult part of geotechnical engineering. The new hybrid predictive framework proposed in this study combines two recent metaheuristic optimization algorithms, Leader Harris Hawks Optimization (LHHO) and Graylag Goose Optimization (GGO), with the Radial Basis Function (RBF) model. Through the incorporation of temporal variations, such as changes in soil resistance due to pore water pressure dissipation, that are commonly overlooked in conventional modeling approaches, the main goal is to improve the predictive accuracy and generalization capability of pile capacity models. By combining time-dependent variables with a correlation-based feature selection mechanism, this work is novel in that it enables the identification of the most important input parameters and the removal of redundant or unnecessary features. In comparison to actual pile load test data, the resulting hybrid models, known as RBGG and RBLH, perform better. During training, the RBLH model demonstrated high accuracy and reliability with an R^2 value of 0.986 and an RMSE of 183.680. In addition, sensitivity analysis showed that the most important factors affecting pile capacity are unit weight, soil cohesiveness, and pile length. With major ramifications for enhancing safety, decreasing uncertainty in construction procedures, and optimizing foundation design, the suggested methodology provides engineers with a strong, data-driven tool.

Povzetek: Prispevek obravnava nosilnost pilotov v povezavi z nelinearnimi interakcijami zemljine in časa. Predlaga hibridni pristop z radialnimi baznimi funkcijami (RBF), optimiziranimi z algoritmoma LHHO in GGO, ki vključuje časovno odvisne geotehnične spremenljivke.

1 Introduction

Pile foundations are structural components typically used when soil settlement is a critical consideration during the design phase or when surface soils are weak. In such cases, the structural load must be transferred to deeper soil layers. For geotechnical engineers, determining the ultimate bearing capacity of pile foundations is crucial, especially when considering the pile's role in load transmission. Some researchers have suggested that pile carrying capacity is a time-dependent variable, showing an upward trend after a certain period [1], [2], [3]. The geotechnical phenomenon known as 'pile setup' refers to the gradual increase in the ultimate bearing capacity of pile foundations. Pile setup is thought to be caused by the dissipation of excess pore water pressure (EPWP) produced during pile installation [4].

Additionally, it is generally acknowledged that this phenomenon progresses through three primary stages: aging, uniform EPWP dissipation, and non-uniform EPWP dissipation [5]. Several studies have found that setup significantly impacts side resistance, whereas relaxation shows little change or even a decrease in tip resistance [6], [7], [8]. Researchers have long been interested in predicting the time-dependent bearing capacity of pile foundations. Furthermore, considering pile setup in the design process could make the procedure more cost-effective.

Numerous studies predicting pile configuration using analytical or numerical models have been reported [9]. Skov and Denver's work [10] which aimed to determine an equation for predicting pile setup, is one of the most well-known studies in this field. To achieve this, several geotechnical characteristics were used to revise the setup equation. They also introduced a valid variable, known as the setup parameter (A), and presented a semi-empirical equation. Other researchers built upon this groundbreaking work. A study by Haque and Abu-Farsakh, for instance, examined the use of a nonlinear multivariable regression model to predict pile configuration. Although research employing this combination of methods has provided useful equations, pile placement remains a challenging problem due to the complex soil–pile interaction. As a result, regression analysis and analytical techniques may not be sufficient for making accurate predictions [11]. The effective use of

intelligent algorithms to model complex issues in geotechnical and civil engineering has been demonstrated in numerous studies in recent years [12], [13], [14]. Many researchers have highlighted the usefulness of these methods in predicting pile-related issues, such as lateral deflection, settlement, and pile capacity. Lee and Lee [15] examined the use of artificial neural networks (ANNs) for predicting pile bearing capacity. The developed model was validated using the results of in situ pile load tests. Ultimately, the error back-propagation neural network demonstrated strong performance, with a maximum prediction error below 25%. Shahin [16] modeled the axial capacity of pile foundations using intelligent computing. For this, 174 data points were used in an ANN approach to predict the axial capacity of driven piles and drilled shafts. The findings indicated that the ANN performed satisfactorily on the validation datasets for driven piles and drilled shafts, with correlation coefficients of 0.85 and 0.97, respectively. To evaluate pile carrying capacity, Samui [17] explored the use of the support vector machine, a powerful machine learning technique. For this, three inputs were considered: the number of bowls, mean normal stress, and penetration depth ratio. Ultimately, the developed model accurately predicted pile carrying capacity, using assessment criteria such as the correlation coefficient. In a different research, Momeni et al. [18] employed an ANN-based prediction model optimized with a genetic algorithm to predict pile carrying capacity, using data from 50 dynamic load tests. With a correlation coefficient of 0.99, the final data showed that the generated model accurately predicted the target value, which was very close to the actual value. Using regression and EPR-MOGA models, Alzabeebee et al. [19] investigated the relationship between internal friction angle and soil characteristics. The EPR-MOGA model demonstrated its efficacy in capturing intricate relationships in fine-grained soils by achieving high precision (R = 0.85–0.92), whereas linear regression showed moderate accuracy (R = 0.16–0.67). Kumar et al. [20] applied advanced ML models, including XGBoost, RF, GBM, and deep learning, to predict the bearing capacity of PGPN piles using 81 static load tests. XGBoost ($R^2 = 0.91$) and RF showed the best performance, while GBM underperformed. SHAP analysis and Taylor diagrams supported the model evaluation.

Pile foundations are crucial in modern geotechnical engineering, particularly when surface soils cannot support heavy structural loads. Traditional approaches for estimating pile bearing capacity rely on analytical or numerical models, which often fail to provide accurate estimations due to the complexities of soil-pile interaction and time-dependent soil behavior. This research introduces a novel framework that leverages advanced machine learning techniques, specifically the Radial Basis Function (RBF) model. This is coupled with state-of-the-art optimization algorithms, including Greylag Goose Optimization (GGO) and Leader Harris Hawks Optimization (LHHO). This unique combination improves prediction accuracy and reliability, addressing gaps in current methodologies. The novelty of this research lies in focusing on the inclusion of time-dependent variables in pile capacity prediction, which have often been neglected in conventional studies. While previous models have focused on static parameters like cohesion, friction, and soil unit weight, this work examines the dynamic development of pile capacity due to processes like pile setup, driven by EPWP dissipation. By incorporating time-dependent parameters such as average pile-soil friction and changes in soil resistance due to pore water pressure dissipation, this research develops a more comprehensive and adaptive predictive framework. Pile length, while not time-varying, is also included as a key geometric input in the model. In this study, the RBF model served as a tool to capture nonlinear relationships between input parameters and pile capacity. Model parameter fine-tuning was achieved using optimization algorithms such as GGO and LHHO, enhancing the model's accuracy. These optimization algorithms are inspired by the collective intelligence of natural systems, which self-organize and develop efficient search strategies to identify global optima in complex, multi-dimensional datasets. By combining these models and optimization techniques, the research achieves unprecedented precision in capturing the interaction between soil and pile parameters. The sensitivity analysis part of the paper shows the most influential parameters concerning the average cohesion, pile length, and soil specific weight on the capacity of the pile. In this way, engineers will be able to focus on the most relevant variables at the design stage in order to optimize material use with regard to cost. This research will add to the benefit of geotechnical engineering by providing a sound and data-based method for pile bearing capacity prediction. Its applications go beyond academia to the provision of practical tools for industry professionals to create more economical and safer designs of foundations, especially on poor soil classes or projects involved in long-term performance monitoring.

The selection of GGO and LHHO was motivated by their demonstrated performance in solving high-dimensional, nonlinear optimization problems with complex search landscapes. Both algorithms have gained attention in recent literature for their strong balance between global exploration and local exploitation, which is critical for effectively tuning the spread and weights in the RBF model. Specifically, LHHO offers enhanced adaptive leadership mechanisms that improve convergence speed, while GGO mimics collective geese flight behavior, which promotes diversity in the solution population and reduces the chance of local optima entrapment. Compared to traditional algorithms like Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), both GGO and LHHO have shown lower sensitivity to parameter initialization and better generalization.

2 Dataset

The dataset comprises 229 pile load test records sourced from [21], [22], [23], [24], [25]. Each instance includes geotechnical and pile-related features, such as unit weight, cohesion, angle of internal friction, pile length, and time-

dependent parameters. Prior to model training, the dataset was carefully examined for missing values and outliers. No missing or anomalous values were detected. All numerical input features were normalized using Min-Max scaling to ensure uniform data ranges and improve model convergence.

The statistical characteristics of a few important input factors that affect the pile foundations' time-dependent bearing capacity are shown in Table 1. It gives the input indicators' lowest, maximum, average, and standard deviation values. The variability and character of the geotechnical parameters in this research study may be evaluated with the help of the following table.

- **Average Cohesion (0 to 475 kPa, Avg: 89.487 kPa):** Indicates how soil stickiness affects side friction (0 to 475 kPa, Avg: 89.487 kPa). Significant variation suggests a variety of soil types, which affect frictional resistance and, in turn, pile capacity.
- **Average Friction Angle (0 to 39 degrees, Avg: 19.554 degrees):** The shear strength of granular soils is influenced by the average friction angle, which ranges from 0 to 39 degrees and averages 19.554 degrees. Significant variation points to various soil stratifications that influence capacity variations over time.
- **Average Soil Specific Weight (0 to 29.54 kN/m³, Avg: 6.913 kN/m³):** This value influences pile bearing capacity and effective stress. When forecasting capacity, variability makes it evident that variations in pore water pressure are crucial.
- **Average Pile-Soil Friction (0 to 41 kPa, Avg: 14.955 kPa):** The pile-soil interaction is reflected in the average pile-soil friction, which ranges from 0 to 41 kPa and averages 14.955 kPa. Side resistance rises with high friction, which is crucial for sustained capacity improvements.
- **Flap Number (0 to 2291, Avg: 187.338):** This represents the inductive resistance of the soil (0 to 2291, average: 187.338). High variability lends credence to the possibility of a diverse foundation. High values indicate increased tip resistance.
- **Pile Area (0.006 to 34.02 m², Avg: 2.620 m²):** The load-bearing capability is directly impacted by this. In contrast to other factors, this trait has less of an association with time-dependent fluctuations.
- **Pile Length (1.000 to 207.00 m, Avg: 20.131 m):** Longer piles have deeper soil layers, which raises the possibility for tip bearing and side resistance when calculating capacity.
- **Pile Capacity (Output: 36.750 to 6789.00 kN, Avg: 2433.239 kN):** Because of the wide variations in circumstances, the pile capacity exhibits substantial fluctuation. A sensitivity study shows that cohesion, friction angle, and pile length are the main factors influencing the capacity projections.

All things considered, this analysis of Table 1 highlights the importance of including these factors in prediction models. Due to the observed variability, advanced optimization techniques are necessary to enhance the accuracy and reliability of pile capacity predictions, which are crucial for designing optimal pile foundations.

Table 1: The statistical properties of the input variable pile capacity

Variables		Indicators			
	Category	Min	Max	Avg	St. Dev.
Average Cohesion	Input	0.000	475.0	89.487	82.164
Average Friction	Input	0.000	39.0	19.554	15.500
Average soil Specific weight	Input	0.000	29.54	6.913	5.663
Average Pile-Soil Friction	Input	0.000	41.00	14.955	13.771
Flap Number	Input	0.000	2291.0	187.34	429.03
Pile Area	Input	0.006	34.02	2.620	5.246
Pile Length	Input	1.000	207.0	20.131	15.688
Pile Capacity	Output	36.750	6789.0	2433.24	1400.32

Fig. 1 presents a grouped 3D bar plot that illustrates the relationship between the input variables—Average Cohesion, Average Friction, Average Soil Specific Weight, Average Pile-Soil Friction, Flap Number, Pile Area, and Pile Length—and the output variable, Pile Capacity. This visualization demonstrates how changes in these key geotechnical parameters affect the predicted pile bearing capacity. It provides an overview of how these parameters influence the output, offering the opportunity for an in-depth analysis of the interactions between soil properties and pile characteristics, and their combined impact on the load-bearing capacity of pile foundations.

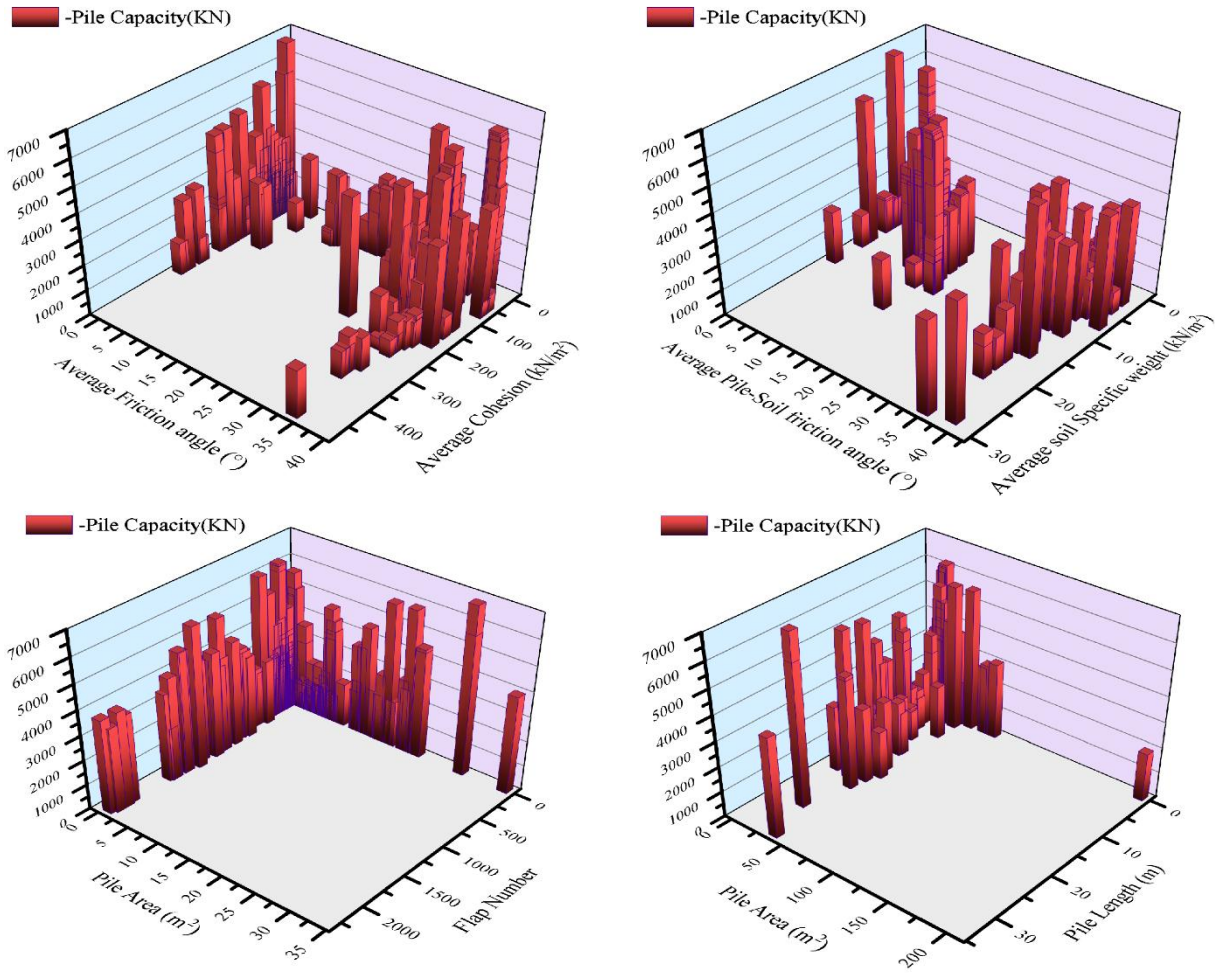


Figure 1: The grouped 3D bar plot displays the relationship between input and output

Fig. 2 shows the feature selection analysis of the dataset using important input variables with regard to the prediction of pile capacity. This indicates the percentage of contribution of every feature towards the model's performance. The most important feature is Pile Length, which has a score of 0.609; this is the most important factor in predicting the capacity of the pile. Flap Number is next, with the value 0.458, indicating a significant role in the variation of soil penetration resistance in determining the pile capacity. Other contributing variables to the model include Average Soil Specific Weight at 0.234, Pile Area at 0.145, and Average Friction at 0.137. The features with the least influence on predictive accuracy are Average Cohesion (0.135) and Average Pile-Soil Friction (0.016). This feature selection analysis helps in prioritizing the most influential factors in the development of models for pile capacity prediction, guiding further modeling and optimization efforts.

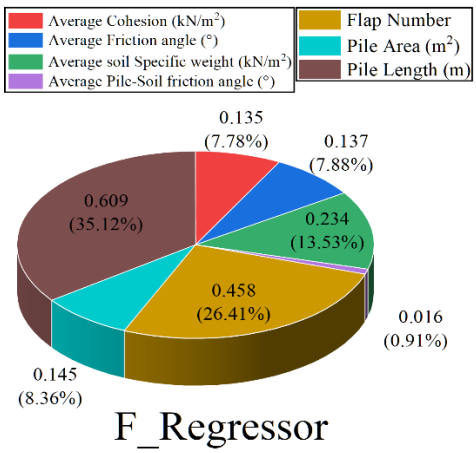


Figure 2: The feature selection analysis of dataset

3 Machine learning models

3.1 Radial basis functions (RBF)

There are several functions that may be utilized as the RBF [26], [27], [28].

$$\varphi(r) = e^{-\frac{r^2}{2\sigma^2}}, \quad Gaussian \tag{1}$$

$$\varphi(r) = \frac{1}{(\sigma^2 + r^2)^{\alpha}}, \quad \alpha > 0, \tag{2}$$

$$\varphi(r) = (\sigma^2 + r^2)^{\beta}, \quad 0 < \beta < 1, \tag{3}$$

$$\varphi(r) = r, \quad liner, \tag{4}$$

$$\varphi(r) = r^2 In(r), \quad thin\ plate\ spline, \tag{5}$$

$$\varphi(r) = \frac{1}{1 + e^{\left(\frac{r}{\sigma^2}\right) - \theta}}, \quad logestic\ function, \tag{6}$$

where σ in Eqs (1), (2), (3), and (6) controls the interpolating function's smoothness, θ in Eq. (6) is an adjustable bias, and $r > 0$ indicates the distance between a data point $\vec{x}$ and a center c. The RBF transforms into Hardy's multiquadric function, which is widely employed in surface interpolation with excellent results, when β in Eq. (3) has the value of $\frac{1}{2}$. $\emptyset r$ is appropriate for DSP implementation when α in Eq. (2) is unity [29].

Localized RBFs with the characteristic that $\emptyset r \to 0$ as $r \to \infty$ are among these RBFs; they are Eqs (1), (2), and (6). Physiologically, cortical cells have receptive fields that resemble Gaussian curves. For this reason, the RBF is usually used as the Gaussian. The Gaussian has a positive density. It is driven by the principles of kernel density estimation and kernel regression. The Gaussian is the best basis function in the least-squares (LS) sense [30] for fitting data with normally distributed noise in the inputs. The only factorizable RBF is the Gaussian, which is a desired feature for the RBF network's hardware implementation.

Selected from a curve-fitting standpoint [31], the thin-plate spline function (Eq. 5) is a popular Radial Basis Function (RBF) for universal approximation. When fitting a surface based on a set of points and incorporating a roughness penalty [32], the thin-plate spline provides the solution. Over the range of $r \in (0, 1)$, it is negative and diverges at infinity. However, the estimated function must only be defined within a predetermined range for training purposes. A little amount of empirical data indicates that in high-dimensional environments, the thin-plate spline provides a superior fit to the data.

By choosing the standard deviation σ in the Gaussian Eq. (1) as two distinct positive values, σ_{-} for $x < 0$ and σ_{+} for $x > 0$ [33], a pseudo-Gaussian function is introduced in the one-dimensional space. The pseudo-Gaussian function in each dimension is multiplied to extend this function to the multiple-dimensional space. Because of its radial asymmetry, the pseudo-Gaussian function is not technically an RBF; yet, this gives the hidden units more freedom in terms of function approximation [34].

It is most challenging and inefficient to use localized RBFs to approximate functions with segments that have values that are almost constant. This issue may be resolved by using the sigmoidal RBF, which is a composite of a collection of sigmoidal functions [35].

$$\varphi(r) = \frac{1}{1 + e^{-\beta[(x-c)+\theta]}} - \frac{1}{1 + e^{-\beta[(x-c)-\theta]}}, \tag{7}$$

Both θ and $\beta > 0$. The radial symmetry of $\varphi(x)$ reaches its greatest at c. The function's breadth is controlled by θ, while its steepness is controlled by β. If $\beta \times \theta$ is big, the form of $\varphi(x)$ is roughly rectangular, or more precisely, soft trapezoidal. For tiny β and θ, it has a bell-shaped form. By multiplying the appropriate function in each dimension, $\varphi(x)$ may be extended for the n-dimensional approximation. By including a compensatory term in the product term $\varphi_i(x_i)$, $\varphi(\vec{x})$ may be changed to allow for constant values of the intended output and prevent the kernel functions from being diminished [36]. Using the raised-cosine function as a one-dimensional RBF is another method. A constant function can be precisely represented by the raised-cosine RBF using two terms. It is possible to generalize this RBF to n dimensions [37]. By appropriately restricting certain parameters, various well-known fuzzy membership functions can accomplish the same goal [38].

The widely used Gaussian RBF has a circular form. To approximate a functional behavior with distinct noncircular properties, a large number of RBF nodes could be needed. A directed graph-based technique is used in [39] to incorporate direction-dependent scaling, shaping, and rotation of Gaussian RBFs for optimal trend sensing with minimal parameter representations for function approximation, therefore reducing the size of the RBF network.

3.2 Greylag goose optimization (GGO)

The study on environmental cleanup, and specifically noise reduction, shows how to effectively improve the BSS with GGO's assistance [40], [41], [42]. This section outlines the full suggestion on how GGO may optimize LSTM hyperparameters and adjust BSS settings.

3.2.1 GGO algorithm

The first step of the GGO algorithm is to create individuals who come up with potential solutions to the problem at random. These potential answers are represented by each individual. The population of the group, n, is represented by the term GGO $(X_i, i = 1,2, \ldots, n)$, where an integer represents the population's size [43], [44].

This illustration perfectly conveys how GGO moves through the solution space, finding a balance between exploitation (improving existing solutions) and exploration (looking for new, maybe better ones). The dynamic depiction of GGO's iterative process in the image highlights how the algorithm combines exploration and exploitation tactics in the best possible way to provide effective and efficient solutions [45].

3.2.2 Exploration operation

The GGO algorithm's exploration process is crucial; it involves searching the search space for the likely best solution, which causes GGO to steer toward the global optima and avoid local optima [46].

Getting to the Best Answer: This tactic lowers the likelihood of becoming locked in local optimality by using explorers' geese, who actively seek out excellent spots close to the ones that are already there. During iterations, the GGO method uses the following equations to update vectors A and C, which are represented as $A = 2a \cdot r_1 - a$ and $C = 2.r_2$, respectively. From 2 to 0, the parameter a change linearly:

$$X(t+1) = X^*(t) - A \cdot |C.X^*(t) - X(t)| \quad (8)$$

In this case, r_1 and r_2 are randomly fluctuating values inside [0,1], $X(t)$ indicates the position of an agent at iteration t, and $X^*(t)$ indicates the position of the best solution (leader).

The method uses the following formula to improve exploration even further, taking into account three randomly selected search agents (paddings), represented by the symbols $X_{"Paddle"1}$, $X_{"Paddle"2}$, and $X_{"Paddle"3}$:

$$\begin{aligned} X(t+1) = {} & w_1 X_{"Paddle"1} \\ & + z w_2 (X_{"Paddle"2} \\ & - X_{"Paddle"3}) \\ & + (1-z) w_3 (X \\ & - X_{"Paddle"1}) \end{aligned} \quad (9)$$

In this case, z drops exponentially in accordance with $z = 1 - (\frac{t}{t_{max}})^2$, where t is the number of iterations and tmax is the maximum number of iterations, whereas w_1, w_2, and w_3 update inside [0,2].

The values of a and vector A are decreased when $r_3 > 0.5$ in the second updating procedure:

$$\begin{aligned} X(t+1) = {} & w_4 |X^*(t) \\ & - X(t)| e^{b\ell} \cos(2\pi\ell) \\ & + 2 w_1 (r_4 + r_5) X^*(t) \end{aligned} \quad (10)$$

Here, w_4 updates within [0,2], r_4 and r_5 update within [0,1], b is a constant, and ℓ is a random number in $[-1,1]$.

3.2.3 Exploitation operation

The exploitative component of the GGO algorithm focuses on refining previously found solutions. After each cycle, GGO identifies the entity with the highest fitness and assigns it the appropriate credit. Two distinct strategies, outlined below, are used to achieve the exploitation goals, and they work together to enhance the overall solution quality.

Getting to the Best Answer: Under the direction of three sentry solutions ($X_{"Sentry1"}$, $X_{"Sentry2"}$, and $X_{"Sentry3"}$), the algorithm uses the following equation to direct people ($X_{"NonSentry"}$) toward the predicted position of the prey:

$$\begin{aligned} X_1 &= X_{"Sentry1"} - A_1 \cdot |C_1 \cdot X_{"Sentry1"} - X|, \\ X_2 &= X_{"Sentry2"} - A_2 \cdot |C_1 \cdot X_{"Sentry2"} - X|, \\ X_3 &= X_{"Sentry3"} - A_3 \cdot |C_1 \cdot X_{"Sentry3"} - X|, \end{aligned} \quad (11)$$

where C_1, C_2, and C_3 are calculated as $C = 2r_2$, and A_1, A_2, and A_3 are calculated as $A = 2a \cdot r_1 - a$. The average of the three solutions, X_1, X_2, and X_3, is used to represent the updated locations for the population, $X(t+1)$, as follows:

$$X(t+1) = 1/3 \sum_{i=1}^{3} X_i \quad (12)$$

3.2.4 Searching the area around the best solution

In the pursuit of enhancements, individuals explore regions proximate to the optimal response (leader), denoted as $X_{"Flock"1}$. This exploration process is formulated by the equation:

$$\begin{aligned} X(t+1) = {} & X(t) \\ & + D(1+z) w (X \\ & - X_{"Flock"1}) \end{aligned} \quad (13)$$

In this case, D, z, and w all contribute to the exploration process, with z being determined by $z = 1 - (\frac{t}{tmax})^2$, where t is the number of iterations and tmax is the maximum number of iterations.

3.3 Leader Harris Hawks optimization (LHHO)

The development of the leader Harris hawks optimization (LHHO) was motivated by the inquisitive behavior of the Harris hawk. The limited exploring capabilities of the HHO are one of its disadvantages. The cause is the perching strategy, which depends on equal chance q. The HHO method [47] states that if $q < 0.5$, the hawks perch based on where other family members are; if $q > 0.5$, they perch randomly on a tall tree. The chance of a perch for a single hawk can assist circumvent this.

Exploration phase ($|E| \geq 1$): Using the current hawk's fitness value with position vector X_i as $f(X_i)$ of ith hawk, the best-performing hawk's position vector X_{prey} as $f(X_{prey})$, and the worst-performing hawk's position vector X_{worst} as $f(X_{worst})$, we can define an adaptive perch probability (p_{ap}^i) of the ith hawk. Next, the following model may be used to express the adaptive perch probability (p_{ap}^i):

$$p_{ap}^i = \frac{|f(X_i) - f(X_{prey})|}{|f(X_{worst}) - f(X_{prey})|}, i = 1, 2, \ldots, N. \quad (14)$$

The inquiry phase may thus be shown as

$$X_i(new) = \begin{cases} X_{rand}(t) - r_1|X_{rand}(t) - 2r_2X_i(t)|, & q \geq p_{ap}^i \\ \left(X_{prey}(t) - X_m(t)\right) - r_3\left(LB + r_4(UB - LB)\right), & q < p_{ap}^i \end{cases}. \quad (15)$$

where $X_m(t)$ represents the mean location vector of the N hawk population at any given time. Phase of exploitation ($|E| < 1$): The exploitation phase can be imitated using the four possible attacking strategies that are comparable to the HHO and are described below [47], [48].

- *Soft besiege, in which $r \geq 0.5$ and $|E| > 0.5$.*

$$X_i(new) = X_{prey}(t) - X_i(t) - E\left|JX_{prey}(t) - X_i(t)\right|. \quad (16)$$

where J is the jump strength as determined by Eq. (16).

- *Hard besiege, in which $r > 0.5$ and $|E| < 0.5$.*

$$X_i(new) = X_{prey}(t) - E\left|X_{prey}(t) - X_i(t)\right|. \quad (17)$$

- *Gradual fast dives with soft besiege ($r < 0.5$ and $|E| \geq 0.5$).*

$$X_i(new) = \begin{cases} Y_i & if\ f(Y_i) < f\left(X_i(t)\right) \\ Z_i & if\ f(Z_i) < f\left(X_i(t)\right) \end{cases}. \quad (18)$$

- *intense barrage with progressively faster dives ($r < 0.5$ and $|E| < 0.5$).*

$$X_i(new) = \begin{cases} Y_i & if\ f(Y_i) < f\left(X_i(t)\right) \\ Z_i & if\ f(Z_i) < f\left(X_i(t)\right) \end{cases}. \quad (19)$$

The HHO algorithm's exploration phase transitions to exploitation based on the prey's escape energy [49].

Leader-dependent mutation selection Let's determine the best position vector ($X^{\,t}_{best}$), second-best ($X^{\,t}_{best-1}$), and third-best ($X^{\,t}_{best-2}$) among N individual hawks. Adjust position vectors according to the new position vector $X(new)$'s fitness function value. Thus, $X_i(mut)$, the new mutation location vector for the ith bird, may be explained as follows:

$$X_i(mut) = X_i(new) + 2 \times \left(1 - \frac{t}{t_{max}}\right) \times (2 \times rand - 1)(2 \times X_{best}^t(X_{best-1}^t + X_{best-2}^t)) + (2 \times rand - 1)(X_{best}^t - X_i(new)). \quad (20)$$

where $rand$ is a random number that falls between 0 and 1.

The location vector for the subsequent generation $X_i(t+1)$ may then be obtained by applying the selection process described in Eq. (21). Similarly, the X_{prey} is updated using Eq. (22).

$$X_i(t+1) = \begin{cases} X_i(mut) & f\left(X_i(mut)\right) < f\left(X_i(new)\right) \\ X_i(new) & f\left(X_i(mut)\right) \geq f\left(X_i(new)\right) \end{cases}. \quad (21)$$

$$X_{prey} = \begin{cases} X_i(mut) & f\left(X_i(mut)\right) < f\left(X_{prey}\right) \\ X_i(new) & f\left(X_i(new)\right) < f\left(X_{prey}\right) \end{cases}. \quad (22)$$

3.4 Performance evaluators

- ***Root Mean Square Error (RMSE):*** RMSE measures the average size of the error between expected and actual values. It provides an overall indication of how well a model fits the data: the lower the RMSE, the better the model's predictions. By squaring the errors, RMSE gives greater weight to larger discrepancies.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(O_i - P_i)^2} \quad (23)$$

- ***R-squared (R^2):*** The quality of fit, or how closely the model's predictions match the real data, is explained by R^2. It indicates the extent to which the model accounts for the variation. An excellent match is indicated by a value near 1, whereas a value around 0 indicates a limited predictive capacity.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(O_i - P_i)^2}{\sum_{i=1}^{n}(O_i - \bar{P}_i)^2} \quad (24)$$

- ***Mean Squared Error (MSE):*** The average squared difference between forecasts and actuals is determined by the MSE. Larger mistakes are highlighted due to the squaring, which makes it helpful for evaluating regressions where a lower value denotes better performance.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(O_i - P_i)^2 \quad (25)$$

- ***Weighted Absolute Percentage Error (WAPE):*** WAPE is a statistic used to assess a forecast's accuracy. It refers to the ratio of the sum of the actual values to the total absolute error. The lower WAPE, the better. Additionally, it is less susceptible to outliers than, say, RMSE or MSE.

$$WAPE = \frac{\sum_{i=1}^{n}\left|O_{i_i} - P_i\right|}{\sum_{i=1}^{n}|O_i|} \times 100 \quad (26)$$

- ***Nash-Sutcliffe Efficiency (NSE):*** NSE compares a model's prediction ability to the average of the observed data. The values vary from 1.0, which denotes perfect coincidence of anticipated and actual values, to negative values, which indicate bad forecasts. Poor prediction performance is shown by values close to 0.

$$NSE = 1 - \frac{\sum_{i=1}^{n}(O_i - P_i)^2}{\sum_{i=1}^{n}(O_i - \bar{O})^2} \quad (27)$$

where:
O_i = observed values,
P_i = predicted values,
$\bar{O}$ = mean of observed values,
n = number of observations.

3.5 Hyperparameter tuning

The hyperparameter tuning outcomes for the three predictive models, RBF+GGO (RBGG), RBF+LHHO (RBLH), and RBF, used in this investigation are compiled in Table 2. Since hyperparameter tuning directly affects the model's capacity to identify intricate patterns in the data, it is an essential step in maximizing model performance. The spread and the number of centers (Num_centers) are the two main hyperparameters taken into account. These two factors are crucial in determining the architecture and generalization potential of the RBF network. With 20 centers and a spread of 0.3, the RBGG model performed at its best, suggesting that a larger network is preferred in order to capture intricate interactions. Despite requiring a wider spread (0.6) but slightly fewer centers (18), the RBLH model appears to benefit from a smoother function approximation. On the other hand, due to its simpler architecture, the base RBF model performed best with just 10 centers and a spread of 0.5. These setups guarantee that every model is optimized for precision, dependability, and computational effectiveness.

The standalone RBF model required approximately 2 minutes to train, while the hybrid RBGG and RBLH models needed around 8 and 11 minutes, respectively, on a system equipped with 16 GB RAM and an Intel Core i7-1165G7 processor (2.80 GHz). The additional runtime for the hybrid models is due to the iterative nature of the optimization algorithms, which balance computational cost with improved predictive performance.

All data preprocessing, model development, optimization, and visualization tasks were performed using Python. Core libraries included NumPy and Pandas for data handling, Scikit-learn for machine learning implementation, and Matplotlib and Seaborn for plotting and visualization.

Table 2: The result of the hyperparameter

Models	Hyperparameter	
	Num_centers	Spread
RBGG	20	0.3
RBLH	18	0.6
RBF	10	0.5

4 Results

Table 3 compares three predictive models RBGG, RBLH, and RBF across the training, validation, and testing phases, using key performance metrics such as RMSE, R^2, MSE, WAPE, and NSE. Evaluating these models will help identify the most accurate and reliable model for predicting pile capacity, a critical factor in geotechnical engineering projects.

In the training phase of the RBGG model, the predictive performance was strong, with an RMSE value of 335.915 and a high R^2 value of 0.957. This indicates that the model explains 95.7% of the variance in pile capacity based on input features including geotechnical properties (such as average cohesion, average friction, and average soil specific weight) and geometric characteristics (such as pile area). These distinct feature categories collectively influence the model's ability to predict axial bearing capacity. However, validation and testing phases showed a decline in performance. RMSE increased to 392.524 and 427.938 for validation and testing, respectively, while R^2 dropped to 0.931 during testing. Additionally, the WAPE rose to 0.136 in testing, and the NSE deteriorated to 0.910, suggesting that the model has less ability to generalize and may be overfitting with new data.

In contrast, the RBLH model shows a different trend, performing well across all phases with a consistent RMSE reduction from training to validation and testing: 293.570, 343.919, and 368.305, respectively. The R^2 values reflect strong predictive accuracy, especially with a high-test score of 0.961. This suggests that the RBLH model is stable in its predictions, likely due to its ability to effectively utilize relationships between input variables such as average pile-soil friction and pile length, which are crucial for determining pile performance. While WAPE and MSE slightly increased during validation and testing, the NSE values remain relatively high at 0.931, indicating that the RBLH model offers reliable generalization on unseen data.

The RBF model outperforms both the RBGG and RBLH models across all phases. In the training phase, an RMSE value of 183.680 with an R^2 of 0.986 indicates that the model effectively captures complex interactions among input variables, particularly flap number and pile area, which are essential for accurately estimating pile capacity. During the validation phase, the RBF model still shows low prediction errors, with an RMSE of 214.325 and an R^2 of 0.980, demonstrating its strong generalizability. Further confirmation of the model's strength comes from the testing phase, where it achieves an RMSE of 224.246, a low MSE of 5.03E+04, and a competitive NSE of 0.873. While performance is slightly lower during testing, the model still demonstrates strong predictive capabilities, indicating its efficiency in handling the complexity of geotechnical data.

In summary, the RBF model has shown the best performance in the estimation of Pile Capacity with minimum errors and maximum accuracy during all phases. The strong results can be attributed to the model's ability to effectively make use of the input features, which were representatives of critical geotechnical factors such as soil properties, pile dimensions, and load-bearing characteristics. The RBLH model follows closely, showing stable and reliable predictions, while the RBGG model has some overfitting issues that question its generalizability. These findings bring to light the importance of choosing robust predictive models, such as RBF for Pile Capacity estimation, toward the optimization of design for ensuring structural safety in geotechnical engineering projects.

In contrast, the RBLH model consistently outperformed both RBF and RBGG in terms of R^2 and RMSE on both training and testing datasets, demonstrating high accuracy and robustness. The RBGG model, although slightly less accurate than RBLH in some instances, exhibited stable generalization behavior,

particularly when tested under varied soil and pile conditions. This suggests that while RBLH is the overall best-performing model, RBGG may be preferable in scenarios where robustness across diverse inputs is a priority.

Therefore, RBLH is identified as the most accurate and reliable model for predicting pile bearing capacity in this study, whereas RBGG offers valuable generalization potential under fluctuating conditions.

Table 3: The outcome of the constructed models for RBF

Model	Phase	Index values				
		RMSE	R^2	MSE	WAPE	NSE
RBGG	Train	335.915	0.957	1.13E+05	0.103	0.945
	Validation	392.524	0.942	1.54E+05	0.120	0.925
	Test	427.938	0.931	1.83E+05	0.136	0.910
RBLH	Train	293.570	0.950	8.62E+04	0.144	0.937
	Validation	343.919	0.932	1.18E+05	0.170	0.914
	Test	368.305	0.961	1.36E+05	0.165	0.931
RBF	Train	183.680	0.986	3.37E+04	0.083	0.974
	Validation	214.325	0.980	4.59E+04	0.096	0.964
	Test	224.246	0.923	5.03E+04	0.107	0.873

Fig. 3 shows the convergence trace for two hybrid models, RBGG and RBLH, during the training process, along with their respective Root Mean Square Error (RMSE) values. Both models start with the same initial RMSE of 585, indicating they begin at the same level of predictive accuracy. However, during the first 70 iterations, the RBGG model exhibited higher RMSE values compared to RBLH, suggesting initial instability or slower adaptation to the data. As the iterations progressed, RBGG showed substantial improvement, demonstrating a strong convergence path. By the end of the optimization process, the RBGG model significantly outperformed RBLH, achieving an RMSE value 56.61 times lower, with a final RMSE score of 392.52. This result highlights the superior learning capability and optimization efficiency of the RBGG model, making it more suitable for enhancing the predictive accuracy of pile capacity in geotechnical applications.

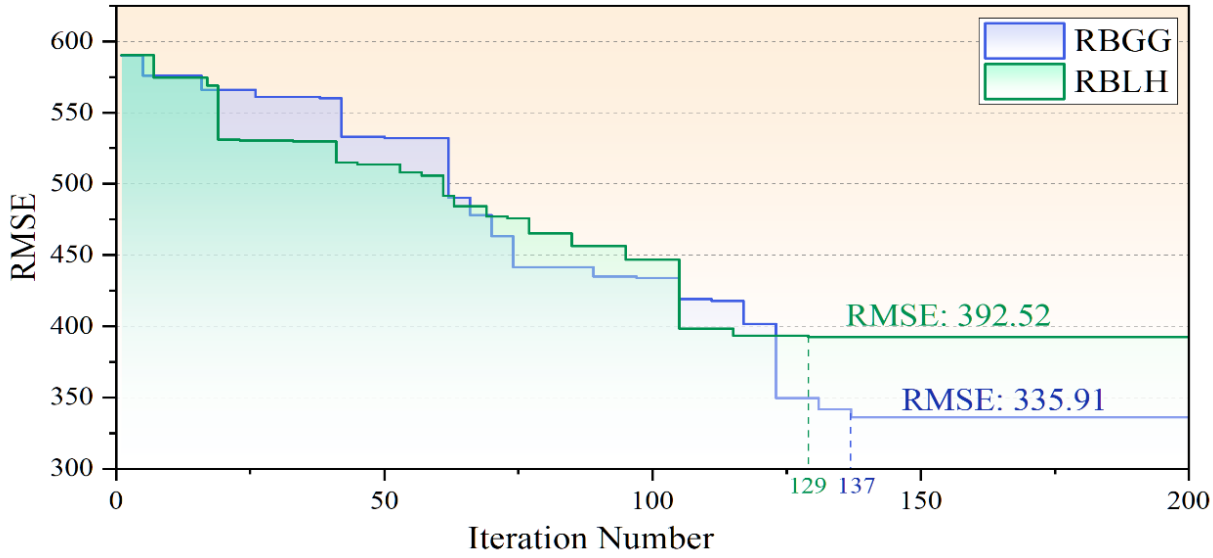


Figure 3: The convergence curves of the two hybrid models being showcased

The dispersion graphs of the hybrid models that evolved are shown in Fig. 4, together with the performances regarding RMSE and R^2 scores for the Train, Validation, and Test phases. This graph is needed to deeply understand how each model performs with respect to the ideal Best Fit line. Among the model comparison presented, RBGG proved to be more consistent, where results are well-clustered in a tight range of $[-10\%x, 10\%x]$, depicting very strong predictive accuracy in each phase. On the other hand, the base model RBF yielded more scattered results from the Best Fit line, indicating greater variability and lower reliability of that model's predictions. These patterns highlight how developments made using the RBGG model minimized errors and increased correlation, thus providing close agreement with the focus of this study on the optimization of hybrid models to enhance pile capacity prediction accuracy.

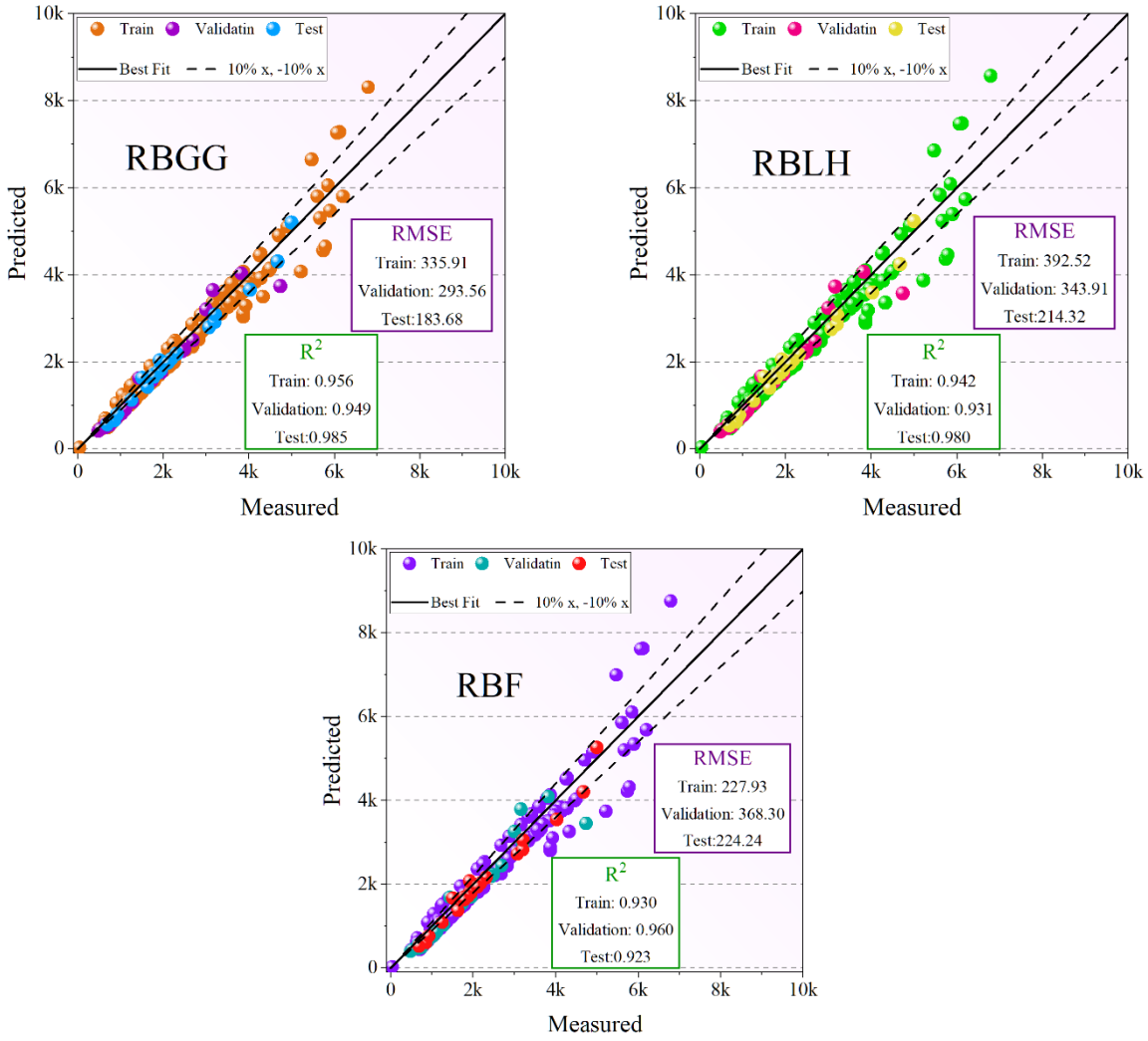
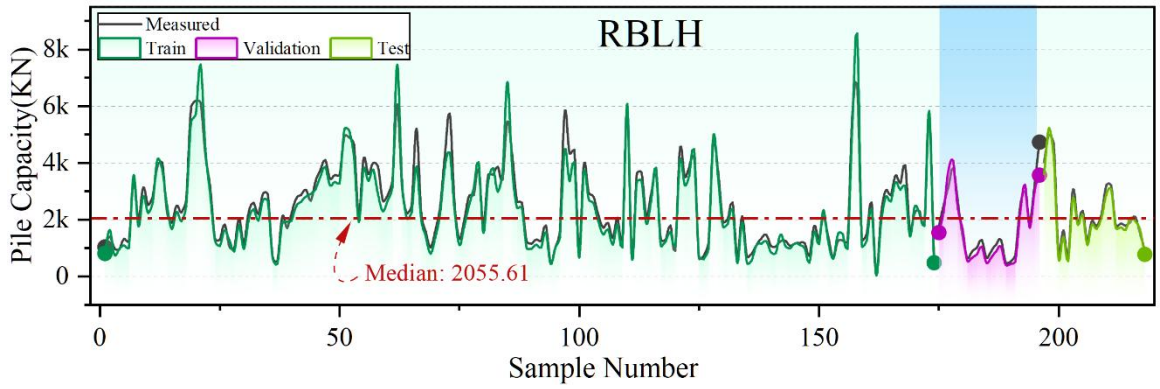


Figure 4: The dispersion of evolved hybrid models

Fig. 5 provides a detailed comparison between the predictions of the hybrid models and the actual observed data, emphasizing the predictive accuracy of each model. The RBGG model consistently outperforms the others across all three important phases: Training, Validation, and Testing. It demonstrates an impressive match with the actual values, with deviations being minimal compared to the other models. This indicates that RBGG is the most reliable model for generalization across different datasets. By making predictions that closely align with the measured data, the RBGG model stands out as the most dependable among those studied. As a result, it is particularly suitable for applications requiring high precision in predicting pile capacity and related geotechnical parameters. The consistency of RBGG in delivering accurate predictions reinforces its practical value, especially in optimizing pile design and ensuring structural safety in engineering projects.



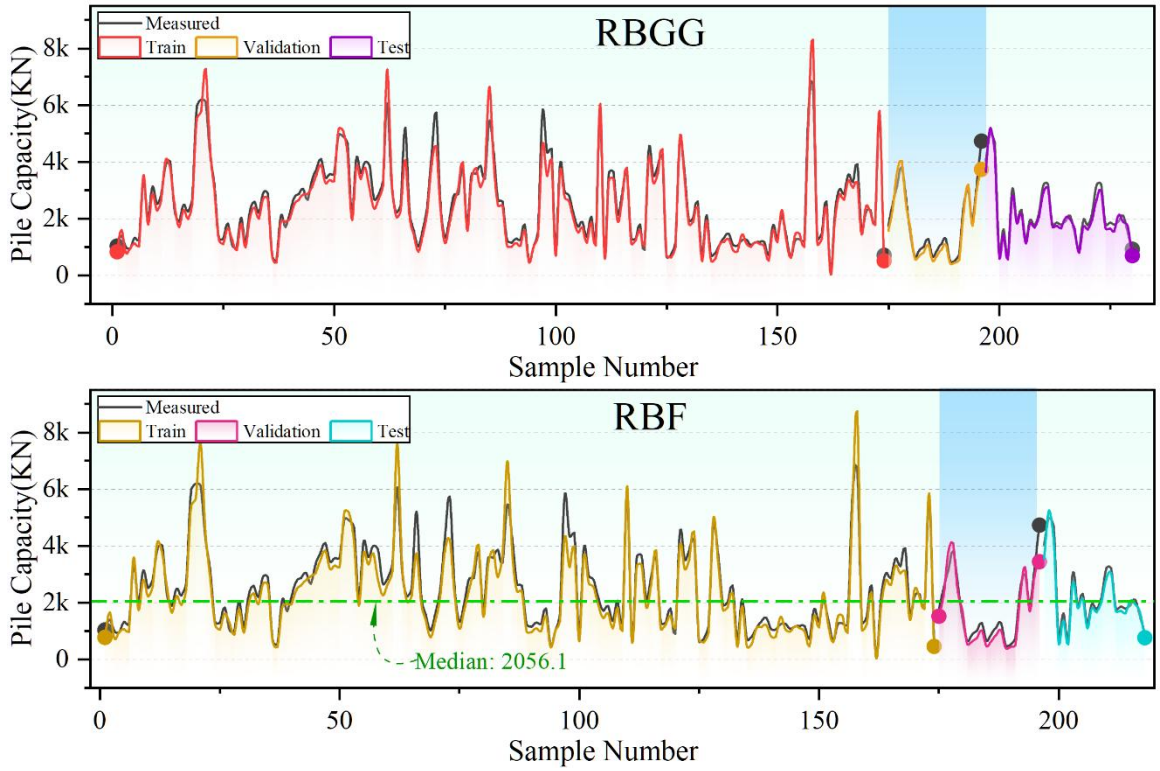
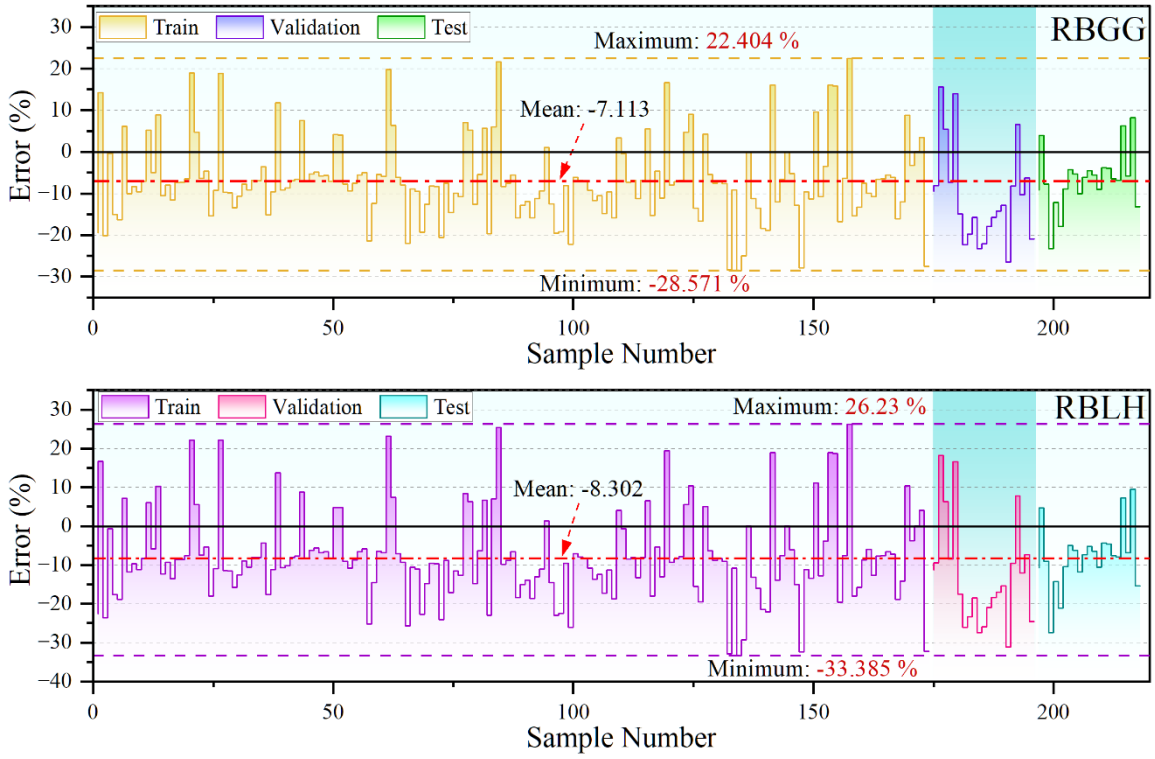


Figure 5: The comparison between forecasted and actual values

Fig. 6 illustrates the error percentages of the RBF, RBLH, and RBGG models using a vertical bar plot across the Training, Validation, and Testing phases. The enhanced figure includes labeled axes for error percentage, consistent color coding for each model, and a comprehensive legend. The RBGG model exhibits the most constrained error range, from -28.571% to 22.404%, suggesting high accuracy and low variance. This narrow span reflects the model's robustness and adaptability, particularly under varying data conditions. Despite its overall superior performance, a few mild outliers can be observed in the RBGG model, primarily during the training phase. These may be attributed to inherent noise or measurement uncertainties in the input dataset rather than model instability. Importantly, these outliers do not significantly impact the model's overall predictive performance, as evidenced by its consistent results in the validation and testing phases.



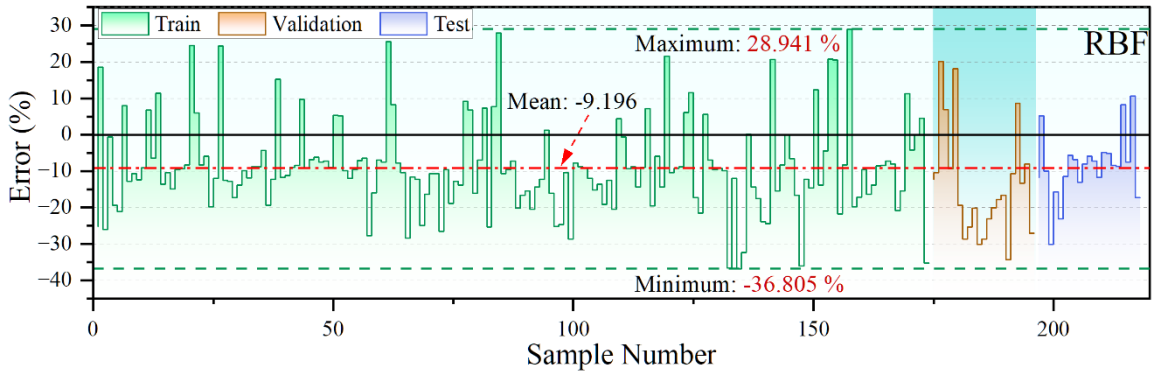


Figure 6: The error percentage of the models is derived from the vertical plot

Fig. 7 presents a raincloud plot to provide a comprehensive view of the error distribution across all three phases. This visualization combines density curves, box plots, and raw data scatter, offering insights into spread and skewness. RBGG again demonstrates the tightest distribution and lowest error spread (Train: -7%, Validation: -11%, Test: -6%). The minimal presence of long tails or extreme values supports the model's precision and generalization capability. Collectively, the updated visualizations and expanded discussion reinforce the strength of the RBGG model not only in average accuracy but also in error consistency and resilience to outlier effects. This robustness is particularly valuable in practical geotechnical engineering applications, where prediction reliability under uncertainty is critical.

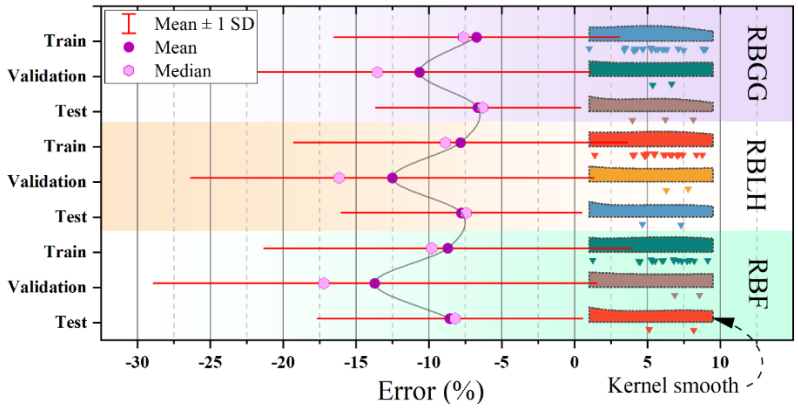


Figure 7: The error raincloud plot of the proposed models

The SHAP (SHapley Additive exPlanations) values that are used to assess each input feature's relative contribution to pile capacity prediction are shown in Fig. 8. By giving each feature an importance value determined by how much it contributes to the model's prediction, SHAP analysis is a potent interpretability technique that clarifies the results of machine learning models. Pile Capacity has the highest SHAP value in this figure, confirming its direct function as the output target. The most important characteristics among the input variables are Pile Length, Average Friction Angle, and Flap Number; each makes a substantial contribution to the prediction model. These factors are essential for capturing the mechanical, depth, and resistance effects of soil conditions on pile performance. Both average soil specific weight and average pile-soil friction angle show a moderate amount of influence, suggesting that they play a part in determining the characteristics of effective stress and load transfer. Pile Area, on the other hand, exhibits little influence, indicating that its variation makes a smaller contribution to capacity prediction within the examined dataset. For more effective, precise, and understandable pile capacity prediction techniques in practical engineering applications, the SHAP analysis thus offers direct insights into which geotechnical parameters should be given priority during model development and optimization.

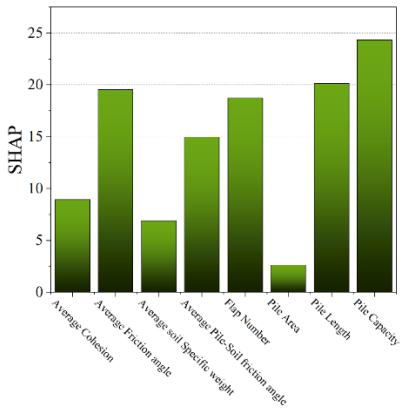


Figure 8: SHAP sensitivity analysis for feature impact evaluation

A comparative overview of the current study and earlier studies on pile capacity prediction using different modeling techniques is shown in Table 4. Regression and EPR-MOGA techniques were used by Alzabeebee et al. [19], who demonstrated how well EPR-MOGA captures nonlinear soil-friction interactions. For drilled shafts, Shahin [16] used ANN on 174 data points and obtained R^2 values as high as 0.97. Using important geotechnical inputs, Samui [17] employed SVM and achieved high, though unspecified, accuracy. M. Kumar et al. [20] used deep learning (e.g., XGBoost, RF) and ensemble methods, with XGBoost producing an R^2 of 0.91. Alzabeebee et al. [50] demonstrated in another study that EPR can predict undrained adhesion better than empirical models. When DNN and RNN variants were investigated by M. Kumar et al. [51], DNN performed the best (R^2 = 0.97). When Kumar et al. [52] combined ANN with different metaheuristics, GP and ANN-PSO achieved R^2 values greater than 0.97. Using 220 actual pile load test data, the current study presents a hybrid RBF-based model optimized with LHHO and GGO. The RBLH model, which included temporal factors, sensitivity analysis, and correlation-based input selection, had the highest accuracy (R^2 = 0.986), suggesting a significant contribution to predictive modeling in geotechnical engineering.

Table 4: Comparison between the presented and published studies

Study	Method Used	Dataset	Performance (R^2)	Highlights
[19]	Regression, EPR-MOGA	Not specified	EPR-MOGA: 0.85–0.92Regression: 0.16–0.67	EPR-MOGA captured complex soil-friction relationships in fine-grained soils
[16]	ANN	174 data points	0.85 (Driven piles)0.97 (Drilled shafts)	Effective for axial capacity prediction using ANN
[17]	SVM	Not specified	High accuracy (value not reported)	Used number of bowls, mean stress, and depth ratio as inputs
[20]	XGBoost, RF, GBM, DL	81 static load tests	XGBoost: 0.91	SHAP and Taylor diagrams used for model evaluation
[50]	EPR (Multi-objective)	Not specified	Superior to empirical methods	Predicted undrained adhesion factor in bored piles
[51]	DNN, CNN, RNN, LSTM, BiLSTM	257 dynamic load tests	DNN: 0.97	BiLSTM second best; RNN underperformed due to multicollinearity
[52]	ANN + PSO, GOA, ABC, ACO, ALOGP, MARS, GMDH	Dynamic pile load tests	ANN-PSO: 0.9773GP: 0.9859	Metaheuristic-optimized ANN superior to traditional models
Present Study	RBF + LHHO (RBLH), RBF + GGO (RBGG)	220, Real pile load test data	RBLH: R^2 = 0.986RMSE = 183.680	Novel hybrid model with time-dependent variables, correlation-based feature selection, high generalization, and sensitivity analysis

5 Conclusions

The key finding of this paper, in particular, highlights the central position of advanced predictive models to perform pile capacity estimations, the crucial element in geotechnical engineering. Key performance metrics, such as RMSE, R^2, MSE, WAPE, and NSE, were chosen for the evaluation of RBGG, RBLH, and RBF for the training, validation, and testing phases. It can be seen from the results that the best overall performance was given by the RBF model, with the minimum RMSE value and the highest R^2 value, especially in the training phase, which reflects its strong ability in capturing complex interactions of critical geotechnical variables, including soil properties and pile dimensions. However, this study also included a sensitivity analysis of time-varying features, which provided good insight into the behavior of models wrt

changes in dynamic factors over time. This sensitivity analysis continued to support the strength of the RBF model, given that at every instance it showed strong predictive performance, even when exposed to variations in input features across different phases. On the other hand, highly effective though the RBGG model was in training, it resulted in significant overfitting upon validation and testing, especially in decreasing performance with time-dependent variables introduced. It thus appears that RBGG fails to generalize over time, limiting its practical applicability.

The RBLH model, though not outcompeting RBF, showed consistent performance across all phases. Therefore, it proved to be a good choice for pile capacity prediction, especially when time-dependent factors were included. It gives solid performance with lower error margins and higher generalizability, pointing out its capacity to effectively incorporate the change of input features over time. In general, the results seem to underline the importance of precision and generalization in model selection for geotechnical applications. Among these, the performance of the RBF model is the most consistent and reliable due to its ability to adapt to time-dependent variables, thus it can be considered a very powerful tool in pile capacity estimation. Further studies can be done by incorporating more optimization techniques into the models for their strength and flexibility in adjusting to changing geotechnical conditions, which will be beneficial in enhancing prediction accuracy and reliability in real-world engineering projects.

Acknowledgements

I would like to take this opportunity to acknowledge that there are no individuals or organizations that require acknowledgment for their contributions to this work.

Competing of interests

The authors declare no competing of interests.

Authorship contribution statement

Shanwei Zhang: Writing-Original draft preparation, Conceptualization, Supervision, Project administration.

Data availability

On Request

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Author statement

The manuscript has been read and approved by all the authors, the requirements for authorship, as stated earlier in this document, have been met, and each author believes that the manuscript represents honest work.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Ethical approval

All authors have been personally and actively involved in substantial work leading to the paper, and will take public responsibility for its content.

References

[1] K. Fakharian and M. Khanmohammadi, “Comparison of pile bearing capacity from CPT and dynamic load tests in clay considering soil setup,” *Frontiers in Offshore Geotechnics III; CRC Press: Boca Raton, FL, USA*, pp. 539–544, 2015.

[2] C. Su, L. Wang, Z. Wu, and H. Xu, “Research on weighing of concrete aggregate pile based on binocular vision,” *Informatica*, vol. 46, no. 7, 2022.

[3] B. Naeim, M. R. Akbarzadeh, and V. Jahangiri, “Machine learning-based prediction of seismic response of elevated steel tanks,” *Structures*, vol. 70, p. 107649, 2024, doi: https://doi.org/10.1016/j.istruc.2024.107649.

[4] M. Khanmohammadi and K. Fakharian, “Evaluation of performance of piled-raft foundations on soft clay: A case study,” *Geomechanics and Engineering*, vol. 14, no. 1, pp. 43–50, 2018.

[5] V. E. Komurka, A. B. Wagner, and T. B. Edil, “Estimating soil/pile set-up,” Citeseer, 2003.

[6] M. N. Haque and E. J. Steward, “Evaluation of pile setup phenomenon for driven piles in Alabama,” in *Geo-Congress 2020*, American Society of Civil Engineers Reston, VA, 2020, pp. 200–208.

[7] B. Naeim, A. J. Khiavi, P. Dolatimehr, and B. Sadaghat, “Novel Optimized Support Vector Regression Networks for Estimating Fresh and Hardened Characteristics of SCC,” 2024.

[8] E. Khajavi, A. R. Taghavi Khanghah, and A. Javadzade Khiavi, “An efficient prediction of punching shear strength in reinforced concrete slabs through boosting methods and metaheuristic algorithms,” *Structures*, vol. 74, p. 108519, 2025, doi: https://doi.org/10.1016/j.istruc.2025.108519.

[9] K. Fakharian and M. Khanmohammadi, “Effect of OCR and pile diameter on load movement response of piles embedded in clay over time,” *International Journal of Geomechanics*, vol. 22, no. 7, p. 4022091, 2022.

[10] R. Skov and H. Denver, “Time-dependence of bearing capacity of piles,” in *Proc. Third International Conference on the Application of Stress-Wave Theory to Piles. Ottawa*, 1988, pp. 25–27.

[11] M. W. Gardner and S. R. Dorling, “Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences,” *Atmos Environ*, vol. 32, no. 14–15, pp. 2627–2636, 1998.

[12] M. Parsajoo, D. J. Armaghani, A. S. Mohammed, M. Khari, and S. Jahandari, “Tensile strength prediction

of rock material using non-destructive tests: A comparative intelligent study," *Transportation Geotechnics*, vol. 31, p. 100652, 2021.

[13] D. Li, Z. Liu, D. J. Armaghani, P. Xiao, and J. Zhou, "Novel ensemble tree solution for rockburst prediction using deep forest," *Mathematics*, vol. 10, no. 5, p. 787, 2022.

[14] M. Koopialipoor, P. G. Asteris, A. S. Mohammed, D. E. Alexakis, A. Mamou, and D. J. Armaghani, "Introducing stacking machine learning approaches for the prediction of rock deformation," *Transportation Geotechnics*, vol. 34, p. 100756, 2022.

[15] I.-M. Lee and J.-H. Lee, "Prediction of pile bearing capacity using artificial neural networks," *Comput Geotech*, vol. 18, no. 3, pp. 189–200, 1996.

[16] M. A. Shahin, "Intelligent computing for modeling axial capacity of pile foundations," *Canadian Geotechnical Journal*, vol. 47, no. 2, pp. 230–243, 2010.

[17] P. Samui, "Prediction of pile bearing capacity using support vector machine," *International Journal of Geotechnical Engineering*, vol. 5, no. 1, pp. 95–102, 2011.

[18] E. Momeni, R. Nazir, D. J. Armaghani, and H. Maizir, "Prediction of pile bearing capacity using a hybrid genetic algorithm-based ANN," *Measurement*, vol. 57, pp. 122–131, 2014.

[19] S. Alzabeebee, B. Ismael, S. Keawsawasvong, and A. W. Alshami, "Evolutionary Polynomial Regression Model to Predict Effective Angle of Internal Friction of Fine-Grained Soils," *Transportation Infrastructure Geotechnology*, vol. 12, no. 2, p. 97, 2025, doi: 10.1007/s40515-025-00557-0.

[20] M. Kumar, S. Pijush, K. Divesh Ranjan, and P. G. and Asteris, "State-of-the-art XGBoost, RF and DNN based soft-computing models for PGPN piles," *Geomechanics and Geoengineering*, vol. 19, no. 6, pp. 975–990, Nov. 2024, doi: 10.1080/17486025.2024.2337702.

[21] F. Milad, T. Kamal, H. Nader, and O. E. Erman, "New method for predicting the ultimate bearing capacity of driven piles by using Flap number," *KSCE Journal of Civil Engineering*, vol. 19, no. 3, pp. 611–620, Mar. 2015, doi: 10.1007/s12205-013-0315-z.

[22] D. Jahed Armaghani, R. S. N. S. B. R. Shoib, K. Faizi, and A. S. A. Rashid, "Developing a hybrid PSO–ANN model for estimating the ultimate bearing capacity of rock-socketed piles," *Neural Comput Appl*, vol. 28, no. 2, pp. 391–405, Feb. 2017, doi: 10.1007/s00521-015-2072-z.

[23] S. Shaik, K. S. R. Krishna, M. Abbas, M. Ahmed, and D. Mavaluru, "Applying several soft computing techniques for prediction of bearing capacity of driven piles," *Eng Comput*, vol. 35, no. 4, pp. 1463–1474, Oct. 2019, doi: 10.1007/s00366-018-0674-7.

[24] A. Dehghanbanadaki, M. Khari, S. T. Amiri, and D. J. Armaghani, "Estimation of ultimate bearing capacity of driven piles in c-φ soil using MLP-GWO and ANFIS-GWO models: a comparative study," *Soft comput*, vol. 25, no. 5, pp. 4103–4119, Mar. 2021, doi: 10.1007/s00500-020-05435-0.

[25] H. Harandizadeh, D. Jahed Armaghani, and M. Khari, "A new development of ANFIS–GMDH optimized by PSO to predict pile bearing capacity based on experimental datasets," *Eng Comput*, vol. 37, no. 1, pp. 685–700, Jan. 2021, doi: 10.1007/s00366-019-00849-3.

[26] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1481–1497, 1990.

[27] C. A. Micchelli, *Interpolation of scattered data: distance matrices and conditionally positive definite functions*. Springer, 1984.

[28] Y. Liao, S.-C. Fang, and H. L. W. Nuttle, "Relaxed conditions for radial-basis function networks to be universal approximators," *Neural Networks*, vol. 16, no. 7, pp. 1019–1028, 2003.

[29] K.-L. Du, K. K. M. Cheng, and M. N. S. Swamy, "A fast neural beamformer for antenna arrays," in *2002 IEEE International Conference on Communications. Conference Proceedings. ICC 2002 (Cat. No. 02CH37333)*, IEEE, 2002, pp. 139–144.

[30] A. R. Webb, "Functional approximation by feed-forward networks: a least-squares approach to generalization," *IEEE Trans Neural Netw*, vol. 5, no. 3, pp. 363–371, 1994.

[31] D. Lowe, "On the use of nonlocal and non positive definite basis functions in radial basis function networks," 1995.

[32] J. Meinguet, "Multivariate interpolation at arbitrary points made simple," *Zeitschrift für angewandte Mathematik und Physik ZAMP*, vol. 30, no. 2, pp. 292–304, 1979.

[33] I. Rojas *et al.*, "Time series analysis using normalized PG-RBF network with regression weights," *Neurocomputing*, vol. 42, no. 1–4, pp. 267–285, 2002.

[34] C.-J. Lin and W.-H. Ho, "A pseudo-Gaussian-based compensatory neural fuzzy system," in *The 12th IEEE International Conference on Fuzzy Systems, 2003. FUZZ'03.*, IEEE, 2003, pp. 214–219.

[35] C.-C. Lee, P.-C. Chung, J.-R. Tsai, and C.-I. Chang, "Robust radial basis function neural networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 29, no. 6, pp. 674–685, 1999.

[36] S.-J. Lee and C.-L. Hou, "An ART-based construction of RBF networks," *IEEE Trans Neural Netw*, vol. 13, no. 6, pp. 1308–1321, 2002.

[37] R. J. Schilling, J. J. Carroll, and A. F. Al-Ajlouni, "Approximation of nonlinear systems with radial basis function neural networks," *IEEE Trans Neural Netw*, vol. 12, no. 1, pp. 1–15, 2001.

[38] K.-L. Du and M. N. S. Swamy, *Neural networks in a softcomputing framework*, vol. 501. Springer, 2006.

[39] P. Singla, K. Subbarao, and J. L. Junkins, "Direction-dependent learning approach for radial basis function networks," *IEEE Trans Neural Netw*, vol. 18, no. 1, pp. 203–222, 2007.

[40] G. Szipl, A. Loth, C. A. F. Wascher, J. Hemetsberger, K. Kotrschal, and D. Frigerio, "Parental behaviour and family proximity as key to gosling survival in Greylag Geese (Anser anser)," *J Ornithol*, vol. 160, pp. 473–483, 2019.

[41] M. Hoarau, F. Angelier, F. Touzalin, T. Zgirski, C. Parenteau, and P. Legagneux, "Corticosterone: foraging and fattening puppet master in pre-breeding greylag geese," *Physiol Behav*, vol. 246, p. 113666, 2022.

[42] A. M. Zaki, S. K. Towfek, W. Gee, W. Zhang, and M. A. Soliman, "Advancing parking space surveillance using a neural network approach with feature extraction and dipper throated optimization integration," *Journal of Artificial Intelligence and Metaheuristics*, vol. 6, no. 2, pp. 16–25, 2023.

[43] T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, and A. Cosar, "A survey on new generation metaheuristic algorithms," *Comput Ind Eng*, vol. 137, p. 106040, 2019.

[44] M. Braik, A. Sheta, and H. Al-Hiary, "A novel meta-heuristic search algorithm for solving optimization problems: capuchin search algorithm," *Neural Comput Appl*, vol. 33, no. 7, pp. 2515–2547, 2021.

[45] A. H. Halim, I. Ismail, and S. Das, "Performance assessment of the metaheuristic optimization algorithms: an exhaustive review," *Artif Intell Rev*, vol. 54, no. 3, pp. 2323–2409, 2021.

[46] D. S. Khafaga *et al.*, "An Al-Biruni earth radius optimization-based deep convolutional neural network for classifying monkeypox disease," *Diagnostics*, vol. 12, no. 11, p. 2892, 2022.

[47] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future generation computer systems*, vol. 97, pp. 849–872, 2019.

[48] S. L. Ayinla *et al.*, "Optimizing DC Motor Control via Leader Harris Hawks Algorithm with ITSE-ZLG Objective Function," in *2023 10th IEEE International Conference on Power Systems (ICPS)*, IEEE, 2023, pp. 1–6.

[49] Z. Liu, Y. Fang, L. Liu, and S. Ma, "A multi-leader Harris hawks optimizer with adaptive mutation and its application for modeling of silicon content in liquid iron of blast furnace," *Math Comput Simul*, vol. 213, pp. 466–514, 2023.

[50] S. Alzabeebee, A. A. Zuhaira, and R. K. S. Al-Hamd, "Development of an optimized model to compute the undrained shaft friction adhesion factor of bored piles," *Geomechanics and Engineering*, vol. 28, no. 4, pp. 397–404, 2022.

[51] M. Kumar, D. R. Kumar, J. Khatti, P. Samui, and K. S. Grover, "Prediction of bearing capacity of pile foundation using deep learning approaches," *Frontiers of Structural and Civil Engineering*, vol. 18, no. 6, pp. 870–886, 2024, doi: 10.1007/s11709-024-1085-z.

[52] M. Kumar, V. Kumar, B. G. Rajagopal, P. Samui, and A. Burman, "State of art soft computing based simulation models for bearing capacity of pile foundation: a comparative study of hybrid ANNs and conventional models," *Model Earth Syst Environ*, vol. 9, no. 2, pp. 2533–2551, 2023, doi: 10.1007/s40808-022-01637-7.