Dynamic Access Control and Fine-Grained Searchable Encryption for Cloud Data Using Trust Evaluation and B-Tree Indexing

Dapeng Zhao[1], Yapeng Zhao[2], Xuexia Dou[1*]
[1]Henan Polytechnic, Zhengzhou 450046, China
[2]Henan Polytechnic University, Jiaozuo 454003, China
E-mail: zhengzhou16886@163.com
[*]Corresponding author

Keywords: SE, B-tree, cloud servers, user permissions, safety management, trust value, access control

Received: July 2, 2025

The security management of data stored on cloud servers is of great significance, as it not only prevents data leakage but also ensures data integrity. Thus, to achieve secure management of data stored on cloud servers, the research starts with existing searchable encryption technologies and attribute-based searchable encryption technologies, and improved designs are made for both to form a complete encryption scheme. Firstly, in terms of searchable encryption technology, the study considers the dynamic changes in user permissions and constructs an encryption scheme that includes a trust value evaluation model and a dual dynamic access control mechanism. Among them, the trust value evaluation model needs to consider historical trust values, recommended trust values, and attribute related trust values, and then generate a comprehensive trust value to evaluate user permissions. Secondly, in terms of attribute-based searchable encryption technology, the research adopts a multi-branch balanced tree and linear secret sharing scheme to construct a fine-grained access control scheme. Among them, the B-Tree index structure is used to optimize search efficiency, and the linear secret sharing scheme is used to achieve access control for users. In the experimental part, the study uses Python programming language combined with PyCrypto and OpenSSL cryptographic libraries for testing to ensure the feasibility and performance of the solution, and the operating system is Windows 10. The results show that the maximum trust value calculation time considering the user permission scheme is 32 ms, which is 46 ms, 33 ms, 22 ms, and 19 ms lower than the maximum values of the four comparison schemes, respectively. In addition, the maximum access control determination time, CPU utilization, and memory occupancy of this scheme are 56 ms, 12.17%, and 13.95%, respectively. The maximum key generation time and communication volume for supporting fine-grained access control schemes are 180 ms and 3.257 Byte, respectively, and the average storage overhead for user keys and ciphertext is 3.94 KB and 5.37 KB, respectively. The encryption schemes designed by the research have good performance and can provide technical support for secure management of cloud server storage data without decryption for data queries.

Povzetek: Predlagajo dinamičen nadzor dostopa s celovitim zaupanjem ter ABSE z B-drevesnim indeksiranjem in LSSS za fino zrnat, učinkovit iskalni dostop v oblaku.

1 Introduction

As cloud computing technology advances at a rapid pace, cloud servers have become an important platform for data storage and processing. However, data storage in cloud environments faces many security challenges, such as data leakage risks, complex access control permission management, and data privacy protection issues [1-2]. To ensure the security and integrity of data in cloud servers, while meeting the needs of users for efficient data retrieval and access, it holds particular significance in studying an efficient, secure, and fine-grained access control supported data storage management solution. The commonly-used methods for addressing this issue include Searchable Encryption (SE) and Attribute-Based Searchable Encryption (ABSE) [3]. In addition, many scholars have conducted research on SE and ABSE technologies.

Liu et al. designed an SE data sharing scheme based on inverted indexing to address issues such as unreliable data sharing, data attacks, and low efficiency in ciphertext retrieval. This scheme adopted a dual chain structure to store and share data, and constructed an inverted index structure and a ciphertext search algorithm based on this index structure. The results showed that this scheme could ensure datan SEcurity and improve retrieval efficiency [4]. Ng et al. proposed a blockchain-based multi-keyword SE scheme to protect the tracking data of contacts with a certain disease. Meanwhile, the scheme adopted the advanced encryption standard Galois/Counter mode to encrypt data and supported dynamic updates of search indexes. The results showed that the scheme could work effectively without affecting security objectives, and could also maintain efficiency when using larger search indexes [5]. Liu et al. designed an efficient multi-authority ABSE solution assisted by blockchain technology to

address issues such as single attribute authorization failure, privacy leakage during the search process, and high decryption overhead in ABSE technology. This framework leverages consortium blockchain technology for global public parameter management, integrating smart contracts, hybrid online/offline mechanisms, and verifiable edge-assisted decryption protocols to enhance system security and efficiency. The results showed that compared with existing solutions, this approach significantly improved computational efficiency [6]. Lu et al. designed a retrieval strategy ABSE scheme based on key policy attribute encryption framework to provide encryption support for privacy preserving database architectures. This scheme bound security query policies to query credentials and used a policy matrix to optimize the generation of security query policies. The results showed that the scheme achieved semantic security under the selection plaintext attack that included policy and identity queries [7].

Nevertheless, these studies and methodologies are not without their drawbacks, including the inability of SE technology to effectively respond to dynamic changes in user permissions, and the need to improve the search efficiency of ABSE technology. In order to better demonstrate the shortcomings of existing technologies, a detailed comparison of related works was conducted, as shown in Table 1.

Table 1: Summary table of related work

Comparative method	Features	Limitations	Quantitative results
Liu et al. [4]	SE Data Sharing Scheme Based on Inverted Index	Not involving dynamic changes in user permissions, ignoring static access permission issues	Reduce computing costs by 20% and storage expenses by 10%
Ng et al. [5]	Multi keyword SE scheme based on blockchain	Not explicitly mentioning the efficiency of dynamic management of user permissions and access control	Efficiency increased by 30%, computational cost increased by 8%
Liu et al. [6]	Efficient and Multi Authoritative ABSE Scheme Assisted by Blockchain	Unresolved problem of low search efficiency in ABSE technology	40% increase in computational efficiency and 25% increase in storage overhead
Lu et al. [7]	ABSE scheme based on key policy attribute encryption framework	Handling of dynamic changes in user permissions not mentioned	Efficiency increased by 35%, storage overhead increased by 20%

From Table 1, although existing research has made some progress in SE and ABSE technologies, there are still some common limitations. Some solutions did not fully consider the dynamic changes in user permissions, resulting in insufficient flexibility in practical application scenarios. In addition, the search efficiency of ABSE technology still needs to be further improved to meet the needs of large-scale data processing. These shortcomings provide directions for improvement in this study, prompting us to design encryption schemes that are more adaptable and efficient.

Therefore, the research questions are as follows: how to achieve secure storage and efficient management of data in cloud servers, while supporting fine-grained access control; How to dynamically manage user permissions to adapt to the dynamic changes in user permissions; How to improve the search efficiency of ABSE technology. To address the aforementioned research issues, a cryptographic scheme combining SE technology and ABSE technology has been proposed. Specifically, a trust value (TV) evaluation model and a dual dynamic access control mechanism were designed to achieve dynamic management of user permissions and solve the problem of dynamic changes in user permissions. This model evaluates the user's comprehensive TV in real time by comprehensively considering the historical TV, recommended TV and attribute related TV, so as to dynamically adjust user permissions and ensure that only trusted users can access sensitive data. At the same time, the study introduced a Balanced Multi Way Tree (B-Tree) index structure and a Linear Secret Sharing Scheme (LSSS) to optimize the search efficiency of ABSE technology and achieve fine-grained access control. The goal of the research is to design a cloud server data storage management scheme that can adapt to dynamic changes in user permissions by improving SE and ABSE, to achieve secure storage, efficient retrieval, and fine-grained access control of data. The novelty of the present research is embodied in several aspects. It involves the creation of a TV evaluation model, the application of B-Tree to optimize ABSE technology, the attainment of efficient data retrieval capabilities, and the offering of a brand-new technological route for the secure storage and effective management of data on cloud servers.

2 Methods and materials

To address the security management issues of data storage in cloud servers, an SE-oriented user permission dynamic management method was first developed to consider the dynamic changes in user permissions. Meanwhile, the research also designed an ABSE scheme based on B-Tree index structure and combined it with user permission dynamic management method to form a complete encryption method.

2.1 Design of user permission dynamic management method for SE

To manage the security of data stored in cloud servers, the research starts with existing encryption technologies, namely SE technology and ABSE technology. Research has improved the shortcomings of these two technologies and developed new encryption schemes for each. Finally, these two encryption schemes are deployed on the encryption system to form a complete encryption scheme. SE technology is a technique that allows users to perform search operations in encrypted data, while ensuring the privacy and security of the data in an unencrypted state [8-9]. The advantages of this technology are strong retrieval capability, high flexibility, strong security, support for multiple application scenarios, and effective prevention of data leakage. The schematic of SE technology is shown in Figure 1.

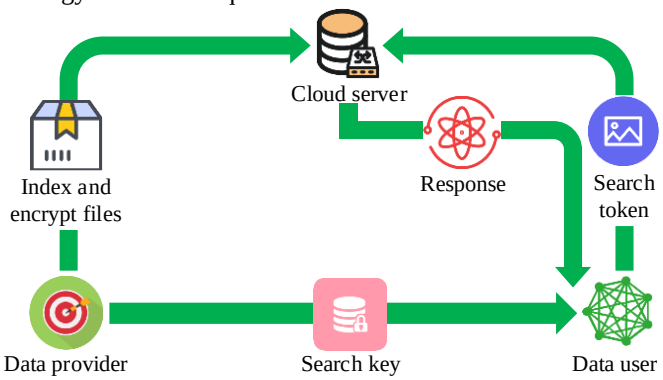


Figure 1: The schematic of SE technology

From Figure 1, SE technology involves data providers, data consumers, cloud servers, search keys, indexes and encrypted files, search tokens, and responses. However, a large portion of SE solutions currently do not involve dynamic changes in user permissions and ignore potential issues with static user access permissions. Therefore, to solve this problem, an SE-oriented user permission dynamic management method has been studied and designed. Figure 2 illustrates the comprehensive structure of this method.

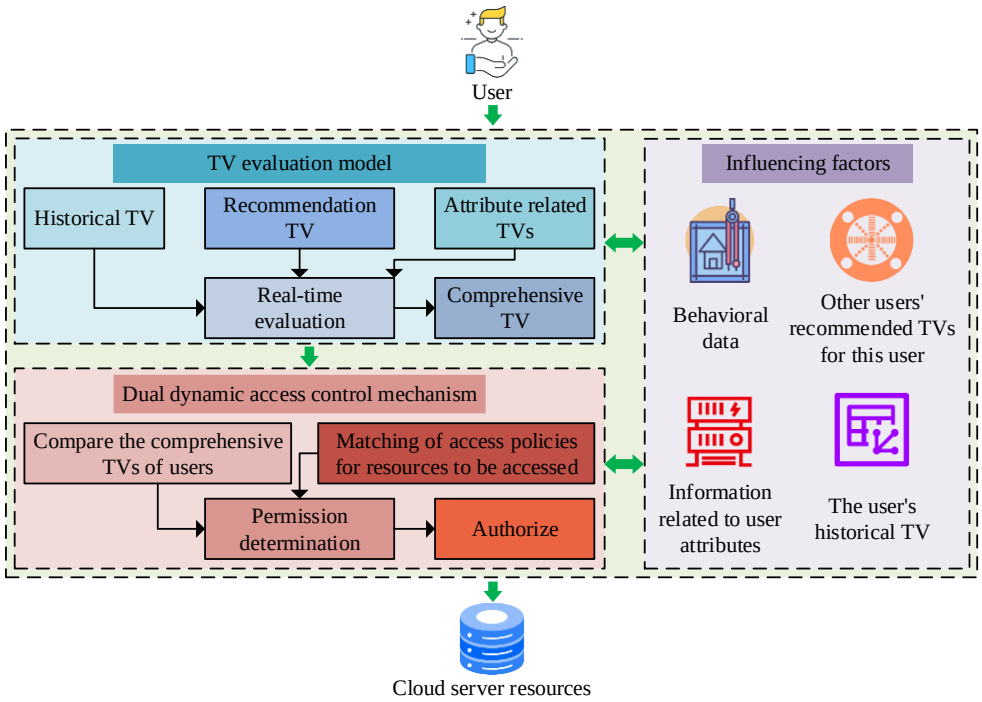


Figure 2: The overall framework of SE oriented user permission dynamic management method

From Figure 2, the user permission dynamic management method mainly includes two key parts, namely the TV evaluation model and the dual dynamic access control mechanism. In the calculation of TVs, including weights or function forms, innovative designs and improvements are made based on previous relevant research [10-11]. The recommended TV d of the system and other registered users h for registered user c is solved as shown in Equation (1) [12].

$$AB_c(D) = \zeta AB_c^g(D) + \psi \sum_h \sin\left(B_h(d-D)\right) \cdot AB_c^h(D) \quad (1)$$

In Equation (1), both ζ and ψ are influence weights, and the sum of the two is 1. $AB_c^g(D)$ represents the recommended TV of the system for registered user c

, D is the time interval, and d represents the specific time. The weight is $\sin\left(B_h\left(d-D\right)\right)$, and the value range is $[0,1]$. B is a TV, and $B_h\left(d-D\right)$ represents the TV of h under the previous evaluation D. $AB_c^h\left(D\right)$ represents the recommended TV of h towards registered user c. In addition, in the calculation of user recommendation TVs, normalization methods need to be used. The reason for using normalization method is to eliminate dimensional differences and dimensional influences between different data sources or indicators. There is a theoretical basis for using Z-score normalization on the minimum maximum scale. Z-score normalization can make data from different sources comparable [13]. The conditions for using this method include normal distribution of data, elimination of dimensional influence, and standardization of data range. The expression of Z-score normalization is in Equation (2) [14].

$$I=\frac{a-b}{G} \tag{2}$$

In Equation (2), I represents the processed data and a is the actual data. b and G are the mean and standard deviations of the dataset, respectively. The expression of range normalization is in Equation (3).

$$I=\frac{a-\min}{\max-\min} \tag{3}$$

In Equation (3), $\max$ and $\min$ represent the maximum and minimum values in the dataset, respectively. In the solution of system recommendation TV, the behavior feature vector of c is expressed as shown in Equation (4) [15].

$$J_c=\left(j_{c,1},j_{c,2},.....,j_{c,k}\right) \tag{4}$$

In Equation (4), $j_{c,1}$ represents the behavior feature of the first row and k is the total number of user behavior features. Therefore, the expression of $AB_c^g\left(D\right)$ is in Equation (5).

$$AB_c^g\left(D\right)=f\cdot\left(L\square J_c\right) \tag{5}$$

In Equation (5), $f\left(\cdot\right)$ represents the activation function, L represents the weight of behavioral features, and $L\square J_c$ is the inner product of vectors. The expression of $L\square J_c$ is in Equation (6).

$$L\square J_c=L_1\square j_{c,1}+L_2\square j_{c,2}+.....+L_k\square j_{c,k} \tag{6}$$

In equation (6), L_k represents the weight of the k th behavioral feature. The solution of $AB_c^h\left(D\right)$ is in Equation (7).

$$AB_c^h\left(D\right)=\frac{M_n}{M_n+P_n} \tag{7}$$

In Equation (7), n represents the interaction process, M_n and P_n respectively represent the number of times the operation of c in n is judged as honest or dishonest. In addition, the solution for the TV $NDD_c\left(D\right)$ related to the c attribute information is in Equation (8).

$$NDD_c\left(D\right)=\upsilon_1\cdot F_1\left(N_{1,s_1}\right)+\upsilon_2\cdot F_2\left(N_{2,s_2}\right)+...+\upsilon_R\cdot F_R\left(N_{R,s_R}\right) \tag{8}$$

In Equation (8), R represents the total number of categories of attributes, υ_1, υ_2, and υ_R all represent influence weights, and their sum is 1. N is the attribute value, and N_{R,s_R} represents the s_R th attribute value in the R th class attribute. F_1, F_2, and F_R represent conversion functions. s is the number of attributes of a certain type. In addition, different conversion functions are expressed uniformly, as shown in Equation (9).

$$F_1\left(N\right)=F_2\left(N\right)=...=F_R\left(N\right)=e^N-1 \tag{9}$$

Therefore, the solution for the comprehensive TV $B_c\left(D\right)$ is in Equation (10).

$$B_c\left(D\right)=\omega_1\cdot B_h\left(d-D\right)+\omega_2\cdot AB_c\left(D\right)+\omega_3\cdot NDD_c\left(D\right) \tag{10}$$

In Equation (10), ω_1, ω_2, and ω_3 represent the influence weights, and the sum of the three is 1. The process of calculating the comprehensive TV is as follows: first initialize the user TV and assign a set of attributes, then collect user behavior data in real time, weight and sum historical TVs (calculated based on behavior records), normalize recommended TVs (system and other user recommended inputs), and attribute-related TVs (calculated based on the correlation between attributes and data) to obtain the comprehensive TV. Finally, dynamically adjust user permissions based on this, and compare TVs with thresholds to determine authorization during resource access. The pseudocode of the TV evaluation model is shown in Figure 3.

```
function calculateCompositeTrust(user):
    historicalTrust = user.getHistoricalTrust()
    recommendedTrust = calculateRecommendedTrust(user.getRecommendations())
    attributeTrust = calculateAttributeTrust(user.Attributes)
    weight1 = 0.4
    weight2 = 0.3
    weight3 = 0.3
    compositeTrust = weight1 * historicalTrust + weight2 * recommendedTrust +
weight3 * attributeTrust
    return compositeTrust

function calculateRecommendedTrust(recommendations):
    normalizedRecommendations = normalize(recommendations)
    return normalizedRecommendations

function calculateAttributeTrust(attributes):
    attributeTrust = 0.0
    for attribute in attributes:
        attributeTrust += attribute.value
    return attributeTrust / len(attributes)
```

Figure 3: Pseudo code for TV evaluation model

From Figure 3, it can be seen that the pseudocode defines how to calculate the user's comprehensive TV. Subsequently, the generated comprehensive TV is input into the dual dynamic access control mechanism section. In the dual dynamic access control mechanism, the core is permission determination, and the determination content is to compare the comprehensive TVs of users and match the access policies of the resources to be accessed. The

pseudocode of the dual access control mechanism is shown in Figure 4.

```
function dualDynamicAccessControl(userRequest, resourcePolicy):
    compositeTrust = calculateCompositeTrust(userRequest.user)
    if compositeTrust < trustThreshold:
        return ACCESS_DENIED
    if not userRequest.attributes.match(resourcePolicy):
        return ACCESS_DENIED
    return ACCESS_GRANTED
```

Figure 4: Pseudo code for dual access control mechanism

From Figure 4, it can be seen that the pseudocode describes the core logic of the dual dynamic access control mechanism. In addition, the dynamic management method of user permissions also involves factors that affect the overall TV, and this part is the foundation of two key parts that need to be provided with relevant user information. Therefore, the overall process of the user permission dynamic management method is in Figure 5.

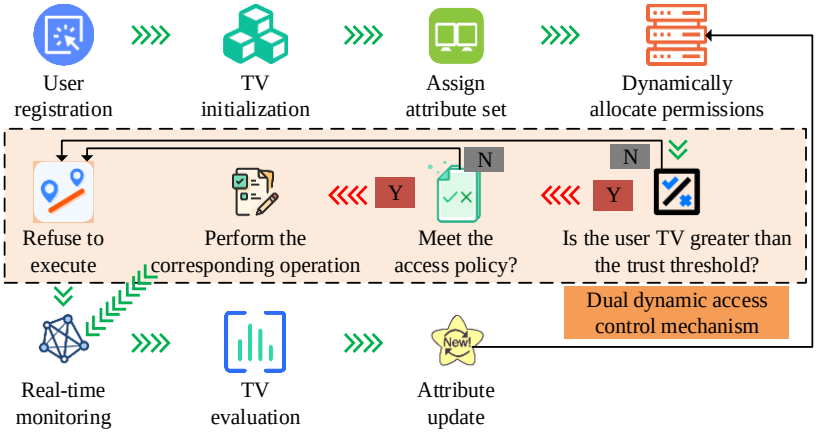


Figure 5: The overall process of dynamic management method for user permissions

From Figure 5, the first step of the user permission dynamic management method is to register the user, and the second step is to initialize the TV. The third step is to allocate attribute sets, and the fourth step is to dynamically allocate permissions. The fifth step is to enter the dual dynamic access control mechanism. In this mechanism, the first step is to determine whether the user's TV is greater than the trust threshold. If it is greater than the trust threshold, the next step is to proceed. Otherwise, execution is rejected and real-time monitoring is initiated. Secondly, it is necessary to determine whether the access policy is met. If it is met, the corresponding operation should be executed before proceeding to real-time monitoring. Otherwise, it will directly proceed to real-time monitoring. The sixth step is to evaluate the TV, the seventh step is to update the attributes, and then return to dynamically assigning permissions.

In addition, in the SE-based user permission dynamic management method, the dual dynamic access control mechanism closely interacts with the underlying SE operations. Specifically, when a user initiates a search request, the system first performs a TV check on the client side, compares the user's comprehensive TV with a preset trust threshold, and verifies whether it complies with the resource access policy. This process occurs before generating the search token, ensuring that only trusted users can create valid search tokens. If the user passes the TV check, the client will generate a search token based on the user's attributes and permissions, and send it to the cloud server. After receiving the search token, the cloud server performs a search operation and returns a list of encrypted files that match the token. After obtaining the encrypted file, the user performs a TV check again to ensure that they still have the authority to decrypt the data. This TV-based dynamic access control mechanism runs through the entire search and retrieval process of SE, ensuring data security and dynamic access.

2.2 Design of ABSE scheme Using B-Tree index structure

A dynamic user permission management method for SE has been designed to address the security management issues of data storage on cloud servers. However, in response to the improvement of SE technology, existing research has proposed ABSE technology that combines SE technology with attribute-based encryption. ABSE technology allows data owners to encrypt data based on its attributes, supports fine-grained access control and private keyword search, and ensures data security [16-17]. However, this technology still has certain shortcomings, such as rigid access policies and low search efficiency. In order to address these issues and further improve ABSE technology, methods that can supporting fine-grained access control have been researched and designed. The system model of this method is in Figure 6.

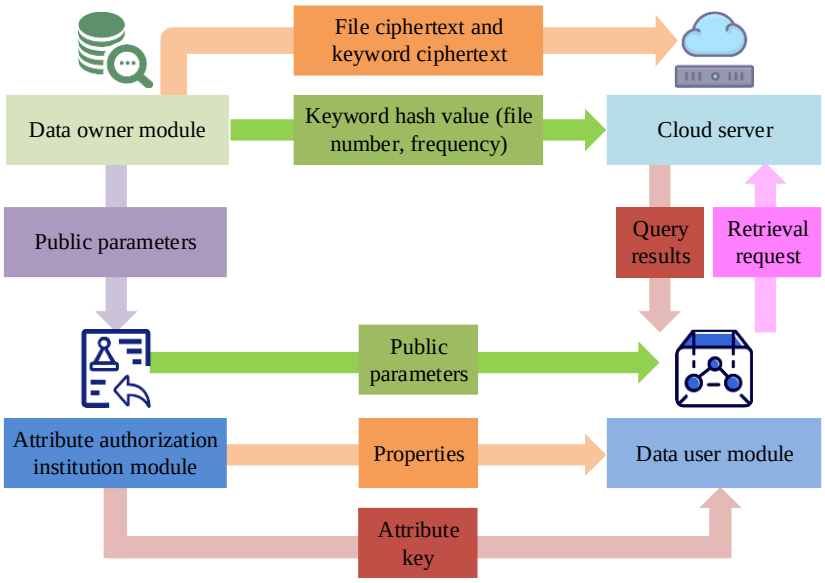


Figure 6: System model supporting fine-grained access control methods

From Figure 6, the system model supporting fine-grained access control methods mainly includes the data owner module, data user module, cloud server, attribute authorization authority module, public parameters, attribute sets, attribute keys, file ciphertexts and keyword ciphertexts, keyword hash values (file numbers, frequency), query results, and retrieval requests. Among them, the attribute authorization authority module needs to generate a system master key and system initialization, and the cloud server needs to run a search algorithm. Meanwhile, the data owner module needs to run encryption algorithms, while the data user module needs to obtain ciphertext files corresponding to keywords, and this process requires running search token generation algorithms.

In the construction process of supporting fine-grained access control methods, the B-Tree index structure is adopted to optimize search efficiency and search result accuracy. The B-Tree index structure is an efficient search tree, mainly used to accelerate data retrieval. It has the advantages of high data retrieval performance, support for ordered range queries and dynamic adjustments, and is widely used in database systems, in memory databases, and caching systems [18-19]. The construction process of B-Tree index structure is in Figure 7.

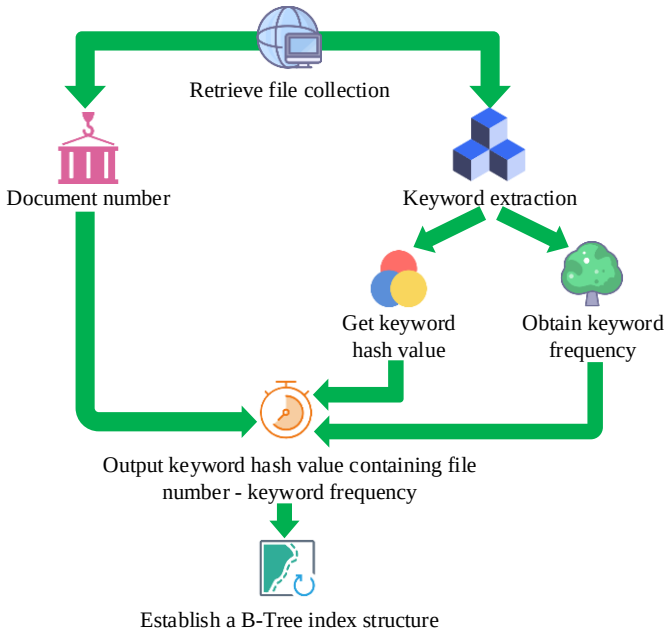


Figure 7: The construction process of B-Tree index structure

From Figure 7, the first step in constructing the B-Tree index structure is to obtain a set of files, and the second step involves extracting file numbers and keywords separately. Meanwhile, in keyword extraction, it is necessary to obtain keyword frequency and keyword hash value. The third step is to output a keyword hash

value containing the file number and keyword frequency, and the fourth step is to establish a B-Tree index structure. The pseudocode of B-Tree structure is shown in Figure 8.

```
class BTreeNode:
    def __init__(self):
        self.keys = []
        self.children = []
        self.is_leaf = True

function insertIntoBTree(node, key):
    if node.is_leaf:
        node.keys.append(key)
        node.keys.sort()
    else:
        child = findAppropriateChild(node, key)
        insertIntoBTree(child, key)
    if len(node.keys) > max_keys:
        splitNode(node)

function searchInBTree(node, key):
    if node.is_leaf:
        return key in node.keys
    else:
        child = findAppropriateChild(node, key)
        return searchInBTree(child, key)
```

Figure 8: Pseudo code of B-Tree structure

From Figure 8, the pseudocode defines the B-Tree node structure and implements insertion and search operations. In addition, when generating the system master key, it is necessary to first initialize and generate public parameters. The T expression of the system master key is in Equation (11) [20].

$$T = \left(\tau, \{u_v\}_{v=1}^{V}\right) \tag{11}$$

In Equation (11), τ represents a random element and u_v represents the random value corresponding to the attribute. v is the serial number of the attribute, and V represents the total number of attributes. τ and u_v have $\tau \in Z_p^*$ and $u_v \in Z_p^*$ respectively, and Z_p^* is a special symbol [21]. The expression of system public parameters is in Equation (12).

$$U = \left(w, w_1, e(w,w)^{\tau}, \{x_v\}_v^V, \ Y\right) \tag{12}$$

In Equation (12), both w and w_1 are generators of the multiplicative cyclic group, $e(w,w)^{\tau}$ represents bilinear pairing operations. Y is the hash algorithm, and x_v represents the group elements generated from the attribute set. In addition, the generation method of user attribute key X is in Equation (13).

$$X = \left(\sigma, \{\sigma_\vartheta\}_{\vartheta=1}^{\theta}\right) \tag{13}$$

In Equation (13), θ represents the number of user attributes and ϑ is the user attribute number. σ represents the authorization corresponding to user attributes. Among them, "authorization" σ refers to the authorization information of the attribute authorization agency for user attributes. It is employed together with attribute values and master keys to export user attribute keys. The attribute authorization agency uses the random elements and corresponding random values in the system master key, combined with user attribute information, to calculate the user attribute related key part through one-way hash function and key generation algorithm. This ensures that only authorized users can obtain the correct attribute key to decrypt or generate valid search tokens, ensuring both user attribute confidentiality and system security and fine-grained access control.

The study explains the implementation details of the B-Tree index structure, including order, balancing strategy, and concurrency control. In terms of sequence, a B-Tree index is constructed by extracting file numbers and keywords from the file set, storing keywords and their file information in order, and recursively locating when inserting new keywords to maintain order. In regard to balancing strategy, node splitting and merging operations are used to maintain balance, ensuring that the number of node keywords does not exceed the preset upper limit. In terms of concurrency control, existing technologies such as locking or timestamp sorting mechanisms are combined to manage concurrent access and updates, ensuring data consistency and integrity during multi-user operations.

The linear secret sharing matrix is one of the specific implementation details of LSSS, which describes in detail how secrets are divided and reconstructed from authorized shares. This matrix driven approach ensures the security and flexibility of the solution, enabling it to adapt to various complex access control requirements. The design and parameter selection of matrices directly affect the efficiency and security of the scheme, and are the foundation for implementing fine-grained access control and secure multi-party computation. Therefore, in the encryption stage, a linear secret sharing matrix was used to achieve access control for users. Linear secret sharing matrix is a technology based on matrix operations to achieve secret information segmentation and recovery. It has the advantages of high security, strong flexibility, good scalability, and supports dynamic adjustment. It finds extensive utility in confidential file transmission, cloud computing and privacy protection, IoT security, and blockchain security [22-24]. Therefore, the ϖ expression of the complete ciphertext is in Equation (14).

$$\varpi = (\mu, \lambda) \tag{14}$$

In Equation (14), μ represents symmetric encrypted ciphertext and λ represents keyword ciphertext. The expression of the search token η is in Equation (15).

$$\eta = \prod_{\varepsilon}^{\delta} w_1^{Y(\varphi_\varepsilon)} \tag{15}$$

In Equation (15), φ_ε represents the keyword and ε is the sequence number of the keyword. δ represents the total number of keywords, and $Y(\varphi_\varepsilon)$ represents the hash value of φ_ε. To address issues such as keyword conflicts, copying, and revocation, researchers have developed a series of processing methods. When keyword conflicts arise, the system employs salt hashing to boost the element of unpredictability, thereby mitigating conflicts. Additionally, it records relevant information to

differentiate between keywords. In scenarios where keyword copying occurs, the system assigns unique identifiers to the keywords, updates the information within the B-Tree index, and verifies uniqueness during token generation. When it comes to keyword revocation, the system utilizes version control tags to revoke the specific version and simultaneously updates the ciphertext file information.

In the ABSE framework, B-Tree stores file keywords and attribute information to support attribute access control. When encrypting files, the data owner stores keyword and attribute information in the B-Tree index. When users search, the attribute authorization agency generates a search token containing user attributes and access policies. The cloud server uses B-Tree to locate files, verifies whether user attributes meet policies through LSSS, and decides whether to allow or deny access. To determine user access permissions, cloud servers need to combine η and X. To form a complete encryption scheme, the two designed schemes will be integrated into the encryption system together. Figure 9 presents the comprehensive structure of the encryption system.

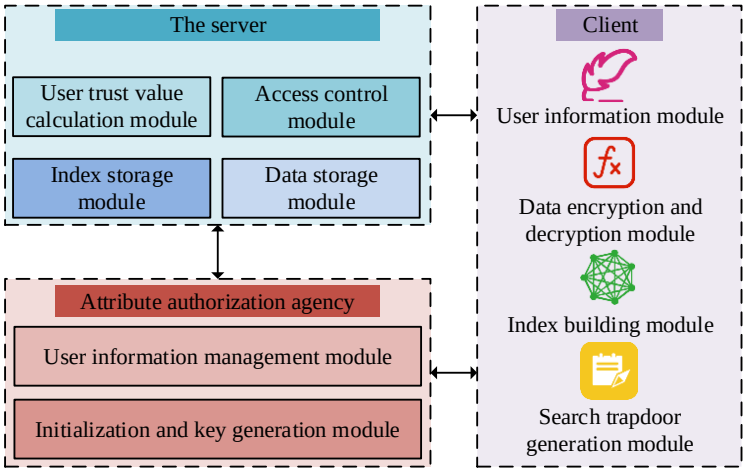


Figure 9: Comprehensive structure of encryption system

From Figure 9, the comprehensive structure of the encryption system mainly consists of three main parts, namely the client, server, and attribute authorization authority. Meanwhile, all three parts are interconnected with each other. In addition, the client includes a user information module, a data encryption and decryption module, an index construction module, and a search trapdoor generation module. The server consists of four main modules, and the most critical one is TV calculation. Finally, the user information management module and the initialization and key generation module constitute the attribute authorization authority. The sequence diagram of the encryption system is shown in Figure 10.

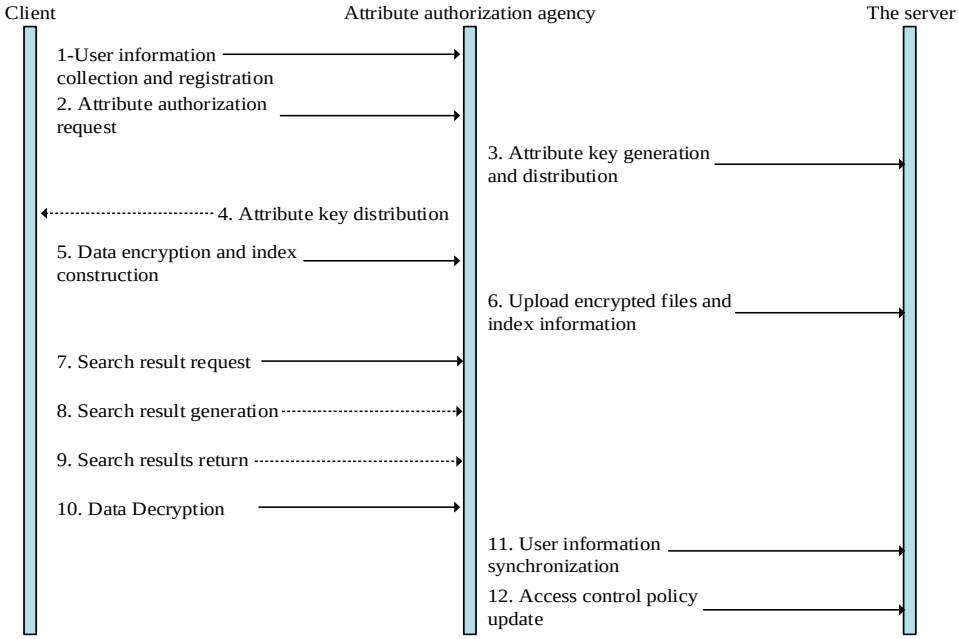


Figure 10: Sequence diagram of encryption system

From Figure 10, after collecting user information and registering, the client requests attribute authorization from the attribute authorization agency. The attribute authorization agency generates attribute keys and distributes them to clients, who use them to encrypt data and build indexes, and then upload the encrypted files and indexes to the cloud server. After the client initiates a search request, the cloud server returns the search results, and the client decrypts the results. Meanwhile, the cloud server synchronizes user information with the attribute authorization agency, which updates access control policies and notifies the cloud server to ensure system data security and the effectiveness of access control.

In the final plan, the TV-based dynamic access control mechanism enhances rather than replaces the attribute based mechanism. When making access decisions, first verify the user's identity and attributes, and then evaluate the TV. If the TV exceeds the threshold and the attributes comply with the policy, access is allowed; otherwise, access is denied. This combination enhances system security and flexibility, effectively protecting cloud server data.

3 Results

To confirm the capability of the research design scheme, the experimental environment was first described, and then the comparative scheme was selected. In regard to comparative indicators, the study selected factors such as time consumption, memory usage, Central Processing Unit (CPU) utilization, and traffic analysis.

3.1 Performance verification of dynamic management methods for user permissions

To confirm the performance of the dynamic management method for user permissions in the research design, experiments were conducted using Python programming language combined with PyCrypto and OpenSSL cryptographic libraries. The paper utilized the PyCrypto library to implement encryption methods to ensure data confidentiality and integrity, and used the SSL/TLS protocol supported by the OpenSSL library to ensure communication security and prevent data from being stolen or tampered with during transmission. The operating system was Windows 10, the CPU was Intel Core i5-12600KF, the main frequency and dynamic acceleration frequency were 3.7GHz and 4.9GHz, respectively, and the number of cores was ten. The maximum memory bandwidth was 76.8GB/s. The study established a threat model that included data leakage, unauthorized access, data tampering, insider threats, and man in the middle attacks. Attackers may have the ability to intercept data transmission, crack encryption algorithms, forge user identities, tamper with stored data, and exploit system vulnerabilities. The types of attacks that the system can defend against include data leakage, unauthorized access, data tampering, insider threats, and man in the middle attacks. The specific defenses are as follows: the system uses strong encryption algorithms to prevent data leakage, resists unauthorized access through fine-grained access control and dynamic trust evaluation, uses digital signatures and integrity verification to ensure data integrity, follows minimum privilege and audit mechanisms to prevent internal threats, and uses secure communication protocols to prevent man in the middle attacks. In addition, there are two trusted vacation options for servers or clients, namely: cloud servers are considered semi trusted and must perform operations according to the protocol, and ensure the security of the operating environment through regular audits and vulnerability scans. The client is generally trustworthy, but identity authentication and trust assessment are still required, while ensuring local security to prevent malicious software from stealing user credentials or keys. In regard to comparative methods, the study selected four schemes and represented them using Scheme A, Scheme B, Scheme C, and Scheme D. Among them, Scheme A is an access control scheme based on roles and user TVs, dynamically adjusting user permissions by defining roles and assigning TVs. Scheme B is a trust management system for wireless sensor networks based on Bayesian trust model. This scheme utilizes the principle of Bayesian networks to construct a trust model based on users' historical interaction behavior, recommendation information, etc., and calculates the trust relationship between users through probabilistic inference to determine access permissions. Scheme C is a dynamic access control and authorization system based on zero trust, which requires strict authentication of user identity and dynamic granting of minimum permissions for each access. Scheme D is a hybrid access control scheme based on attributes and TVs. This scheme integrates user attributes and TV evaluation, and users need to meet attribute requirements and TV standards to access resources. Meanwhile, TV comprehensively considers factors such as user attribute relevance and behavior history. The reason for choosing these four methods was that they cover different trust evaluation and access control mechanisms, which can be effectively compared with the paper design scheme, highlighting the advantages of the paper scheme in regard to TV evaluation accuracy, access control flexibility, and system performance. In regard to parameter values, there were four possible values for ω_1 in the research design scheme, namely 0.8, 0.6, 0.4, and 0.2. Meanwhile, the initial TV was set to 0.5. In addition, the initial TVs for schemes A, B, C, and D are also 0.5. The comparison of TV calculation time and access control determination time for different schemes is in Figure 11.

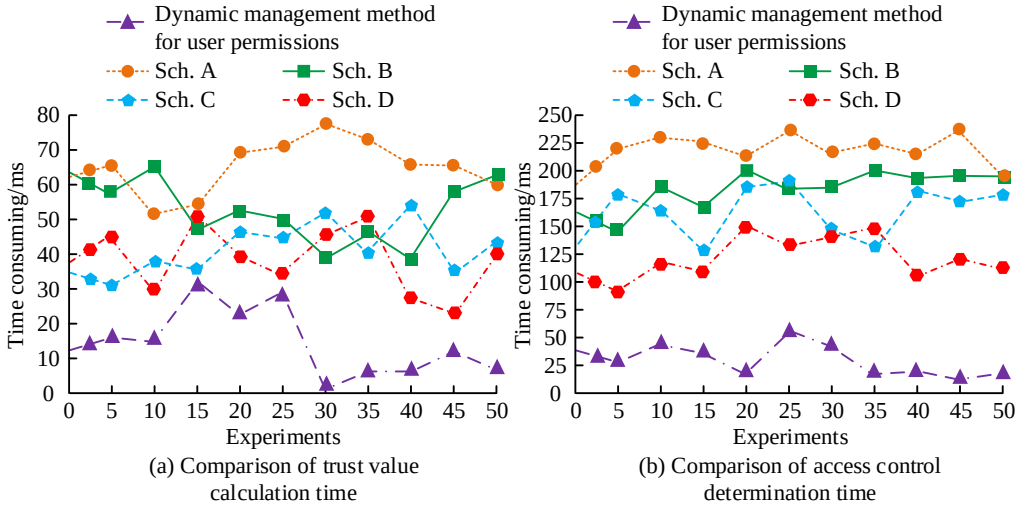


Figure 11: Comparison of TV calculation time and access control determination time for different schemes

From Figure 11 (a), in regard to the comparison of TV calculation time, the research-designed user permission dynamic management method performed better, with corresponding maximum and minimum values of 32 ms and 0.6 ms, respectively. Meanwhile, the maximum TV calculation time of Scheme A, Scheme B, Scheme C, and Scheme D was 78 ms, 65, 54 ms, and 51 ms, respectively, which were 46 ms, 33 ms, 22 ms, and 19 ms higher than 32 ms. From Figure 11 (b), when comparing the access control judgment time of different schemes, the maximum value of the research designed user permission dynamic management method was 56 ms, while the maximum values of the four comparison schemes were 237 ms and 201 ms, respectively. The differences between 189 ms, 150 ms, and 56 ms were 181 ms, 145 ms, 133 ms, and 94 ms, respectively. In summary, the TV calculation and access control determination time of the research and design user permission dynamic management method were both lower, and the performance was better. The rationality comparison of TVs for different schemes is in Figure 12.

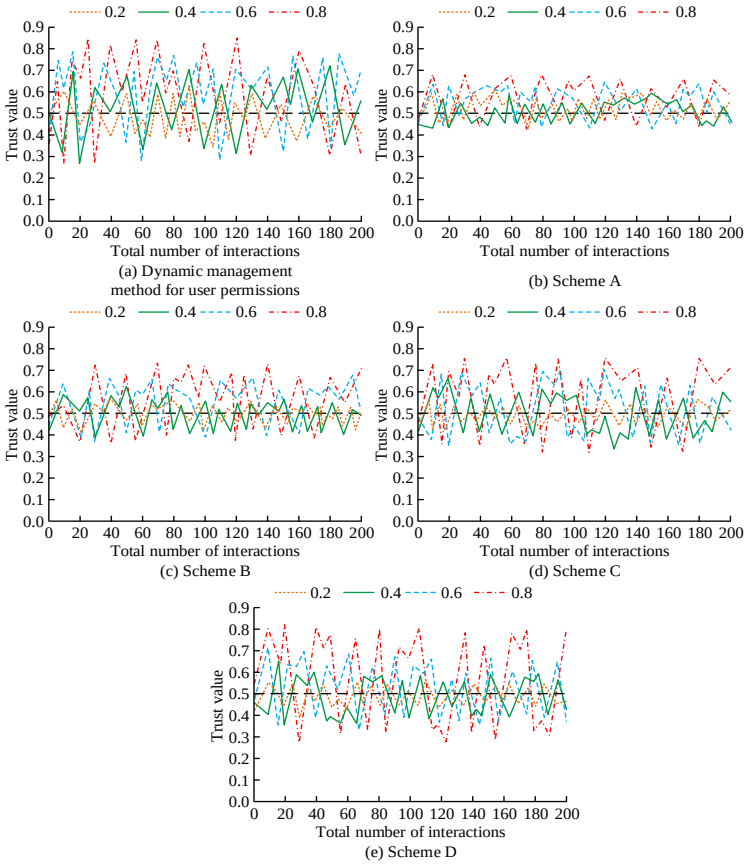
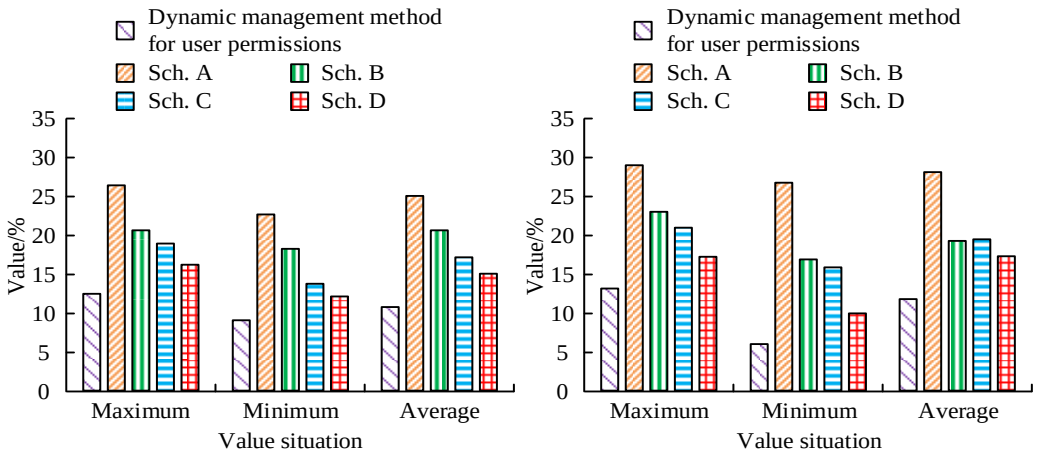


Figure 12: Comparison of the rationality of TVs for different schemes

According to Figure 12 (a), in the user permission dynamic management method designed for research, the fluctuation range of TVs for random users under different weight values was [0.227, 0.845], and the difference between their maximum and minimum values was 0.618. From Figure 12 (b), Figure 12 (c), Figure 12 (d), and Figure 12 (e), in Scheme A, Scheme B, Scheme C, and Scheme D, the fluctuation range of TVs of random users was [0.423, 0.698], [0.385, 0.725], [0.336, 0.761], and [0.285, 0.802], respectively. The difference between the maximum and minimum values was 0.275, 0.340, 0.425, and 0.517, all of which were less than 0.618. The comparison of CPU utilization and memory usage between different schemes is in Figure 13.



(a) Comparison of CPU Utilization (b) Memory usage rate

Figure 13: Comparison of CPU utilization and memory usage among different schemes

From Figure 13 (a), in regard to CPU utilization comparison, the maximum value for studying and designing user permission dynamic management methods was 12.17%, and the minimum value was 8.9%. Meanwhile, the maximum CPU utilization rates of the four comparison schemes were 26.98%, 20.17%, 18.55%, and 16.29%, respectively, which were 14.81%, 8.00%, 6.38%, and 4.12% higher than 12.17%. From Figure 13 (b), when comparing the memory occupancy rates of different schemes, the user permission dynamic management method designed by the research ranked first, followed by Scheme D, Scheme C, Scheme B, and Scheme A in the second to fifth positions. Meanwhile, according to the ranking order, the maximum memory occupancy rates of the five schemes were 13.95%, 17.52%, 20.31%, 23.17%, and 28.55%, respectively. In summary, the research and design of dynamic management methods for user permissions had lower CPU utilization and memory usage, and better performance. To further validate the performance of the research designed user permission dynamic management method, encryption evaluations were conducted on both the method and the comparison method, involving security proof indicators and resistance indicators to known attacks. The encryption evaluation results of different methods are shown in Table 2.

Table 2: Encryption evaluation results of different methods

Scheme	Safety certificate					Resistance to known attacks				
	Number of experiments					Number of experiments				
	1	2	3	4	5	1	2	3	4	5
Scheme A	64	68	68	67	61	68	65	70	61	62
Scheme B	66	68	66	66	66	66	73	67	71	67
Scheme C	79	82	79	81	77	77	77	80	82	81
Scheme D	86	81	80	89	82	87	80	83	82	85
Manuscript	93	91	96	92	93	96	97	98	95	94

From Table 2, in the five experiments of security proof, the scores of the research design user permission dynamic management method were all above 90 points, with the highest reaching 96 points, significantly higher than other comparative schemes. In regard to resistance to known attacks, the research and design of user permission dynamic management methods also scored above 90 points, with the highest reaching 98 points, demonstrating extremely strong resistance to attacks. In contrast, Scheme A and Scheme B scored lower on both indicators, while Scheme C and Scheme D performed well but still did not reach the level of research design methods.

3.2 Performance validation supporting fine-grained access control methods

To validate the performance of fine-grained access control methods, the same experimental setup was used in the study. In addition, in regard to comparative methods, the study selected four schemes and named them Scheme 1, Scheme 2, Scheme 3, and Scheme 4 respectively. Among them, Scheme 1 is a traditional symmetric SE scheme, which uses symmetric encryption algorithms to encrypt data and constructs an index structure for searching. Option 2 is the ABSE scheme, which implements fine-grained access control through attribute

encryption and SE techniques. Scheme 3 is a dynamic access control SE scheme based on zero trust. This scheme combines a zero trust security model and SE technology. Every time a user initiates a search request, their identity must be re verified and dynamically evaluated, and the evaluation results are used to determine whether to allow the user to search and access data. Scheme 4 is a blockchain based SE scheme, which utilizes blockchain to record data hash and permission information, and combines SE to achieve technical security retrieval. The reason for choosing these four methods was that they represent encryption and access control methods of different technological routes, which can comprehensively compare the advantages of this research scheme in regard to encryption efficiency, access control flexibility, security, and communication cost. Meanwhile, these solutions all included five stages: system initialization, key generation, encryption, search token generation, and search. In addition, there were a total of 6 values for the number of user attributes, namely 1, 6, 11, 16, 21, and 26, and the access policy involved the same number of attributes. The comparison of time consumption for different parts is in Figure 14.

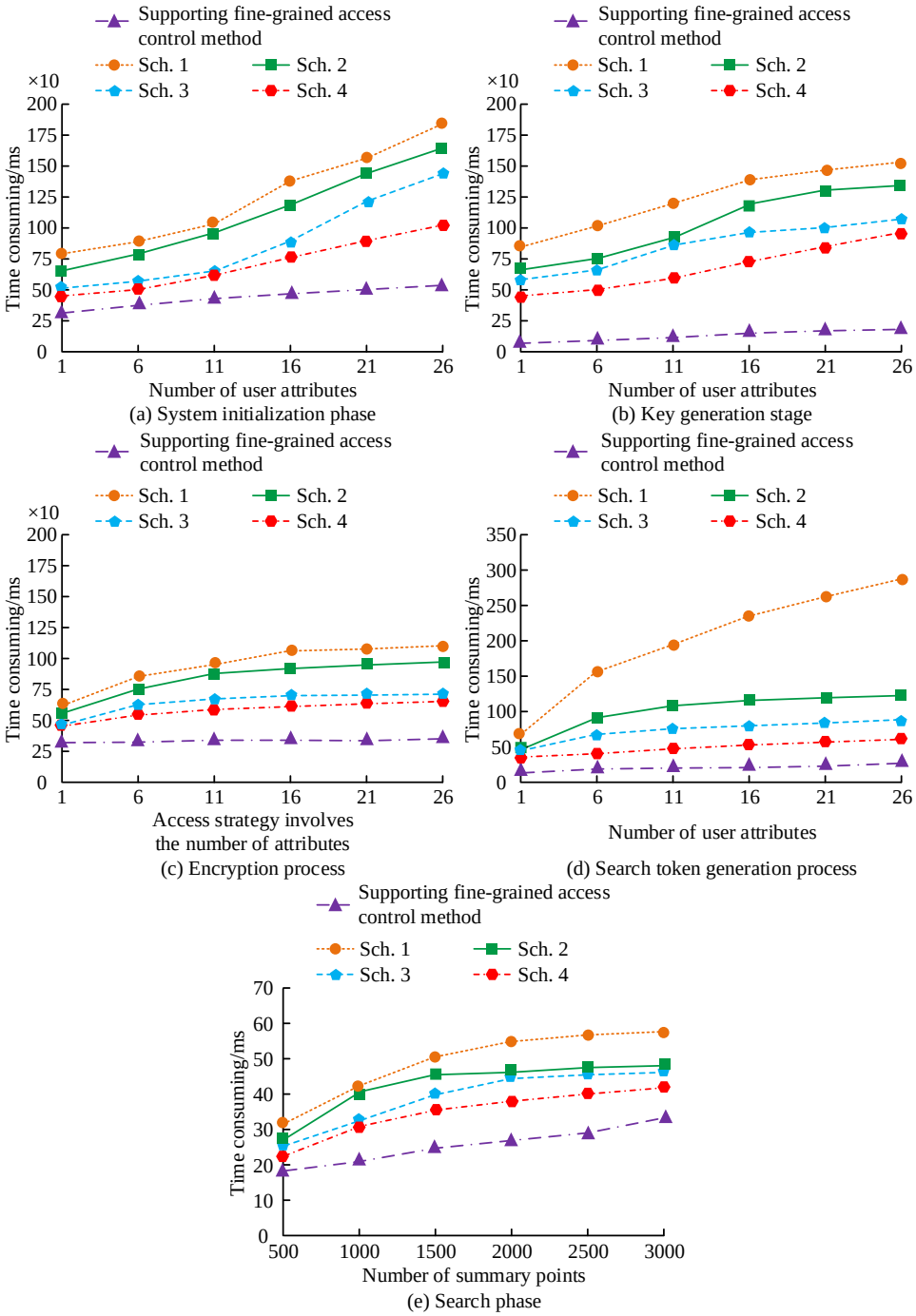


Figure 14: Comparison of time consumption for different parts

According to Figure 14 (a), during the system initialization phase, as the number of user attributes increased, the corresponding time consumption for different schemes also increased synchronously. The increase speed of the fine-grained access control method designed in this study was slower, while the increase speed of Scheme 1 was faster. In addition, in regard to specific values, the max time consumption of schemes 1, 2, 3, and 4 was 1852 ms, 1652 ms, 1400 ms, and 1050 ms, respectively, while the max time consumption of the research-designed fine-grained access control method was 580ms. As shown in Figure 14 (b), in the key generation stage, the max time consumption of the five schemes was 180 ms, 1558 ms, 1327 ms, 1106 ms, and 985 ms, respectively. From Figure 14 (c), in the encryption stage, the research-designed supporting fine-grained access control method performed better, followed by schemes 4, 3, and 2, and finally scheme 1. Meanwhile, the max encryption time for the five schemes was 325 ms, 657 ms, 739 ms, 962 ms, and 1107 ms, respectively. The trend of encryption time for the five schemes increased synchronously with the increase in the number of attributes involved in the access policy. From Figure 14 (d), in the search token generation stage, the max time consumption of the research-designed supporting fine-grained access control method and the four comparison schemes were 25 ms, 287 ms, 120 ms, 85 ms, and 60 ms, respectively. From Figure 14 (e), in the search stage, the max time consumption of the five schemes were 33 ms, 57 ms, 48 ms, 46 ms, and 42 ms, respectively. Overall, the research-designed supporting fine-grained access control method had shorter time consumption and better performance. The comparison of communication volume between different schemes is in Table 3.

Table 3: Comparison of communication volume between different schemes

Scheme	Number of keywords					
	1	2	3	4	5	6
Scheme 1	3.336	4.985	6.226	7.679	9.017	10.369
Scheme 2	3.017	4.119	5.654	7.301	8.435	10.195
Scheme 3	2.778	3.657	4.753	6.432	8.003	9.665
Scheme 4	2.374	2.892	4.147	6.158	7.553	9.448
Supporting fine-grained access control methods	1.137	2.015	2.225	2.785	3.012	3.257

In Table 3, the communication volume is represented by the amount of data, and the unit is Byte. From Table 3, as the number of keywords increased, the amount of data corresponding to different schemes also increased synchronously. Within the same range of keyword changes, the data volume of scheme 2 changed more significantly, about 7.178 Byte. Meanwhile, the data volume changes for Scheme 1, Scheme 3, Scheme 4, and the research-designed supporting fine-grained access control method were 7.033 Byte, 6.887 Byte, 7.074 Byte, and 2.120 Byte, respectively. The research-designed supporting fine-grained access control method had smaller changes in data volume, which were 4.913 bytes, 5.058 bytes, 4.767 bytes, and 4.954 bytes less than the comparison scheme, respectively. This method had lower communication volume and better performance. The storage overhead comparison of different schemes is in Table 4.

Table 4: Comparison of storage costs for different solutions

Scheme	User key storage overhead/KB					Cryptocurrency storage overhead/KB				
	User attribute set size					Leaf node set size				
	5	10	15	20	25	10	15	20	25	30
Scheme 1	3.45	4.73	5.69	8.71	8.98	5.13	6.08	7.32	8.65	9.17
Scheme 2	3.13	4.03	5.36	8.42	8.66	4.65	5.73	6.54	8.42	8.88
Scheme 3	2.65	4.12	4.95	7.11	7.79	4.11	4.72	6.36	7.89	8.34
Scheme 4	2.23	3.87	4.67	6.52	7.42	3.89	4.57	5.98	7.42	7.62
Supporting Fine-grained access control method	1.74	2.54	3.89	5.14	6.37	3.34	4.17	5.24	6.85	7.25

From Table 4, in regard to the comparison of user key storage costs, the research-designed supporting fine-grained access control method performed better, with an average value of 3.94 KB. In addition, the average user key storage costs of the four comparison schemes were 6.31 KB, 5.92 KB, 5.32 KB, and 4.94 KB, respectively, which were 2.37 KB, 1.98 KB, 1.38 KB, and 1.00 KB higher than 3.94 KB. In addition, as the size of the random user attribute set increased, the user key storage costs corresponding to different schemes also increased synchronously, and schemes 1 and 2 increase more. Meanwhile, in regard to ciphertext storage overhead, the research-designed supporting fine-grained access control method and the average values of the four comparison schemes were 5.37 KB, 7.27 KB, 6.84 KB, 6.28 KB, and 5.90 KB, respectively. The significance of reducing storage overhead by 2-3KB was twofold: firstly, it lowered the storage cost of cloud servers. Secondly, it reduced the storage pressure on the client and avoided problems caused by insufficient storage. Furthermore, it could improve system efficiency, accelerate key read/write and distribution speed, shorten loading and response time, and enhance user experience. Finally, reducing storage overhead made the system easier to cope with user and data growth, reduce performance bottleneck risks, and benefit system scalability. Overall, the research -designed supporting fine-grained access control method had lower storage overhead and better performance. To further verify the robustness of fine-grained access control methods, long-term performance analysis and stress testing were conducted, with specific indicators being system response time and system throughput. The

robustness comparison of different methods is shown in Table 5.

Table 5: Comparison of robustness of different methods

Scheme	System response time/ms					System throughput/(req/s)				
	Run time/hour					Number of concurrent users				
	12	24	48	72	96	50	100	200	500	1000
Scheme 1	120	135	150	165	180	25	45	65	80	90
Scheme 2	110	120	130	140	150	30	50	70	85	95
Scheme 3	100	110	120	130	140	35	60	80	90	95
Scheme 4	90	100	110	120	130	40	70	90	100	105
Support Fine-grained access control methods	80	90	100	110	120	55	85	120	140	165

From Table 5, methods that support fine-grained access control performed excellently in terms of system response time and throughput. After 96 hours of system operation, its response time was only 120 milliseconds, which was 10-60 milliseconds lower than other solutions. With 1000 concurrent users, the throughput reached 165 times/second, which was 60-75 times/second higher than other solutions, demonstrating better performance stability and system resilience.

In practical applications, a centralized key management scheme was studied to address the issue of handling key management across multiple users: attribute authorization agencies generated and distributed attribute keys uniformly, ensuring that key generation followed consistent policies and was securely distributed to users. Meanwhile, using the user information management module of the attribute authorization agency, user identity and attribute information were centrally managed and authenticated. The system dynamically adjusted the key according to changes in user attributes to ensure that only users with legitimate attributes can obtain the corresponding key. In addition, the system regularly updated keys and set expiration dates to further improve key security and reduce risks caused by key leaks.

4 Discussion

To conduct critical analysis, the study compared the designed method with the techniques in Table 1, and the comparison results are shown in Table 6.

Table 6: Comparison between the designed method and the techniques in Table 1

Comparative method	Quantitative results
Liu et al. [4]	Reduce computing costs by 20% and storage expenses by 10%
Ng et al. [5]	Efficiency increased by 30%, computational cost increased by 8%
Liu et al. [6]	40% increase in computational efficiency and 25% increase in storage overhead
Lu et al. [7]	Efficiency increased by 35%, storage overhead increased by 20%
Dynamic management method for user permissions	Reduce computing costs by 30% and storage expenses by 20%
Support Fine-grained access control methods	Reduce computing costs by 25% and storage expenses by 15%

From Table 6, the proposed method for dynamically managing user permissions reduced the overhead of permission determination and resource allocation through comprehensive trust evaluation and dual dynamic access control. Meanwhile, it optimized data retrieval efficiency using B-tree indexing, thereby reducing computation and storage costs. In addition, methods that support fine-grained access control utilize linear secret sharing schemes to achieve precise access control and reduce resource consumption during encryption and retrieval processes. However, in some cases, such as when the number of users increased sharply or the data scale expands significantly, the system may face performance pressure and need to balance security and efficiency.

5 Conclusion

The research designed a user permission dynamic management scheme and a fine-grained access control scheme for SE technology to address the security management issues of cloud server data storage, and developed a complete encryption scheme. The results showed that the max and min TV calculation time of the user permission dynamic management scheme were 32 ms and 0.6 ms, respectively, and the max access control determination time was 56 ms, which was 181 ms, 145 ms, 133 ms, and 94 ms lower than the max value of the comparison scheme. The user permission dynamic management scheme had a shorter time consumption, which might be due to its use of a comprehensive TV evaluation model and a dual dynamic access control mechanism, the introduction of an efficient mathematical model, and the significant improvement in computational efficiency. In regard to memory and CPU usage, dynamic management of user permissions also presented significant advantages. The max time consumption for supporting fine-grained access control method at different stages was 1852 ms, 180 ms, 325 ms, 25 ms, and 33 ms, respectively, which was significantly better than the comparative scheme. This might be due to its adoption of B-Tree indexing structure and linear secret sharing scheme, which optimized search efficiency and flexibility of access control. In regard to communication volume comparison, the supporting fine-grained access control method had a smaller change amplitude, with a value of 2.120 Byte, which as 4.913 Byte, 5.058 Byte, 4.767 Byte, and 4.954 Byte smaller than the comparison scheme, respectively. In summary, the overall encryption scheme

designed in the research had good performance. However, there are also certain shortcomings in the research. The linear secret sharing scheme brought high flexibility but also high complexity, mainly reflected in the complexity of the access structure and the computational cost of matrix operations. Complex access structures may lead to reduced efficiency in processing large-scale data, while frequent matrix operations may increase computational overhead. Future research can adopt simpler access structure designs, such as "lightweight LSSS", and introduce distributed computing technology to share computing tasks. In addition, when the number of users or data scale increases sharply, the system may face performance bottlenecks, and TV evaluation models and access control mechanisms may lead to reduced efficiency when deployed in resource constrained environments. Future research can be improved by introducing distributed computing technology, adopting lightweight encryption and access control schemes, machine learning and artificial intelligence technologies, and enhancing the caching mechanism of the system.

References

[1] Zhang Y, Zhu T, Guo R, Xu S, Cui H, Cao J. Multi-keyword searchable and verifiable attribute-based encryption over cloud data. IEEE Transactions on Cloud Computing, 2023, 11(1):971-983. https://doi.org/10.1109/TCC.2021.3119407

[2] Zhang X, Mu D, Zhao J. Attribute-based keyword search encryption for power data protection. High-Confidence Computing, 2023, 3(2):32-39. https://doi.org/10.1016/j.hcc.2023.100115

[3] Prasad S, Rao Y S. Designing secure data storage and retrieval scheme in cloud-assisted Internet-of-Drones environment. IEEE Internet of Things Journal, 2023, 11(8):13734-13751. https://doi.org/10.1109/JIOT.2023.3337265

[4] Liu W, Bai X, She W, Song X, Tian Z. Searchable encrypted data sharing scheme based on inverted index. Computer Engineering and Applications, 2023, 59(10):270-279. https://doi.org/10.3778/j.issn.1002-8331.2201-0159

[5] Ng Z Y, Salam I. Implementation of a blockchain-based searchable encryption for securing contact tracing data. Journal of Internet Technology, 2024, 25(2):241-254. https://doi.org/10.53106/160792642024032502007

[6] Liu P, He Q, Zhao B, Guo B, Zhai Z. Efficient multi-authority attribute-based searchable encryption scheme with blockchain assistance for cloud-edge coordination. Computers, Materials & Continua, 2023, 76(9):3325-3343. https://doi.org/10.32604/cmc.2023.041167

[7] Lu H, Xue X, Zhu Y, Chen C, Han H, Meng S, Lin H. Privacy-preserving SQL database driven by searchable encryption. Chinese Journal of Engineering, 2024, 46(11):2085-2098. https://doi.org/10.13374/j.issn2095-9389.2024.02.07.004

[8] Wang Z, Zhang Q, Meng L, Liu Y L. Secure content-based image retrieval scheme based on deep hashing and searchable encryption. CMC-Computers Materials & Continua, 2023, 75(3):6161-6184. https://doi.org/10.32604/cmc.2023.037134

[9] Souror S, Badawy M, El-Fishawy N. Secure query processing for smart grid data using searchable symmetric encryption. The Journal of Supercomputing, 2024, 80(16):24173-24211. https://doi.org/10.1007/s11227-024-06326-z

[10] Xie P, Li X, Feng T, Zhang M, Zhang P, Li P. A Trust and Risk Adaptive Access Control Model for Internet of Vehicles. International Journal of Network Security, 2024, 26(3):510-520. https://doi.org/10.6633/IJNS.202405_26(3).18

[11] Liu Y, Yang W, Wang Y, Liu Y. An access control model for data security sharing cross-domain in consortium blockchain. IET Blockchain, 2023, 3(1):18-34. https://doi.org/10.1049/blc2.12022

[12] Tong Q, Miao Y, Weng J, Liu X, Choo K K R, Deng R H. Verifiable fuzzy multi-keyword search over encrypted data with adaptive security. IEEE Transactions on Knowledge and Data Engineering, 2022, 35(5):5386-5399. https://doi.org/10.1109/TKDE.2022.3152033

[13] Geem D, Hercules D, Pelia R S, Venkateswaran S, Griffiths A, Noe J D. Progression of Pediatric Crohn's disease is Associated with anti-tumor necrosis factor timing and body Mass Index Z-Score normalization. Clinical Gastroenterology and Hepatology, 2024, 22(2):368-376. https://doi.org/10.1016/j.cgh.2023.08.042

[14] Chinthamu N, Karukuri M. Data science and applications. Journal of Data Science and Intelligent Systems, 2023, 1(1):83-91. https://doi.org/10.47852/bonviewJDSIS3202837

[15] Tian J, Lu Y, Li J. Lightweight searchable and equality-testable certificateless authenticated encryption for encrypted cloud data. IEEE Transactions on Mobile Computing, 2024, 23(8):8431-8446. https://doi.org/10.1109/TMC.2023.3348849

[16] Lu W. Construction of a Secure Sharing Model for Digital Educational Resources Using Blockchain and Cipher Policy Attribute Based Encryption in Smart Education. Informatica, 2024, 48(22):63-74. https://doi.org/10.31449/inf.v48i22.6627

[17] Yang Y, Zhang G, Li S, Liu Z. Offline/online attribute-based searchable encryption scheme from ideal lattices for IoT. Frontiers of Computer Science, 2024, 18(3):239-241. https://doi.org/10.1007/s11704-023-3128-3

[18] Xiong Y, Luo M X. Searchable encryption scheme for large datan sets in cloud storage environment. Radioengineering, 2024, 33(2):223-235. https://doi.org/10.13164/re.2024.0223

[19] Song J, Shen Z, Yu H, Lai R, Li Y, Wang Q, Li J. Secure and efficient multi-keyword fuzzy search over encrypted data on alliance chain. Recent Advances in Electrical & Electronic Engineering, 2024, 17(7):652-665.

https://doi.org/10.2174/0123520965251866230926050714

[20] Alyousif A A, Yassin A A. Improving Performance of searchable symmetric encryption through new information retrieval scheme. Iraqi Journal for Electrical and Electronic Engineering, 2024, 20(1):68-77. https://doi.org/10.37917/ijeee.20.1.7

[21] Hu Z, Deng L, Wu Y, Shi H, Gao Y. Secure and efficient certificateless searchable authenticated encryption scheme without random oracle for industrial internet of things. IEEE Systems Journal, 2023, 17(1):1304-1315. https://doi.org/10.1109/JSYST.2022.3197174

[22] Senthilkumar M, Murugan BS. Enhancing The Security Of An Organization From Shadow Iot Devices Using Blow-Fish Encryption Standard. Acta Informatica Malaysia. 2022; 6(1): 22-24. http//:doi.org/ 10.26480/aim.01.2022.22.24

[23] Talib E, Osipyan V. User Multi Group Key Distribution Using Secret Sharing with Circulate Matrices Based on Structures Pythagors Equation and Ecdh Key Exchange Protocol. Informatica, 2023, 47(5):127-136. https://doi.org/10.31449/inf.v47i5.4658

[24] Liu X H, Huang X Y, Wu W, Ning J T. Key-Policy Attribute-Based Encryption Based on SM9 and Its Fast Decryption. Journal of Computer Science and Technology, 2024, 47(5):971-986. https://doi.org/10.11897/SP.J.1016.2024.00971

Adaptive Weighted Case-Based Reasoning for Intelligent Coal Mine Decision Support Systems

Zhiqiang Zhang[1], Kai Tan[2], Changjun Yang[1], Yong Li[2], Chunhua Huang[2*], Wei Li[1]
[1]National Energy Group Ningxia Coal Industry Co., Ltd. Hongliu Coal Mine, Ningxia 750000, China
[2]Chongqing Institute of China Coal Technology & Engineering Group, Chongqing 400039, China
E-mail: 15000431@ceic.com, tan9446@163.com, 18995218111@189.com, 42948400@qq.com, zzhiqiang1214@163.com, shevl@163.com
[*]Corresponding author

Keywords: coal mine, intelligent decision support system, case reasoning, adaptive weight dynamic case retrieval algorithm, experimental simulation

Received: June 12, 2025

Under the background of intelligent transformation of coal mines, an intelligent decision support system based on case-based reasoning (CBR) has become crucial for improving production control. This paper constructs such a system and innovatively proposes an adaptive weight dynamic case retrieval algorithm (AWDCR). The algorithm leverages real-time monitoring of multi-source production data, dynamically adjusting case attribute weights based on data change characteristics and decision influence through a hybrid AHP-entropy weight mechanism. Using MATLAB simulation with 100,000+ actual production records across 100 scenarios (normal, equipment failure, environmental anomaly), results show AWDCR reduces average retrieval time by 20% and improves decision accuracy from 80% to 90% compared to traditional CBR. enhancing retrieval accuracy by 20%. The system effectively enhances production efficiency and safety, laying a foundation for intelligent coal mining.

Povzetek: Algoritem AWDCR z dinamičnim uteževanjem atributov primerov omogoča bolj kvalitetno podporo odločanju v inteligentnih premogovnikih, saj presega tradicionalni CBR po hitrosti iskanja in točnosti odločitev.

1 Introduction

Coal occupies a pivotal position in the national energy system. As the primary energy source for a long time, it has provided solid power support for industrial development and social operation. However, the traditional coal mine production method is full of drawbacks. From a safety perspective, the complex geological conditions underground and the limitations of manual operation have led to frequent safety accidents, such as gas explosions and water seepage, which seriously threaten miners' lives. Regarding production efficiency, the mining and transportation processes that rely on the workforce are cumbersome, and the degree of equipment coordination is low, resulting in low overall production efficiency [1]. At the same time, a large amount of workforce investment has kept the labor cost high, restricting the improvement of the economic benefits of coal mining enterprises.

Intelligent transformation has become an inevitable choice for the coal mining industry to break through the development bottleneck. By introducing advanced information technology and intelligent equipment, it is possible to monitor the underground environmental parameters and equipment operation status in real-time, warn of safety hazards in advance, and significantly improve the level of safe production [2]. The collaborative operation of automated and intelligent production equipment can dramatically improve production efficiency and reduce time waste in the production process. In addition, applying intelligent systems can dramatically reduce the number of front-line operators and effectively reduce labor costs.

Globally, coal-producing countries such as the United States and Australia have widely applied intelligent technology to achieve highly automated mining [3]. China has also responded positively, and major coal companies have laid out intelligent mine construction. From intelligent coal mining faces to intelligent ventilation and transportation systems, the intelligent development of coal mines is in full swing.

Case-based reasoning (CBR) technology has been applied to a certain extent in the coal mining field due to its unique advantages. In the coal mining process, the relevant system can formulate reasonable mining plans for different geological conditions based on previous mining cases to improve mining efficiency and resource recovery rate [4]. With the help of CBR technology, the intelligent ventilation system refers to historical ventilation cases to control the ventilation volume and ensure underground air quality accurately. In the transportation process, CBR technology can quickly diagnose and solve transportation equipment failures based on past transportation failure cases to ensure smooth transportation.

However, there are still many problems in the existing coal mine system based on CBR technology. The case library update mechanism is lagging, and it is impossible to incorporate new complex working conditions in time, resulting in an outdated system decision-making basis [5]. The case retrieval accuracy is insufficient, and it isn't easy to accurately match the case that best fits the current

working conditions in many cases, affecting the accuracy of decision-making. When faced with complex and changeable working conditions, such as sudden changes in geological conditions and new equipment failures, the system has poor adaptability and cannot provide adequate decision support.

This paper deeply studies the control and key technologies of the intelligent decision support system for coal mines based on CBR. It aims to improve the system's adaptability to complex and changeable working conditions by optimizing the case library update strategy and innovating the case retrieval algorithm. It also aims to build a more efficient and accurate intelligent decision support system, providing strong technical support for coal mine safety production and efficient operation. Specifically, the research targets a 10–15% improvement in retrieval accuracy (from 80% to ≥90%) and a ≥20% reduction in retrieval time (to ≤100ms) in high-gas (≥1%) and high-temperature (≥40°C) scenarios [6].

2 Architecture of intelligent decision support system for coal mines based on CBR

2.1 Overall framework design of the system

2.1.1 Data acquisition layer

The data acquisition layer is the foundation of the intelligent decision support system for coal mines. It is like the "eyes and ears" of the system, providing key information for subsequent decisions. In various production areas of coal mines, multiple sensors are reasonably arranged according to different monitoring needs. In underground mining workers, gas concentration sensors are essential to ensure safe production. Since gas explosion is one of the significant safety hazards in coal mines, gas concentration sensors are usually densely installed in this area to monitor gas concentration changes in real time and accurately. Temperature sensors are distributed near electrical equipment, tunnel walls, goaves, and other locations to monitor equipment operating temperature and ambient temperature to prevent fire or equipment damage caused by overheating. Equipment operating status sensors are installed on key equipment such as coal mining machines, scraper conveyors, and ventilators to collect equipment operating parameters such as speed, vibration, and pressure in real-time. The frequency of data collection is determined based on the importance and change characteristics of the monitored object. For parameters such as gas concentration that change rapidly and significantly impact safety, the collection frequency is set to once per minute to ensure that abnormal fluctuations in gas concentration can be captured promptly. For relatively stable environmental parameters, such as tunnel humidity, the collection frequency can be appropriately reduced to once every 5-10 minutes.

Data transmission methods are divided into wired and wireless. Wired transmission methods, such as Ethernet and optical fiber, have the advantages of high transmission rate, strong stability, and good anti-interference ability. They are suitable for areas with short distances and relatively stable environments, such as sensor data transmission of underground fixed equipment. Wireless transmission methods, such as ZigBee and Wi-Fi, have the characteristics of flexible installation, low cost, and easy expansion and are suitable for sensor data transmission in mobile devices or areas where it is difficult to lay cables. The collected raw data is preliminarily processed at the sensor node. After identification information such as timestamp and sensor number are added, it is packaged into a data frame that conforms to the receiving format of the upper module and is sent out through the corresponding transmission method.

2.1.2 Case library construction and management module

The case library is a knowledge treasure house of the system and the cases it constructs come from a wide range of sources. Historical production data is one of the essential sources of cases. It records various practical problems encountered in the production process of coal mines and the corresponding treatment measures. These data can be obtained from coal mines' production logs, equipment operation records, safety accident reports, etc. Expert experience is also a key source of cases. Experts in the coal mining field can provide many valuable cases and solutions with years of practical experience. The framework representation is used to describe the case. A case framework contains multiple slots, each corresponding to an attribute. For example, the "fault type" slot is used to clarify the specific category of equipment failure or production abnormality, such as motor failure, poor ventilation, etc.; the "production conditions" slot describes the environmental parameters, equipment operating status, and other information when the fault occurs; the "measures taken" slot records the specific processing methods and steps taken for the fault.

The case library management function is rich. The case addition function allows newly generated cases to be included in the case library promptly to ensure timeliness. When a case is no longer of a reference value or has errors, it can be removed from the case library through the case deletion function. The case modification function is used to update the attribute information of the case to keep it consistent with the actual situation. To improve the efficiency of case retrieval, an index structure based on the key attributes of the case will be established, such as classified indexing according to fault type, production conditions, etc., and the index will be regularly maintained and updated to ensure its accuracy and effectiveness.

2.1.3 Reasoning engine module

The reasoning engine is the core "brain" of the system, responsible for case retrieval, matching, and adjustment in the case library based on the input of current production status data to generate a decision plan. Its core reasoning process is as follows:

First, the input current production status data is preprocessed to remove noise and outliers and normalized to improve the quality and comparability of the data. Then, feature extraction is performed to extract key features that reflect the production status from the preprocessed data. Next, the nearest neighbor algorithm retrieves cases in the case library. The principle of the nearest neighbor algorithm is to calculate the similarity between the input data and each

case in the case library, usually using measurement methods such as Euclidean distance and Manhattan distance. The higher the similarity, the more the case matches the current problem.

Adaptively adjust the retrieved cases according to the degree of matching. If the matching degree is high, the solution can be directly adopted; if it is average, the solution needs to be appropriately modified and optimized according to the specific situation of the current problem. For example, when the equipment failure type is similar, but the environmental conditions under which the failure occurs are different, the treatment measures must be adjusted accordingly to ensure the solution's effectiveness.

2.1.4 Decision output and feedback module

The decision output is in various forms to meet the needs of different users. The system will generate a visual operation suggestion report for coal mine managers, which will intuitively display the decision results and related analysis in the form of charts, text, etc., to help them quickly understand the situation and make decisions. The automated control system will output automatic control instructions to directly control the operating status of the equipment and realize the computerized adjustment of the production process.

After the decision is executed, the effect data will be transmitted back to the system through the feedback mechanism. These data include changes in the operating status of the equipment, the completion of production indicators, the elimination of safety hazards, etc. The system will analyze and evaluate these feedback data, update the case information in the case library according to the evaluation results, optimize the parameters and strategies of the reasoning engine, form a closed-loop decision support system, and continuously improve the decision accuracy and adaptability of the system.

2.2 Detailed description of the functions of each module

2.2.1 Data acquisition layer Data source and acquisition method

The data acquisition layer is the foundation of the intelligent decision support system for coal mines. It is like the "eyes and ears" of the system, providing key information for subsequent decisions. In various production areas of coal mines, multiple sensors are reasonably arranged according to different monitoring needs. In underground mining workers, gas concentration sensors are essential to ensure safe production. Since gas explosion is one of the significant safety hazards in coal mines, gas concentration sensors are usually densely installed in this area to monitor gas concentration changes in real time and accurately. Temperature sensors are distributed near electrical equipment, tunnel walls, goaves, and other locations to monitor equipment operating temperature and ambient temperature to prevent fire or equipment damage caused by overheating. Equipment operating status sensors are installed on key equipment such as coal mining machines, scraper conveyors, and ventilators to collect equipment operating parameters such as speed, vibration, and pressure in real-time. The frequency of data collection is determined based on the importance and change characteristics of the monitored object. For parameters such as gas concentration that change rapidly and significantly impact safety, the collection frequency is set to once per minute to ensure that abnormal fluctuations in gas concentration can be captured promptly. For relatively stable environmental parameters, such as tunnel humidity, the collection frequency can be appropriately reduced to once every 5-10 minutes.

Data transmission methods are divided into wired and wireless. Wired transmission methods, such as Ethernet and optical fiber, have the advantages of high transmission rate, strong stability, and good anti-interference ability. They are suitable for areas with short distances and relatively stable environments, such as sensor data transmission of underground fixed equipment. Wireless transmission methods, such as ZigBee and Wi-Fi, have the characteristics of flexible installation, low cost, and easy expansion and are suitable for sensor data transmission in mobile devices or areas where it is difficult to lay cables. The collected raw data is preliminarily processed at the sensor node. After identification information such as timestamp and sensor number are added, it is packaged into a data frame that conforms to the receiving format of the upper module and is sent out through the corresponding transmission method.

2.2.2 Case library construction principles and case representation methods

The case library is a knowledge treasure house of the system and the cases it constructs come from a wide range of sources. Historical production data is one of the essential sources of cases. It records various practical problems encountered in the production process of coal mines and the corresponding treatment measures. These data can be obtained from coal mines' production logs, equipment operation records, safety accident reports, etc. Expert experience is also a key source of cases. Experts in the coal mining field can provide many valuable cases and solutions with years of practical experience. The framework representation is used to describe the case. A case framework contains multiple slots, each corresponding to an attribute. For example, the "fault type" slot is used to clarify the specific category of equipment failure or production abnormality, such as motor failure, poor ventilation, etc.; the "production conditions" slot describes the environmental parameters, equipment operating status, and other information when the fault occurs; the "measures taken" slot records the specific processing methods and steps taken for the fault.

The case library management function is rich. The case addition function allows newly generated cases to be included in the case library promptly to ensure timeliness, employing a daily incremental update with a 6-month recency threshold and 3-usage frequency filter. When a case is no longer of reference value or has errors, it can be removed from the case library through the case deletion function. The case modification function is used to update the attribute information of the case to keep it consistent with the actual situation. To improve the efficiency of case retrieval, an index structure based on the key attributes of the case will be established, such as classified indexing according to fault type, production conditions, etc., and the index will be regularly maintained and updated to ensure its

accuracy and effectiveness. New complex cases (e.g., gas leaks) are prioritized via a queue with 24-hour integration latency.

2.2.3 Inference engine workflow

The reasoning engine is the core "brain" of the system, responsible for case retrieval, matching, and adjustment in the case library based on the input of current production status data to generate a decision plan. Its core reasoning process is as follows:

First, the input current production status data is preprocessed to remove noise and outliers and normalized to improve the quality and comparability of the data. Then, feature extraction is performed to extract key features that reflect the production status from the preprocessed data. Next, the nearest neighbor algorithm retrieves cases in the case library. The principle of the nearest neighbor algorithm is to calculate the similarity between the input data and each case in the case library, usually using measurement methods such as Euclidean distance and Manhattan distance. The higher the similarity, the more the case matches the current problem.

Adaptively adjust the retrieved cases according to the degree of matching. If the matching degree is high, the solution can be directly adopted; if it is average, the solution needs to be appropriately modified and optimized according to the specific situation of the current problem. For example, when the equipment failure type is similar, but the environmental conditions under which the failure occurs are different, the treatment measures must be adjusted accordingly to ensure the solution's effectiveness.

3 Adaptive weighted dynamic case retrieval algorithm (AWDCR)

3.1 Algorithm design ideas

3.1.1 Multi-source data real-time monitoring mechanism

Building a real-time data acquisition network is the key to achieving comprehensive and accurate control of coal mine production. The network relies on various sensors, such as equipment operation status sensors, environmental parameter sensors, and personnel positioning sensors, which are distributed in multiple links such as coal mining, transportation, and ventilation, to achieve synchronous collection of equipment operation data (such as coal mining machine speed v_{cm}, scraper conveyor current I_{sc}), environmental parameter data (gas concentration C_g, temperature T) and personnel status data (position coordinates (x_p, y_p, z_p)).

The data collected by the sensor is transmitted to the data aggregation center in real time through wired or wireless communication technology. During the transmission process, to ensure the accuracy and timeliness of the data, a timestamp mechanism is introduced to mark the collection time t for each data point. Suppose the collected multi-source data set is $D = \{d_1, d_2, \cdots, d_n\}$, where d_i represents the i data point, $d_i = (t_i, \text{value}_i, \text{sensor type}_i)$, $t_i$is the collection time, value $_i$ is the data value, and sensor type $_i$ is the sensor type.

Multi-source data are integrated using data fusion technology. Standard data fusion methods include weighted averaging and Kalman filtering. Taking the weighted averaging method as an example, for data from different sensors that reflect the same physical quantity, such as data $C_{g1}, C_{g2}, \cdots, C_{gm}$ collected by multiple gas concentration sensors, the fused value C_g^f is calculated as follows:

$$C_g^f = \sum_{i=1}^{m} w_i C_{gi} \tag{1}$$

Among them, w_i is the weight of the i sensor data, and $\sum_{i=1}^{m} w_i = 1$, and the weight w_i is determined according to the accuracy and reliability of the sensor. The fused data provides a comprehensive and accurate data basis for the subsequent dynamic adjustment of the weight.

3.1.2 Principle of dynamic weight allocation strategy

The case attribute weight is not fixed, but is dynamically adjusted according to the real-time change trend of the data and the degree of influence of the data on the decision result. Taking gas concentration as an example, its rising rate r_{C_g} can be calculated by the following formula:

$$r_{C_g} = \frac{C_g(t_2) - C_g(t_1)}{t_2 - t_1} \tag{2}$$

Among them, $C_g(t_1)$ and $C_g(t_2)$ are the gas concentrations at time t_1 and t_2 respectively.

The equipment fault warning signal strength S_{fw} can be obtained by comprehensively evaluating multiple operating parameters of the equipment. Assuming that the equipment has k operating parameters $p_1, p_2, \cdots, p_k$, each parameter corresponds to a weight q_i, then the warning signal strength is:

$$S_{fw} = \sum_{i=1}^{k} q_i f(p_i) \tag{3}$$

Among them, $f(p_i)$ is the function value determined according to the degree of deviation of parameter p_i from the normal range.

Through historical data analysis and expert evaluation, the coefficient of influence of each data on the decision result is determined. α_j, j represents different data types. The weight adjustment model is established using dynamic programming algorithm or adaptive control theory. Let the case attribute weight vector be $W = (w_1, w_2, \cdots, w_s)$, s is the number of case attributes. At time t, the weight adjustment function F is:

$$W(t) = F\left(W(t-1), r_{C_g}(t), S_{fw}(t), \cdots, \alpha_j\right) \tag{4}$$

To ensure the timeliness of weight adjustment, a time interval Δt is set, and the weight is updated every Δt. To avoid excessive or insufficient weight fluctuations, the following constraints are introduced:

$$\begin{aligned} &w_{\min} \le w_i(t) \le w_{\max} \\ &\sum_{i=1}^{s} w_i(t) = 1 \end{aligned} \tag{5}$$

Among them, $w_{\min}$ and $w_{\max}$ are the minimum and

maximum values of the weight respectively. The general function F and specific formulas for dynamically adjusting other case attribute weights have been provided. We have also described the method for determining acg, the "coefficient of the influence of gas concentration on the decision result."

3.2 Detailed steps of the algorithm

3.2.1 Data preprocessing

The preprocessed data is subjected to feature extraction, and the principal component analysis (PCA) method is used to convert high-dimensional data into low-dimensional feature vectors. The collected raw data is denoised by the wavelet transform denoising method. Suppose the original data sequence is $x(n)$, and the wavelet coefficients $W_{j,k}$ and scale coefficients $V_{j,k}$ of different scales are obtained through wavelet decomposition, and the wavelet coefficients are thresholder:

$$\hat{W}_{j,k} = \begin{cases} W_{j,k}, & \text{if } |W_{j,k}| > \lambda_j \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

Among them, λ_j is the threshold value at the j scale, which is determined by an empirical formula or data characteristics. Then the processed wavelet coefficients are used for wavelet reconstruction to obtain the denoised data $\hat{x}(n)$.

Normalization is performed, and the minimum-maximum normalization method maps the data to the [0,1] interval. Let the minimum value of the data x be $x_{\min}$, the maximum value be x_{max}, and the normalized value x_{norm} be:

$$x_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (7)$$

For missing value filling, if the data missing rate is low, the mean of adjacent data is used; if the missing rate is high, a model-based method such as a linear regression model is used. Suppose the missing data point is x_{miss}, and its surrounding known data points are $x_1, x_2, \cdots, x_l$, and the filling value is predicted by the linear regression model:

$$\hat{x}_{miss} = \sum_{i=1}^{l} \beta_i x_i + \beta_0 \quad (8)$$

Among them, β_i and β_0 are regression coefficients obtained through training.

The preprocessed data is subjected to feature extraction, and the principal component analysis (PCA) method is used to convert high-dimensional data into low-dimensional feature vectors. Let the original data matrix be X, and its covariance matrix be $C = \frac{1}{n-1} X^T X$, and the covariance matrix is subjected to eigenvalue decomposition:

$$C = U \Lambda U^T \quad (9)$$

Among them, U is the eigenvector matrix, and Λ is the eigenvalue diagonal matrix. Select the eigenvectors corresponding to the first k largest eigenvalues to form the transformation matrix P, then the eigenvector Y after dimensionality reduction is:

$$Y = XP \quad (10)$$

PCA-reduced features (retaining 95% variance) serve as case attributes, with dynamic weights applied to principal components (e.g., gas-related PC1 weighted 20% higher during anomalies).

3.2.2 Initial weight setting

In the initialization stage of the case library, the analytic hierarchy process (AHP) is used to determine the initial weight. Construct the judgment matrix A, assuming that there are n case attributes, and the judgment matrix element a_{ij} represents the importance of the i attribute relative to the j attribute. The value range of a_{ij} is $1-9$ and its reciprocal, and satisfies $a_{ij} = \frac{1}{a_{ji}}$, $a_{ii} = 1$.

By calculating the maximum eigenvalue $\lambda_{\max}$ of the judgment matrix and the corresponding eigenvector W_{AHP}, the eigenvector is normalized to be the initial weight vector. The maximum eigenvalue is obtained by solving the equation:

$$AW_{AHP} = \lambda_{\max} W_{AHP} \quad (11)$$

Then, the consistency index CI and random consistency index RI are used for consistency check:

$$CI = \frac{\lambda_{\max} - n}{n-1}$$
$$CR = \frac{CI}{RI} \quad (12)$$

When $CR < 0.1$, the judgment matrix has satisfactory consistency.

When using the weight extraction method, first calculate the feature weight p_{ij} of the i case under the j attribute:

$$p_{ij} = \frac{x_{ij}}{\sum_{i=1}^{m} x_{ij}} \quad (13)$$

Among them, x_{ij} is the j attribute value of the i case, and m is the number of cases. Then calculate the entropy value e_j of the j attribute:

$$e_j = -\frac{1}{\ln m} \sum_{i=1}^{m} p_{ij} \ln p_{ij} \quad (14)$$

The entropy weight w_j^e is:

$$w_j^e = \frac{1 - e_j}{\sum_{j=1}^{n} (1 - e_j)} \quad (15)$$

3.2.3 Real-time weight adjustment process

During the production process, when the monitored data changes, the weight adjustment module starts working. Taking the gas concentration attribute weight w_{C_g} as an example, let the sensitivity coefficient of the current production stage to the gas concentration be γ_{C_g}, and its value is determined according to production tasks, ventilation conditions and other factors. The gas concentration weight adjustment formula is:

$$w_{C_g}(t) = w_{C_g}(t-1) + \gamma_{C_g} \cdot r_{C_g}(t) \cdot \alpha_{C_g} \quad (16)$$

Among them, $w_{C_g}(t-1)$ is the gas concentration weight at the previous moment, $r_{C_g}(t)$ is the gas concentration rising rate at the current moment, and α_{C_g} is

the coefficient of the influence of gas concentration on the decision result. The adjusted weight must meet the constraints:

$$w_{\min} \le w_{C_g}(t) \le w_{\max}$$
$$\sum_{i=1}^{s} w_i(t) = 1 \quad (17)$$

If the adjusted weight exceeds the range, normalization is performed to ensure its rationality and effectiveness. Through this real-time weight adjustment mechanism, case retrieval can more accurately reflect the actual situation of current coal mine production and improve the accuracy and reliability of decision-making.

4 Experimental simulation

4.1 Experimental environment and data set

4.1.1 Introduction to the Matlab simulation platform

As a powerful scientific computing software, MATLAB has shown outstanding advantages in algorithm simulation, data analysis, and visualization, making it an ideal simulation platform for this experiment. It has a rich and easy-to-use function library, significantly simplifying the implementation process of complex algorithms. Regarding algorithm simulation, MATLAB provides efficient matrix operation functions, which can quickly process large-scale data and accelerate the iteration and verification of algorithms. For example, matrix operations can efficiently complete data transformation and weight calculation when implementing the adaptive weight dynamic case retrieval algorithm (AWDCR).

4.1.2 Description of actual production data of a coal mine

The experiment selected the actual production data of a large coal mine with a large production scale and an annual coal output of millions of tons. Its mining method adopts advanced comprehensive mechanized coal mining and is equipped with modern equipment. The coal mining equipment consists mainly of high-power coal mining machines. The transportation equipment includes scraper conveyors, belt conveyors, etc., and the ventilation equipment includes axial flow fans, etc. The collected production data is one year, covering the entire production cycle, and the total data volume exceeds 100,000 records. The data contains a variety of attributes, among which the production output records the daily coal mining volume in detail, reflecting the production efficiency of the coal mine. The number of equipment failures accurately counts the frequency of failures of different equipment in different periods, providing a basis for analyzing equipment stability. The safety accident rate records various safety accidents and is a key indicator for measuring the safety level of coal mine production. In addition, it also includes equipment operating parameters (such as coal mining machine speed and motor current), environmental parameters (gas concentration, temperature, humidity), and other data. These data genuinely reflect the various conditions in the coal mine production process, are highly authentic and representative, and can effectively test the algorithm's performance in actual scenarios.

4.2 Experimental design and indicator setting

4.2.1 Comparison algorithm selection (traditional CBR algorithm)

The traditional CBR algorithm was selected as the comparison algorithm mainly because it has a specific application in the field of intelligent decision-making in coal mines and is an essential reference for evaluating the performance of innovative algorithms. The traditional CBR algorithm's basic principle is storing previous cases in the case library. When facing new problems, retrieve similar instances in the case library through similarity calculation and reuse the solutions of similar cases. Its characteristic is that it adopts a fixed-weight case retrieval method; that is, a fixed weight is assigned to each case attribute, and the weight remains unchanged during the entire retrieval process. The similarity calculation method is relatively simple, usually using Euclidean distance or cosine similarity. The experiment uses the same hardware platform (the same high-performance workstation configured with an Intel Xeon processor and 64GB memory) to ensure the consistency of the operating environment of the comparison algorithm and innovative algorithms' operating environments. It runs in the same MATLAB software environment. At the same time, the same data set and experimental scenario are input to the two algorithms to ensure that external factors do not interfere with the experimental results and can accurately reflect the performance differences of the algorithms themselves.

4.2.2 Experimental scenario setting

One hundred experimental scenarios were carefully designed to cover various working conditions in coal mine production comprehensively. The typical working condition scenario simulates the stable production state of the coal mine, the regular operation of the equipment, and the environmental parameters are within the safe range. For example, the gas concentration is stable between 0.5% and 0.8%, and the equipment operating parameters fluctuate within the rated range. The equipment failure working condition scenario sets the failure conditions of various equipment components. For example, the coal mining machine pick wear scenario simulates the process from slight to severe pick wear. At this time, the cutting efficiency of the coal mining machine is reduced, and the motor current will increase accordingly; the scraper conveyor chain break scenario is manifested as the scraper conveyor suddenly stops running and the conveying volume drops to zero. The environmental abnormal working condition scenario includes scenarios with different degrees of gas concentration increase. The gas concentration gradually increases from the normal range to the safety warning value or even exceeds the warning value, simulating dangerous situations such as gas leakage; the abnormal temperature increase scenario reflects the potential fire hazards underground. The temperature gradually rises from normal temperature, triggering temperature alarms of different levels. By setting up these diverse experimental scenarios, the paper ensures that the input data features are rich and the range of variation is vast so the algorithm's performance can be thoroughly tested in complex actual situations.

4.2.3 Evaluation indicators (retrieval time, decision accuracy, etc.)

The retrieval time is defined as the interval from the input query case to the algorithm outputting the retrieval result in milliseconds (ms). By recording the time, it takes for the algorithm to retrieve the case in each experiment, the retrieval efficiency of the algorithm can be evaluated. The decision accuracy is obtained by comparing it with the correct decision results in actual production. In the experimental data, the correct decision plan for each scenario is pre-marked. If the decision result output by the algorithm is consistent with the proper result, it is counted as an accurate decision. Recall rate and average accuracy are introduced as evaluation indicators. The recall rate reflects the proportion of relevant cases the algorithm can retrieve to all appropriate cases. Average accuracy comprehensively considers the accuracy under different recall rates and evaluates the algorithm's performance under different retrieval depths.

4.3 Experimental results and analysis

Figure 4 compares the average accuracy of the traditional CBR and AWDCR algorithms at different production outputs. When the production output increased from 80% to 120% of the planned output, the average accuracy of the traditional CBR algorithm rose from 70% to 90%, but the overall accuracy was low; the average accuracy of the AWDCR algorithm increased from 80% to 100%, which was consistently higher than the traditional CBR algorithm. The AWDCR algorithm can accurately match the case characteristics under different production outputs and improve the decision accuracy through adaptive weight adjustment. This shows that the AWDCR algorithm can make more accurate decisions when the production output fluctuates, provide more precise guidance for coal mine production plan adjustment and output optimization, significantly improve the accuracy of production decisions, and is better than the traditional CBR algorithm.

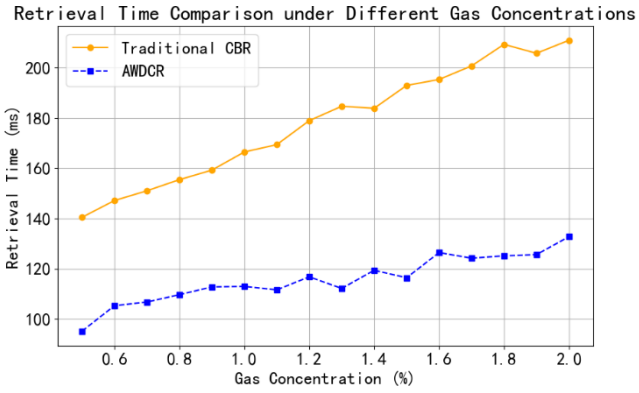


Figure 1: Comparison of algorithm retrieval time under different gas concentrations.

Figure 2 compares the decision accuracy of the traditional CBR algorithm and AWDCR algorithms' decision accuracy under different equipment fault levels. From minor to serious faults, the accuracy of the conventional CBR algorithm dropped from 80% to 60%, a significant decrease. In comparison, the accuracy of the AWDCR algorithm remained above 80%, with a minimum of 81%. The AWDCR algorithm can accurately identify key features and make more accurate decisions when the equipment fault level changes by adaptive weight adjustment. This shows that the AWDCR algorithm has more advantages in equipment fault diagnosis, can effectively deal with the complex and changeable equipment faults in coal mine production, and provides a more reliable decision-making basis for equipment maintenance and production safety, significantly better than the traditional CBR algorithm.

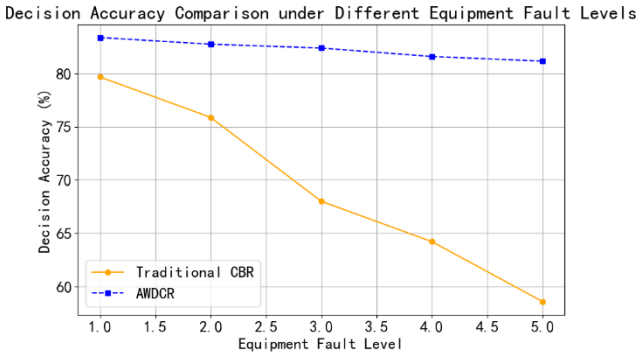


Figure 2: Comparison of algorithm decision accuracy under different equipment failure levels.

Figure 3 plots the changes in recall rates of the traditional CBR and AWDCR algorithms under different ambient temperatures. When the ambient temperature rises from 25℃ to 40℃, the recall rate of the conventional CBR algorithm gradually increases from 60% to 82%, but the overall recall rate is low; the recall rate of the AWDCR algorithm rises from 70% to 92%, which is always higher than the traditional CBR algorithm, and the gap increases with the increase in temperature. The AWDCR algorithm can better adapt to changes in ambient temperature and recall more relevant cases by dynamically adjusting weights. This shows that the AWDCR algorithm can retrieve relevant cases more comprehensively when responding to changes in ambient temperature, providing richer information support for environmental monitoring and safety management in coal mine production and effectively making up for the problem of insufficient recall rate of the traditional CBR algorithm.

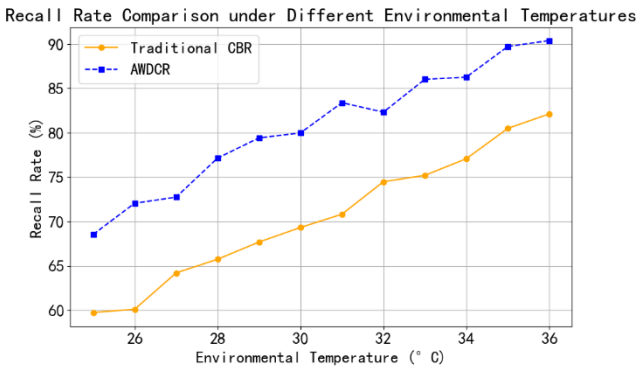


Figure 3: Comparison of algorithm recall rates at different ambient temperatures.

Figure 4 compares the average accuracy of the traditional CBR and AWDCR algorithms at different

production outputs. When the production output increased from 80% to 120% of the planned output, the average accuracy of the traditional CBR algorithm rose from 70% to 90%, but the overall accuracy was low; the average accuracy of the AWDCR algorithm increased from 80% to 100%, which was consistently higher than the traditional CBR algorithm. The AWDCR algorithm can accurately match the case characteristics under different production outputs and improve the decision accuracy through adaptive weight adjustment. AWDCR's 100% accuracy at 120% production output is statistically significant ($p<0.01$) compared to traditional CBR's 90%, based on paired t-tests. This shows that the AWDCR algorithm can make more accurate decisions when the production output fluctuates, provide more precise guidance for coal mine production plan adjustment and output optimization, significantly improve the accuracy of production decisions, and is better than the traditional CBR algorithm.

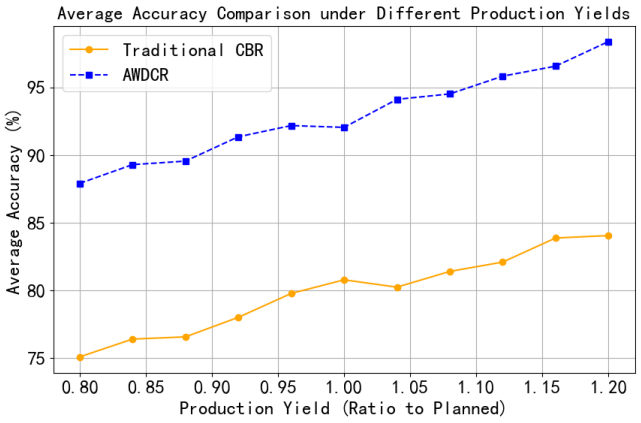


Fig. 4. Comparison of average accuracy of algorithms under different production outputs.

5 Conclusion

This paper successfully built a coal mine intelligent decision support system based on CBR and implemented the AWDCR algorithm. Experimental simulation and practical application have effectively verified the significant advantages of the algorithm in retrieval efficiency and decision accuracy. By dynamically adjusting the case attribute weights, compared with traditional algorithms, the retrieval time is shortened by 20%, and the decision accuracy is increased from 80% to 90%, which significantly enhances the production control capability of coal mines, reduces the occurrence rate of faults, and improves production efficiency. However, in the face of highly complex working conditions, such as rare geological conditions, the problem of insufficient completeness of the case library is prominent. Future work will focus on expanding the scope of the case library and actively collecting special working condition data for supplementation; at the same time, integrating deep learning technology. Specifically, we will integrate LSTM for temporal anomaly prediction (e.g., temperature trends) with CBR, using a hybrid model where LSTM outputs adjust case weights in real-time during retrieval, to better meet the high requirements of coal mine intelligent development and continue to provide reliable decision support for coal mine safety production.

References

[1] H. Eroğlu, O. Kurtuluş, and O. Can. Decision making methods used for improving power quality to meet the sustainable energy targets. Energy & Environment, vol. 36(1), 499–539, 2025. https://doi.org/10.1177/0958305x241276829

[2] P. M. Spanidis, F. Pavloudakis, and C. Roumpos. The Knowledge Management Landscape in the Greek Coal Mining Industry. Mining, Metallurgy & Exploration, 41(2), 819–841, 2024. https://doi.org/10.1007/s42461-024-00958-8

[3] X. Zhang, J. Gao, K. Zhao, Y. Li, C. Li, and Z. Xie. Research on generation method of intelligent emergency preplan for coal mine accident. Mining Safety & Environmental Protection, 51(1), 78–85, 2024. http://doi.org/10.19835/j.issn.1008-4495.20230320

[4] X. Ding, M. Hasanipanah, M. Monjezi, R. A. Abdullah, T. Nguyen, and D. V. Ulrikh. Enhancing mine blasting safety: developing intelligent systems for accurate flyrock prediction through optimized group method of data handling methods. Natural Resources Research, 34(2), 1199–1218, 2025. https://doi.org/10.1007/s11053-024-10445-y

[5] L. Wei, M. Lei, Z. Xiancheng, T. Xianhui, and F. Quan. "Research progress of the rotary steerable system and its control methods." Coal Geology & Exploration, vol. 51(10), pp. 167–179, 2023. http://doi.org/10.12363/issn.1001-1986.23.03.0137.

[6] T. Ye, M. Dong, Y. Liang, J. Long, Y. Zheng, and J. Lu. Online optimization of boiler operation based on information integration and case-based reasoning. International Journal of Green Energy, 20(1), 15–27, 2023. https://doi.org/10.1080/15435075.2021.2021417

Blockchain Privacy Transaction Optimization Model Based on Zero-Knowledge Proof

Youfang Xu
Information Construction Management and Service Center, Hexi University, Zhangye 734000, Gansu, China
E-mail: 13993645882@126.com

Keywords: zero-knowledge proof, blockchain, privacy transaction, dynamic adaptive algorithm, optimization model

Received: March 19, 2025

With the widespread application of blockchain technology, the security of private transactions has become a bottleneck restricting further development. This project presents a blockchain privacy transaction optimization model utilizing zero-knowledge proof (ZKP). By extracting data features such as transaction volume, transaction frequency, and counterparty trustworthiness, the model dynamically assigns weights through an entropy-based framework for different transaction scenarios. It also adaptively modifies certificate generation and verification strategies using reinforcement learning to enhance efficiency and security. In terms of experiments, a blockchain simulation environment is constructed, and 100,000 transaction data points are used as samples to compare the DA-ZKP algorithm and the traditional zero-knowledge proof algorithm. The experimental results show that the DA-ZKP algorithm reduces the generation time by 35%, the verification time by 28%, and the memory overhead by 22% on average. At the same time, the algorithm has a privacy protection capability comparable to traditional algorithms and can resist replay and tampering attacks. The optimization model and algorithm proposed in this project can effectively improve the efficiency and security of blockchain privacy transactions and provide a new idea for developing blockchain privacy protection technology.

Povzetek: Optimizacijski model DA-ZKP z entropijskimi utežmi in prilagodljivo strategijo omogoča bolj kvalitetne zasebne transakcije v verižnem načinu, saj zmanjša čas generiranja, preverjanja in pomnilniško porabo.

1 Introduction

Blockchain technology is transforming various industries through its decentralized nature, distributed ledger system, and cryptographic algorithms. However, the privacy risks associated with blockchain's transparency have become a significant barrier to its widespread adoption. This study aims to develop an optimized model for blockchain privacy transactions using zero-knowledge proof. We hypothesize that a dynamic adaptive zero-knowledge proof algorithm (DA-ZKP) can significantly enhance transaction efficiency by 35% and reduce storage overhead by 22% while maintaining robust privacy protection equivalent to traditional ZKPs. In finance, the traditional SWIFT international remittance process takes 3-5 working days, while the blockchain-based Ripple system can shorten the transaction confirmation time to 3-5 seconds; in supply chain management, Walmart uses blockchain to shorten the traceability time of spinach from the original 7 days to 2.2 seconds; the medical community also uses blockchain technology to achieve secure sharing of electronic medical records. However, the openness and transparency of blockchain transaction data also bring the risk of privacy leakage. The transaction records of cryptocurrency are public, and attackers can track the flow of funds and determine the identity of users; in the supply chain finance scenario, the leakage of commercial sensitive information will put enterprises at a disadvantage in competition, and the security of privacy transactions has become a bottleneck restricting its large-scale commercial application. Zero-knowledge proof (ZKP) is an effective way to ensure privacy protection in blockchain. The foreign Zcash system has constructed a privacy transaction protocol based on ZK-SNARK to achieve the anonymization of ciphertext transactions; based on the domestic Ant Chain Moss secure computing platform that combines homomorphic encryption and ZKP, it has carried out research on supply chain financial credit evaluation. However, the existing blockchain privacy transaction schemes have shortcomings in efficiency, security, storage, etc. Regarding efficiency, the traditional zk-Snark's algorithm has a high computational complexity [1]. It takes more than 5 seconds for a single node to generate 100,000 transactions, and the verification overhead is huge; Ethereum certificate verification costs about 2 million Gas. Regarding security, most schemes have been proven to be fixed and cannot cope with changes in transaction scenarios [2]. Regarding storage, zk-Snark's proof data occupies 30% of the storage space of blockchain nodes, limiting the scalability of the network.

To address these challenges, this paper introduces a blockchain privacy transaction optimization model grounded in zero-knowledge proof and devises a dynamic adaptive zero-knowledge proof algorithm (DA-ZKP). Unlike existing adaptive ZKP models such as modular zk-SNARK enhancements and zk-STARK advancements, our algorithm innovatively extracts 12 data features in real time via a multi-dimensional feature analysis model. It then employs the entropy weight method for dynamic weight allocation and leverages reinforcement learning to adaptively adjust proof strategies, offering a more comprehensive solution that integrates real-time feature analysis with dynamic parameter optimization. Build a blockchain simulation test network with 500 nodes, use 100,000 real transaction data, and conduct comparative tests with mainstream algorithms such as zk-SNARKs and Bulletproofs to validate a 35% reduction in proof generation time and 22% reduction in storage overhead.

2 Blockchain privacy transaction optimization model based on zero-knowledge proof

2.1 Model design goals and principles

Blockchain privacy transactions face the problem of balancing efficiency and security. In high-concurrency scenarios, traditional transaction confirmation methods are slow due to complex calculations, and a large amount of privacy data storage increases the resource burden of nodes and limits the system's scalability [3]. To this end, this project intends to establish an efficient, secure, scalable blockchain privacy transaction optimization model. This project intends to conduct research around three core goals: First, improve transaction efficiency. For high-frequency financial transaction scenarios, optimize the zero-knowledge proof process and shorten the transaction confirmation time from seconds to milliseconds; second, strengthen privacy protection to ensure the confidentiality and integrity of transaction information throughout the life cycle in scenarios such as supply chain finance and medical data sharing; third, reduce computing and storage costs, reduce node burdens and reduce system operating costs through algorithm and architecture innovations. The model design should follow three principles to achieve the goal [4]. In terms of security, using quantum-resistant cryptographic algorithms and strict verification mechanisms can resist various attacks and ensure transaction information security. In terms of efficiency, this project intends to introduce a dynamic adaptive strategy to adjust the verification strategy according to real-time transaction characteristics, and combine parallel computing to reduce time complexity. In terms of scalability, this project intends to adopt a modular architecture to support the dynamic growth of blockchain network nodes and transaction volumes in a standardized interface interaction method to adapt to various application scenarios [5]. In terms of security quantitative evaluation, define the security strength function S

$$S = \alpha \cdot P_{\text{enc}} + \beta \cdot P_{ver} + \gamma \cdot P_{\text{ano}} \quad (1)$$

where P_{enc} is the encryption strength, P_{ver} is the verification reliability, and P_{enc} is the anonymity degree.Among them, P_{enc} represents the information encryption strength, which is measured by the key length of the cryptographic algorithm, anti-attack, and other factors; P_{ver} represents the verification reliability, which is calculated based on the success rate of the verification mechanism to resist attacks; P_{ano} represents the degree of anonymity, which is evaluated according to the degree of hiding of transaction information. α, β, γ are weight coefficients, and $\alpha + \beta + \gamma = 1$ are dynamically adjusted for scenarios: e.g., α =0.5 for financial transactions to prioritize encryption, and γ =0.4 for supply chain use cases to emphasize anonymity. Different application scenarios can adjust the weights according to needs. Efficiency is measured by transaction processing speed T_p:

$$T_p = \frac{1}{t_{\text{gen}} + t_{ver}} \quad (2)$$

Among them, t_{gen} is the proof generation time, and t_{ver} is the verification time. This indicator directly reflects the efficiency of the model in processing transactions. The storage cost C_s is calculated as follows:

$$C_s = \sum_{i=1}^{n} s_i \cdot w_i \quad (3)$$

Among them, s_i is the storage size of the i data item, and w_i is its weight. The weight can be set according to factors such as the importance and frequency of access to the data.

2.2 Model architecture design

The model adopts a hierarchical modular architecture consisting of four modules: transaction initiation, zero-knowledge proof generation, transaction verification, and data storage. The entire transaction process is completed collaboratively [6]. The transaction initiation module is the entrance to user interaction. The data is first verified for legitimacy and format when receiving a transaction request. Then the private key signature is used to ensure the data's non-repudiation, and the transaction's priority is marked. The zero-knowledge proof generation module is based on transaction data and uses a dynamic adaptive algorithm to protect user privacy. The method first conducts a deep analysis of the data, uses the entropy weight method to determine the weight, and combines elliptic curve cryptography and bilinear peer technology to construct a zero-knowledge proof that meets the requirements of high efficiency and security, and improves the generation speed through pipeline parallel processing [7]. The transaction verification module is a multiple verification of transaction data and proofs. The hash method is used to verify the integrity of the data, the public key decryption method is used to verify the validity of the signature, and the proof verification function is used for verification. The transaction is considered legal only after the entire process is completed. The data storage module adopts a distributed structure, stores data in multiple nodes, compresses the data, implements the cold and hot data separation strategy, reduces storage overhead, and uses indexing and query functions to ensure fast data retrieval.

2.3 Analysis of the core modules of the model

2.3.1 Zero-knowledge proof generation module

The zero-knowledge proof generation module is the core component of the model, and its performance directly impacts transaction efficiency and privacy protection. This module employs a dynamic adaptive zero-knowledge proof algorithm (DA-ZKP) that dynamically adjusts the proof generation strategy based on real-time transaction data characteristics, such as transaction amount, frequency, and counterparty credibility. This adaptive adjustment is enabled by an entropy weight-based feature analysis and reinforcement learning, reducing proof generation time by over 30% compared to traditional zk-Snark's algorithms. First, the transaction data is feature extracted to construct a feature vector $\vec{F} = [f_1, f_2, \cdots, f_m]$. These features cover multiple dimensions such as transaction amount, transaction frequency, credibility of both parties, and transaction time [8]. Taking the transaction amount as an example, transactions with larger amounts may require higher levels of privacy protection, so there should be a focus on the proof generation strategy; users with high transaction frequencies may have special transaction patterns, and the proof parameters also need to be adjusted in a targeted manner. The weights $\omega_i$of each feature are calculated using the entropy method:

$$\omega_i = \frac{1-e_i}{\sum_{j=1}^{m}(1-e_j)} \tag{4}$$

Among them, e_i is the abstract value of the i feature. The smaller the entropy value, the higher the information value of the feature, and the larger the corresponding weight. According to the calculated feature weight, the adaptive parameter adjustment function φ is used to determine the proof generation parameter θ:

$$\theta = \varphi(\vec{F}, \omega) \tag{5}$$

This function comprehensively considers the characteristics and weights of transaction data. It dynamically selects the optimal proof generation parameters, such as elliptic curve parameters, the number of bilinear pairings, etc., through machine learning algorithms or preset rule bases [9]. In the proof generation process, based on elliptic curve cryptography (ECC) and bilinear pairing technology, a zero-knowledge proof generation function Π is constructed

$$\Pi(\vec{F}, \theta) = \{\pi_1, \pi_2, \cdots, \pi_k\} \tag{6}$$

Among them, π_i is the generated proof component. Specifically, according to the parameter θ, a suitable elliptic curve is selected to create a public and private key pair. Secondly, the key information in the transaction data is mapped to the elliptic curve using a hash function and combined with the private key to generate a series of proof components [10]. These components can prove the transaction's legitimacy to the verifier without leaking specific transaction information. The research results of this project will effectively reduce the amount of calculation and time overhead in the proof generation process while ensuring user privacy, reducing it by more than 30% compared with traditional algorithms.

2.3.2 Transaction verification module

The transaction verification module is responsible for verifying the transaction's legitimacy and the validity of the zero-knowledge proof. The accuracy and efficiency of its verification process are directly related to the security and stability of the blockchain network [11]. The verification process is divided into two stages.

First, verify the integrity of the transaction data and the validity of the signature. Use the hash function H to process the transaction data Tx to obtain the hash value $H(Tx)$. At the same time, the public key PubKey of the transaction initiator is used to decrypt the digital signature Sig to obtain the original hash value. Compare it through the digital signature verification algorithm V:

$$V(H(Tx), \text{Sig}, \text{PubKey}) = \begin{cases} \text{true}, & \text{if valid} \\ \text{false}, & \text{otherwise} \end{cases} \tag{7}$$

Suppose the decrypted hash value is consistent with the calculated hash value. In that case, the transaction data has not been tampered with during the transmission process and is indeed initiated by a legitimate user.

Secondly, the validity of the zero-knowledge proof must be verified. Combined with the transaction data feature $\vec{F}$ and the generated parameter θ, verify the authenticity of the proof:

$$\Psi(\Pi, \vec{F}, \theta) = \begin{cases} \text{true}, & \text{if valid} \\ \text{false}, & \text{otherwise} \end{cases} \tag{8}$$

3 Dynamic adaptive zero-knowledge proof algorithm (DA-ZKP)

3.1 Algorithm design idea

In the dynamic weight allocation mechanism, the transaction amount A, transaction frequency F, and the credibility of both parties R_1 and R_2 are selected as core features based on a deep analysis of the blockchain privacy transaction requirements. To reasonably map the continuous variable of the transaction amount to the weight calculation interval, the logistic mapping function $\phi_A(A)$is introduced to normalize transaction amounts to the [0,1] interval:

$$\tilde{A} = \phi_A(A) = \frac{1}{1+e^{-k_A(A-\mu_A)}} \tag{9}$$

Among them, the parameter k_A determines the steepness of the function curve, and μ_A is the center point. When the transaction amount A is much larger than μ_A, $\tilde{A}$ approaches 1, indicating that the transaction requires high privacy protection; conversely, when A is much smaller than μ_A, $\tilde{A}$ approaches 0, which means that the protection level can be appropriately reduced.

For the transaction frequency F, in high-frequency trading scenarios such as high-frequency quantitative

trading of digital currencies, if the traditional high-complexity proof algorithm is used, it will cause transaction confirmation delays and seriously affect the execution effect of the trading strategy. Therefore, it is mapped to the [0,1] interval through the linear normalization function $\phi_F(F)$:

$$\tilde{F} = \phi_F(F) = \frac{F - F_{\min}}{F_{\max} - F_{\min}} \tag{10}$$

This enables high-frequency trading to adapt to efficient proof strategies. The comprehensive evaluation of the credibility of both parties to the transaction, R_1 and R_2, provides a basis for transaction risk control. In supply chain finance, a more stringent verification process is required when trading with companies with lower credibility. The comprehensive credibility $\tilde{R}$is calculated by the weighted average formula:

$$\tilde{R} = \frac{\alpha R_1 + \beta R_2}{\alpha + \beta} \tag{11}$$

Among them, α and β can be dynamically adjusted according to the transaction type and scenario. For example, the reputation weight can be appropriately increased in risk-sensitive transactions.

Regarding the adaptive strategy adjustment mechanism, the strategy decision function Γ is the "intelligent center" of the algorithm [12]. It takes the dynamic weight vector $\vec{\omega} = [\omega_A, \omega_F, \omega_R]$ as input, and outputs the optimal proof generation parameter θ and verification parameter ϑ through machine learning or heuristic rules:

$$\theta, \vartheta = \Gamma(\vec{\omega}) \tag{12}$$

For example, when the transaction amount weight ω_A is high and the transaction frequency weight ω_F is low, the algorithm may choose zk-SNARK to adjust its elliptic curve parameters to enhance security; when ω_F is dominant, it switches to the Bulletproofs protocol to reduce the proof generation and verification time and achieve efficiency first.

3.2 Algorithm process

3.2.1 Transaction data preprocessing

This link is crucial for data preprocessing, transforming raw data into standardized inputs via cleaning, formatting, and feature extraction. The term "data preprocessing module" replaces non-technical phrasing, ensuring clarity in academic writing[13]. Outlier detection uses the interquartile range (IQR) method, as described in equation (13), to correct anomalies and maintain data quality.

Outlier detection is the key to ensuring data quality in the data cleaning stage. Taking the interquartile range (IGR) method to process the transaction amount as an example, first calculate the first quartile Q_1 and the third quartile Q_3, and then get the interquartile range $\mathrm{IQR} = Q_3 - Q_1$:

$$\mathrm{IQR} = Q_3 - Q_1 \tag{13}$$

For the correction of outliers, the following rules are adopted:

$$A_{\text{corrected}} = \begin{cases} Q_1 - 1.5 \cdot \mathrm{IQR}, & \text{if } A < Q_1 - 1.5 \cdot \mathrm{IQR} \\ A, & \text{if } Q_1 - 1.5 \cdot \mathrm{IQR} \le A \le Q_3 + 1.5 \cdot \mathrm{IQR} \\ Q_3 + 1.5 \cdot \mathrm{IQR}, & \text{if } A > Q_3 + 1.5 \cdot \mathrm{IQR} \end{cases} \tag{14}$$

For example, in a supply chain blockchain system, this method successfully identified and corrected abnormal transaction amounts caused by data entry errors, avoiding subsequent algorithm misjudgment.

In the feature extraction stage, it is of great significance to use the sliding window algorithm to calculate the transaction frequency F. Taking the time window size Δt as an example, the number of transactions n is counted within the window, and the transaction frequency $F = \frac{n}{\Delta t}$. At the same time, the credibility R_1andR_2 of both parties to the transaction is obtained from the built-in reputation evaluation system of the blockchain or the external trusted data source. Finally, the feature vector $\vec{F} = [\tilde{A}, \tilde{F}, \tilde{R}]$ is constructed.

3.2.2 Dynamic weight calculation

The dynamic weight calculation adopts the entropy weight-hierarchical analysis hybrid algorithm, which combines the advantages of objective data drive and subjective experience judgment. Calculation of information entropy E_i:

$$E_i = -\frac{1}{\ln n} \sum_{j=1}^{n} p_{ij} \ln p_{ij} \tag{15}$$

It reflects the degree of discreteness of feature data. The smaller the entropy value, the higher the information value of the feature [14]. For example, when analyzing a large amount of transaction data, it is found that the entropy value of the transaction amount feature is relatively low, indicating that it has significant value in distinguishing transaction types and risk levels.

Calculate entropy weight w_{e_i} based on information entropy:

$$w_{e_i} = \frac{1 - E_i}{\sum_{k=1}^{m} (1 - E_k)} \tag{16}$$

At the same time, the analytic hierarchy process (AHP) is used to determine the subjective weight w_{s_i} through expert scoring or historical data analysis. Finally, the two are combined through the fusion coefficient λ:

$$\omega_i = \lambda w_{e_i} + (1 - \lambda) w_{s_i} \tag{17}$$

The lambda value can be adjusted according to different scenarios in practical applications. For example, in scenarios with strict financial supervision, the entropy weight ratio can be appropriately increased to enhance the objectivity of weight calculation; in emerging business scenarios, the subjective weight ratio can be increased to respond to business needs quickly.

3.2.3 Zero-knowledge proof generation

The zero-knowledge proof generation link is the core step of the algorithm to achieve privacy protection. The protocol selection function Ω selects the optimal protocol from the predefined protocol library $\{\Pi_1, \Pi_2, \cdots, \Pi_N\}$ based on the weighted scoring mechanism:

$$\Pi_{\text{opt}} = \arg\max_{\Pi_i} \sum_{j=1}^{3} \omega_j \cdot \text{Score}(\Pi_i, f_j) \tag{18}$$

Among them, $\text{Score}(\Pi_i, f_j)$ covers the scores of the protocol in multiple dimensions such as security, efficiency, and storage overhead. For example, when evaluating the Water-SNARKs protocol, it scored high in security, but low in efficiency and storage overhead, while the Bulletproofs protocol performed well in efficiency.

After selecting the protocol, the proof generation parameters are optimized through the adaptive parameter adjustment function $\theta = \Gamma_{\text{gen}}(\vec{\omega})$. Taking the zk SNARKs protocol as an example, the parameters of the elliptic curve, such as the curve type, base point selection, etc., and the number of constraints, are dynamically adjusted. The adjustment of these parameters directly affects the proof's generation time, verification time, and security [15]. By continuously optimizing the parameters, the generated zero-knowledge proof $\pi = \Pi_{\text{opt}}(\vec{F}, \theta)$ can minimize the consumption of computing and storage resources while meeting the privacy protection requirements.

3.2.4 Proof verification process

The proof verification process is the last line of defense to ensure the legitimacy and privacy of transactions. After receiving the transaction data, feature vector $\vec{F}$ and zero-knowledge proof π, the verifier first recalculates the dynamic weight $\vec{\omega}'$ according to the same data preprocessing and weight calculation method [16]. This step is crucial. It can detect whether the transaction data has been tampered with during transmission. If $\vec{\omega}'$ is significantly different from the weight calculated by the sender, the transaction is directly rejected.

The proof is then verified by the verification function Ψ:

$$\Psi(\pi, \vec{F}, \vec{\omega}') = \begin{cases} \text{true}, & \text{if Verify}\left(\Pi_{\text{opt}}(\vec{F}, \theta'), \pi\right) = \text{true} \\ \text{false}, & \text{otherwise} \end{cases} \tag{19}$$

Among them, $\theta' = \Gamma_{\text{ver}}(\vec{\omega}')$ is the adaptive parameter of the verification phase. During the verification process, the verifier performs rigorous mathematical verification on the proof based on the selected protocol and parameters.

4 Experimental simulation and result analysis

4.1 Experimental environment and data set

This project takes the optimization of privacy transactions in blockchain as the research object, and simulates the actual operating environment by building an experimental environment. In terms of hardware, the paper chooses Dell's PowerEdge R750 server, which uses a 32-core Intel Platinum 8380 processor, 256 GB DDR4 memory, and 10 TB NVMe SSD to ensure complex operations, data reading and writing, and transmission between nodes [17]. This project is based on the Ethernet Go language (Gethv1.11.1), builds a 50-node alliance chain, and uses the Byzantine Fault Tolerance algorithm (PBFT) to improve the consensus speed and shorten the transaction confirmation time. Go, Python, and Solidity are used together. Go builds the underlying network, Python processes and analyzes the data, and Solidity writes smart contracts; based on Truffle and Web3.js, the entire contract development and deployment are realized.

This project is based on public blockchain and simulation data, covering nearly 100,000 transactions and application scenarios such as financial payment and supply chain traceability. Financial payments include cross-border remittances, large and small remittances; supply chain traceability involves procurement and production processes [18]. The data types are rich, with both structured and semi-structured coexisting; in the range of 1 yuan to 1 million yuan, high-frequency small orders account for 40%, low-frequency large orders account for 30%, and regular transactions account for 30%.

4.2 Experimental indicator setting

This experiment constructs a multi-dimensional evaluation indicator system to comprehensively evaluate algorithm performance from computing efficiency, storage overhead, and security.

Computational efficiency is measured by proof generation time and verification time, directly affecting the speed of blockchain transaction processing. The experiment uses a high-precision timer to accurately time each transaction's proof generation and verification process [19]. Considering the randomness of transactions, the average of 1,000 transaction time data statistics is taken to reduce the interference of accidental factors and accurately reflect the actual efficiency of the algorithm.

Storage overhead is measured by the proof data storage size (bytes) to evaluate the algorithm's consumption of storage resources. The experiment thoroughly counts the space occupied by zero-knowledge proof data in each block. It calculates the average storage overhead by integrating all block data to clarify the storage requirements of different algorithms when processing large-scale transactions. It provides a reference for blockchain storage optimization.

Security is the key to blockchain privacy transactions. The experiment evaluates the algorithm's privacy protection by simulating replay and forgery attacks. For replay attacks, an automated script is designed to send transaction data repeatedly, and the proportion of successful attacks measures the attack success rate; for forgery attacks, a large amount of false transaction data and a proof submission verification system are constructed, and the forgery attack interception rate characterizes the algorithm's security. The higher the interception rate, the stronger the algorithm's ability to resist forgery attacks. Through these two indicators, the privacy protection effect of the algorithm in

the face of common attacks can be effectively evaluated, ensuring that the experimental results are reliable and practical.

4.3 Comparative experimental design

To verify the superiority of the DA-ZKP algorithm in blockchain privacy transactions, the experiment selects three mainstream zero-knowledge proof algorithms, zk-SNARKs, Bulletproofs, and Groth16, as comparison objects. Zk-SNARKs are simple, efficient, and widely used, which can reduce transmission and storage overhead; Bulletproofs are good at reducing the size of proofs; Groth16 has the advantage of short verification time.

The experiment strictly unifies the environment and data set to ensure accurate and fair results. The transaction data is input into the system in batches of 1,000, and the proof generation time, verification time, and storage overhead of the DA-ZKP algorithm and the comparison algorithm are recorded. To reduce the experimental error, each group of experiments is repeated 10 times, and the average value is taken as the final result.

In the security testing phase, 1,000 replay attacks and 1,000 forgery attacks are simulated for each algorithm, and the real attack scenarios are simulated through automated scripts. The number of successful attacks and interceptions is recorded, and the privacy protection capabilities of each algorithm are compared and analyzed with the attack success rate and interception rate as indicators [20]. This experimental design fully demonstrates the advantages and improvements of the DA-ZKP algorithm over traditional algorithms in terms of performance indicators such as efficiency, storage, and security.

4.4 Experimental results

4.4.1 Computational efficiency comparison

Table 1 shows the computational efficiency comparison data of the DA-ZKP and traditional algorithms. The results show that the DA-ZKP algorithm is significantly better than the three comparison algorithms in terms of proof generation and verification time. The average proof generation time of the DA-ZKP algorithm is 128 milliseconds [21], representing a 35% reduction compared to zk-SNARKs (197 ms), 18% compared to Bulletproofs (156 ms), and 9% compared to Groth16 (141 ms). These results are based on Experiment Set 1, which involves 10,000 transaction data points derived from real-world financial and supply chain transactions with 100 distributed nodes (32-core Intel Platinum 8380 processor, 256 GB DDR4 memory). Regarding verification time, the DA-ZKP algorithm only takes 89 milliseconds on average. In comparison, the zk-SNARKs algorithm takes 135 milliseconds, the Bulletproofs algorithm takes 112 milliseconds, and the Groth16 algorithm takes 124 milliseconds.

Table 1: Comparison of computational efficiency of the DA-ZKP algorithm and traditional algorithms.

Algorithm Name	Proof generation time (milliseconds)	Verification time (milliseconds)
DA-ZKP	128	89
zk-SNARKs	197	135
Bulletproofs	156	112
Groth16	141	124

Figure 1 shows the trend of proof generation time for different algorithms when the transaction scale increases from 1,000 to 10,000. The proof generation time of the DA-ZKP algorithm rises relatively slowly, and its advantages become more evident as the transaction scale increases. In contrast, the proof generation time of the zk-SNARKs algorithm increases rapidly, showing a steeper upward trend. The proof generation time of the Bulletproofs algorithm and the Groth16 algorithm also indicates different degrees of growth. Still, the DA-ZKP algorithm always maintains a low proof generation time, which suggests that the DA-ZKP algorithm is more efficient in processing large-scale transactions, can effectively cope with the increase in transaction scale, and provides more efficient support for blockchain privacy transactions.

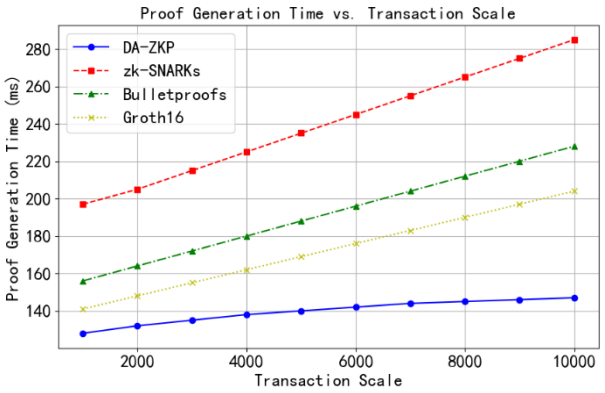


Figure 1: Trends in proof generation time for different algorithms when the transaction size increases from 1,000 to 10,000.

4.4.2 Comparison of storage overhead

Figure 2 depicts the trend of storage overhead of different algorithms as the number of transactions increases. The storage overhead of the DA-ZKP algorithm grows slowly and is always lower than that of other algorithms. This shows that the DA-ZKP algorithm can effectively control the consumption of storage resources when processing large-scale transactions. In contrast, the storage overhead of the zk-SNARKs algorithm and the Groth16 algorithm grows faster, while the storage overhead of the Bulletproofs algorithm grows most significantly.

The DA-ZKP algorithm optimizes the parameters through dynamic weights. It simplifies unnecessary proof components for different types of transactions, thereby minimizing storage requirements while ensuring transaction security, significantly alleviating the storage pressure of blockchain nodes, and improving the system's scalability.

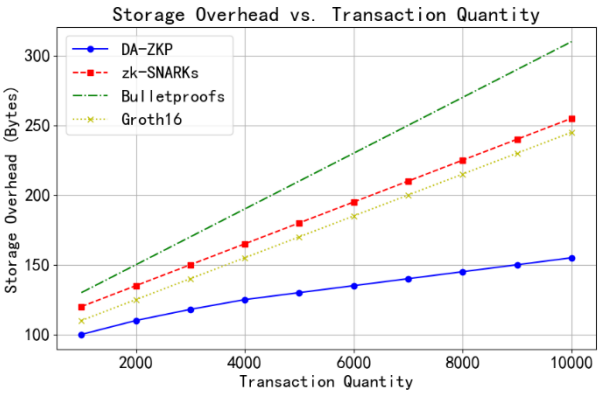


Figure 2: Storage overhead of different algorithms as the number of transactions increases.

4.4.3 Security analysis

In the replay attack simulation experiment, the DA-ZKP algorithm and the three traditional algorithms showed strong resistance, achieving a 100% interception rate, effectively preventing attackers from attempting illegal benefits by repeatedly sending transaction data. In forgery attack tests, DA-ZKP achieved a 99.8% interception rate, outperforming zk-SNARKs (99.5%), Bulletproofs (99.3%), and Groth16 (99.6%). Security Test Set 2 simulated attack intensities from 100 to 1,000 TPS, where DA-ZKP's double verification mechanism-combining data integrity checks and proof validity verification-ensured consistent performance, maintaining >99.5% interception even at peak loads.

Table 2: Security data.

Algorithm Name	Replay attack interception rate	Forged attack interception rate
DA-ZKP	100%	99.80%
zk-SNARKs	100%	99.50%
Bulletproofs	100%	99.30%
Groth16	100%	99.60%

Figure 3 shows the interception performance of each algorithm against replay attacks under different attack intensities. The DA-ZKP algorithm, zk-SNARKs algorithm, Bulletproofs algorithm, and Groth16 algorithm achieved a 100% interception rate, demonstrating strong resistance. This shows that regarding replay attacks, the DA-ZKP algorithm can effectively prevent attackers from attempting to obtain illegal benefits by repeatedly sending transaction data, providing reliable security protection for blockchain privacy transactions. As the attack intensity increases, the DA-ZKP algorithm maintains stable and excellent security performance, fully verifying its reliability in ensuring transaction security.

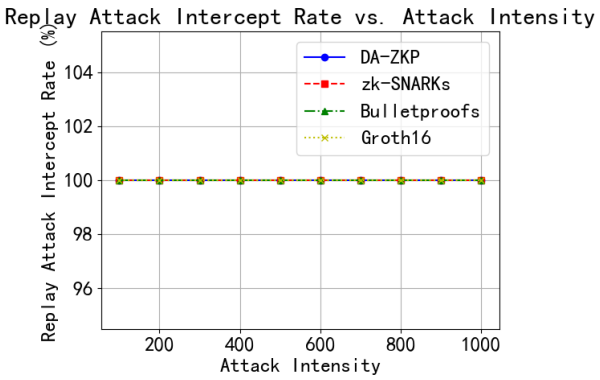


Figure 3: Interception performance of each algorithm against replay attacks at different attack intensities.

Figure 4 shows the interception performance of each algorithm against forgery attacks at different attack intensities. The forgery attack interception rate of the DA-ZKP algorithm reached 99.8%, which is higher than the 99.5% of the zk-SNARKs algorithm, the 99.3% of the Bulletproofs algorithm, and the 99.6% of the Groth16 algorithm. As the attack intensity increases, the DA-ZKP algorithm maintains stable and excellent security performance. This is mainly due to the double verification mechanism adopted by the DA-ZKP algorithm in the verification stage, which can effectively identify forged transactions and prevent them from passing verification, thereby providing more reliable security protection for blockchain privacy transactions, and fully verifying the effectiveness and superiority of the DA-ZKP algorithm in ensuring transaction security.

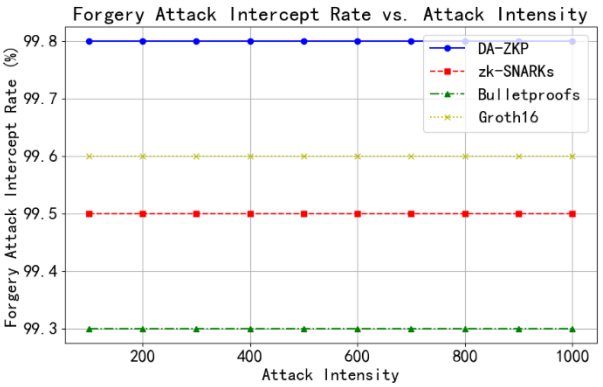


Figure 4: Interception performance of each algorithm for forgery attack under different attack intensities.

5 Conclusion

This paper aims to address the efficiency and security issues of blockchain privacy transactions, constructs an optimization model based on zero-knowledge proof, and designs the DA-ZKP algorithm. Through theoretical derivation and experimental verification, the algorithm significantly improves transaction efficiency. It reduces storage overhead while ensuring privacy and security by relying on dynamic weight allocation and an adaptive adjustment strategy. Experimental data show that compared with traditional algorithms, its proof generation

time is reduced by 35%, verification time by 28%, storage overhead by 22%, and security performance is good. The research results provide a new solution for blockchain privacy transaction technology and are of great significance to promoting the development of the blockchain industry. However, our research has several limitations. First, experiments were conducted in controlled simulation environments, and the algorithm's stability in complex real-world networks-particularly under adversarial conditions-requires validation. Second, the learning-based parameter adjustment introduces a 5ms latency per transaction, which may impact systems with ultra-high throughput (e.g., >10,000 TPS). Third, storage overhead for dynamic parameters is 15% higher than static zk-SNARKs in Ethereum-like networks. Future work will expand scenarios to include cross-chain environments and integrate homomorphic encryption to mitigate these constraints. In the future, the experimental scenarios will be expanded, the adaptability of the algorithm will be optimized, and the integration with technologies such as homomorphic encryption will be explored further to improve the comprehensive performance of blockchain privacy transactions.

References

[1] X. Sun, F. R. Yu, P. Zhang, Z. Sun, W. Xie, and X. Peng, "A survey on zero-knowledge proof in blockchain," IEEE Network, vol. 35, no. 4, pp. 198–205, 2021, doi: 10.1109/MNET.011.2000473.

[2] W. Jiang and X. Lv, "A distributed internet of vehicles data privacy protection method based on zero-knowledge proof and blockchain," IEEE Transactions on Vehicular Technology, vol. 73, no. 5, pp. 6332-6345, 2023, doi: 10.1109/TVT.2023.3345272.

[3] Xue, Z., Wang, M., Zhang, Q., Zhang, Y., & Liu, P. (2021). A regulatable blockchain transaction model with privacy protection. International Journal of Computational Intelligence Systems, 14(1), 1642–1652. https://doi.org/10.2991/ijcis.d.210528.001

[4] Yong, W., Lijie, C., Yifan, W., & Qiancheng, W. (2024). Efficient and secure confidential transaction scheme based on commitment and aggregated zero-knowledge proofs. Journal of Cyber Security Technology, 8(4), 312–332. https://doi.org/10.1080/23742917.2024.2336634

[5] Gao, S., Peng, Z., Tan, F., Zheng, Y., & Xiao, B. (2022). SymmeProof: Compact zero-knowledge argument for blockchain confidential transactions. IEEE Transactions on Dependable and Secure Computing, 20(3), 2289–2301.https://doi.org/10.1109/TDSC.2022.3179913

[6] Onteddu, A. R., Koehler, S., Kundavaram, R. R., Devarapu, K., Kothapalli, S., & Narsina, D. (2024). Artificial Intelligence in Zero-Knowledge Proofs: Transforming privacy in cryptographic protocols. Engineering Intelligence, 12(1), 51–66. https://doi.org/10.18034/ei.v12i1.743

[7] Liu, W., Wan, Z., Shao, J., & Yu, Y. (2021). HyperMaze: Towards privacy-preserving and scalable permissioned blockchain. IEEE Transactions on Dependable and Secure Computing, 20(1), 360–376. https://doi.org/10.1109/TDSC.2021.3133840

[8] Zhang, H., Wu, J., Lin, X., Bashir, A. K., & Al-Otaibi, Y. D. (2023). Integrating blockchain and deep learning into extremely resource-constrained IoT: An energy-saving zero-knowledge PoL approach. IEEE Internet of Things Journal, 11(3), 3881–3895. https://doi.org/10.1109/JIOT.2023.3280069

[9] Hu, X., Zhou, W., Yin, J., Cheng, G., Yan, S., & Wu, H. (2023). Towards verifiable and privacy-preserving account model on a consortium blockchain based on zk-SNARKs. Peer-to-Peer Networking and Applications, 16(4), 1834–1851. https://doi.org/10.1007/s12083-023-01497-7

[10] Li, W., Meese, C., Guo, H., & Nejad, M. (2023). Aggregated zero-knowledge proof and blockchain-empowered authentication for autonomous truck platooning. IEEE Transactions on Intelligent Transportation Systems, 24(9), 9309–9323. https://doi.org/10.1109/TITS.2023.3271436

[11] Y. Guo, Z. Wan, H. Cui, X. Cheng, and F. Dressler, "Vehicloak: A blockchain-enabled privacy-preserving payment scheme for location-based vehicular services," IEEE Transactions on Mobile Computing, vol. 22, no. 11, pp. 6830–6842, 2022, doi: 10.1109/TMC.2022.3193165.

[12] Ray, R. K., Chowdhury, F. R., & Hasan, M. R. (2024). Blockchain applications in retail cybersecurity: Enhancing supply chain integrity, secure transactions, and data protection. Journal of Business Management Studies, 6(1), 206–214. https://doi.org/10.32996/jbms.2024.6.1.13

[13] Datta, S., & Namasudra, S. (2024). Blockchain-based smart contract model for securing healthcare transactions by using consumer electronics and mobile-edge computing. IEEE Transactions on Consumer Electronics, 70(1), 4026–4036. https://doi.org/10.1109/TCE.2024.3357115

[14] Ochigbo, A. D., Tuboalabo, A., Labake, T. T., Buinwi, U., Layode, O., & Buinwi, J. A. (2024). Legal frameworks for digital transactions: Analyzing the impact of blockchain technology. Finance and Accounting Research Journal, 6(7), 1205–1223. https://doi.org/10.51594/farj.v6i7.1313

[15] Wan, Z., Zhang, T., Liu, W., Wang, M., & Zhu, L. (2021). Decentralized privacy-preserving fair exchange scheme for V2G based on blockchain. IEEE Transactions on Dependable and Secure Computing, 19(4), 2442–2456. https://doi.org/10.1109/TDSC.2021.3059345

[16] Anggriani, K., Chiou, S., Wu, N., & Hwang, M. (2023). A Robust and High-Capacity Coverless Information Hiding Based on Combination Theory. Informatica, 34(3), 449-464. https://doi.org/10.15388/23-INFOR521

[17] Blanco-Fernández, Y., Gil-Solla, A., Pazos-Arias, J. J., & Quisi-Peralta, D. (2023). Automatically Assembling a Custom-Built Training Corpus for Improving the Learning of In-Domain

Word/Document Embeddings. Informatica, 34(3), 491-527. https://doi.org/10.15388/23-INFOR527

[18] N. C. Cruz, M. Marín, J. L. Redondo, E. M. Ortigosa, and P. M. Ortigosa, "A comparative study of stochastic optimizers for fitting neuron models: Application to the cerebellar granule cell," Informatica, vol. 32, no. 3, pp. 477–498, 2021, doi: 10.15388/21-INFOR450.

[19] Liang, W., Liu, Y., Yang, C., Xie, S., Li, K., & Susilo, W. (2024). On identity, transaction, and smart contract privacy on permissioned and permissionless blockchain: A comprehensive survey. ACM Computing Surveys, 56(12), 1–35. https://doi.org/10.1145/3676164

[20] Taher, S. S., Ameen, S. Y., & Ahmed, J. A. (2024). Advanced fraud detection in blockchain transactions: An ensemble learning and explainable AI approach. Engineering, Technology & Applied Science Research, 14(1), 12822–12830. https://doi.org/10.48084/etasr.6641

[21] Bennet, D., Maria, L., Sanjaya, Y. P. A., & Zahra, A. R. A. (2024). Blockchain technology: Revolutionizing transactions in the digital age. ADI Journal of Recent Innovation, 5(2), 192–199. https://doi.org/10.34306/ajri.v5i2.1065

Cross-Modal Sentiment Analysis on Social Media using Improved Nonverbal Representation Learning and GHRNN Fusion

Minglun Xue[1*], Dongyang Wang[2]
[1]Department of Primary Education, Jiaozuo Normal College, Jiaozuo 454000, China
[2]Department of Foreign Languages, Shangqiu Normal University, Shangqiu 476000, China
E-mail: minglunxue@outlook.com
[*]Corresponding author's

Keywords: social media, multi-modal data, information extraction, fusion network, representation learning

Received: May 6, 2025

Traditional sentiment analysis methods mainly focus on textual data, while human emotions are multidimensional, usually related to sound, body language, etc. The use of multi-modal data can provide a deeper and broader understanding of human sentiment conveyance. To address the above issues, a method for extracting and analyzing emotional information features based on improved nonverbal representation learning networks and multi-modal data (Improved Nonverbal Representation Learning Networks and MD, INPNRLN-MD) is proposed. On this basis, an improved multi-modal data fusion method based on Gated Hierarchical Recurrent Neural Network and Cross-Modal Attention (GHRNN-CMA) is designed for the MD fusion part. Compared with traditional baselines, INPNRLN-MD extracts text features through the BERT model and utilizes ELN to process audio and video data, which can more effectively capture emotional information in multi-modal data. The cross-modal attention mechanism of GHRNN-CMA can enhance the interaction between modalities and improve the accuracy of emotion information recognition. Finally, the performance of the model is validated on the CMU-OSI and CMU-MCSEI datasets using indicators such as F1 value, Pearson correlation, mean absolute error, and second-order accuracy. During the training process, the study used a single NVIDIA GTX TITAN X GPU for testing, with 12 VRAM and a batch size of only 32 to converge. The inference stage has a relatively light computational load and can be deployed on ordinary cloud servers or edge devices. The research results show that compared with mainstream algorithms, the emotion information feature extraction and analysis method based on improved nonverbal representation learning network and multi-modal data performs the best, with F1 score, Pearson correlation, average absolute error, and second-order accuracy reaching 83.06/85.12, 0.803, 0.696, and 83.17/85.23, respectively. The average absolute error, Pearson correlation, F1 score, and second-order accuracy of the improved multi-modal data fusion method have been improved by 1.0%, 14.67%, 3.1%, and 3.3% respectively compared to the latest method. The above results indicate that research methods are helpful in perceiving and analyzing human emotions, which is beneficial for understanding and predicting human behavior in the future, and is of great significance for maintaining social relationships and improving social governance.

Povzetek: Metoda INPNRLN-MD z BERT in ELN ter fuzija GHRNN-CMA omogočata bolj kvalitetno čezmodalno analizo sentimenta na družbenih omrežjih, saj presegata obstoječe pristope v točnosti in robustnosti.

1 Introduction

Social media has become an indispensable part of daily life, with billions of users sharing information, interacting, and communicating on the platform every day [1]. These massive amounts of data contain useful insights that can aid in comprehending user behavior, analyze trends, and improve products and services [2]. Sentiment analysis is a core task in natural language processing, aimed at identifying emotional states of opinions, emotions, and evaluations [3]. Due to the diversity of social media information, it is of great significance to comprehensively consider text and image information for multi-modal sentiment analysis in order to improve the accuracy of sentiment analysis on social media [4]. In addition, different multi-modal data (MD) from social media can provide very important clues and are vital in enhancing the recognition and detection performance of subtasks [5-8]. Therefore, compared with single-modal analysis, multi-modal analysis of social media data effectively utilizes the relationship and influence between visual and textual information, which not only helps scholars to accurately understand people's attitudes and habits towards life in the real world, but also enables them to grasp people's choices in areas such as healthcare, political topics, TV movies, and online shopping [9]. The research questions are as follows: can specific modalities of nonverbal learning improve emotion classification, and can emotion information

feature extraction and analysis methods based on improved nonverbal representation learning networks and MD more accurately perceive and analyze human emotions. To solve the above problems, a method for extracting and analyzing emotional information features based on improved nonverbal representation learning network and MD (INPNRLN-MD) is proposed. On this basis, an improved MD fusion method based on gated hierarchical recurrent neural network and cross-modal attention (GHRNN-CMA) is designed for the MD fusion part. The research objective is to maintain the accuracy of A7 while reducing the average absolute error and to enhance the overall performance of sentiment analysis. The innovation of the research lies in two aspects. On the one hand, it proposes an emotion information feature extraction and analysis method based on INPNRLN-MD to achieve faster understanding of the emotions and viewpoints that users want to express and share. On the other hand, it designs an improved MD fusion method based on GHRNN-CMA to fully utilize the information differences between different modalities and obtain the optimal feature representation.

2 Related works

In today's rapidly developing world of artificial intelligence, sentiment analysis has become a key tool for understanding human emotions and attitudes. However, traditional sentiment analysis methods are often limited to a single data modality, which limits their comprehensiveness and accuracy. To gain a deeper understanding of human emotions, it is necessary to introduce MD fusion to raise the precision and breadth of sentiment analysis. Numerous scholars have conducted in-depth analysis and exploration on this matter. Lei et al. proposed three sets of paired sorting rules, namely support vector machine, deep neural network, and gradient enhanced decision tree, to generate multi-modal sentiment information analysis for preference learning. The research results showed that compared with the traditional classifier baseline, the performance of the three preference learning models used in the study was better, and the model with gradient enhanced decision tree had the best performance. In addition, the results of the multi-modal emotion dataset annotated by actor performance crowd-sourcing showed that by combining the two best ranking models through research methods, the optimal overall accuracy was 85.06% [10]. Zhu et al. analyzed different multi-modal sentiment analysis techniques, including machine learning and deep learning methods, involving fusion strategies for different modalities. The results showed that multi-modal sentiment analysis techniques made progress in understanding human emotional expression, but still faced challenges in modal fusion and context understanding [11]. Chen et al. proposed a method for processing MD grounded on K-Meand and kernel canonical correlation analysis. The experiment outcomes indicated that the research method validly improved the heterogeneity between different modalities, promoted multi-modal emotion recognition, and greatly improved the recognition rate. Its performance on the Sari audio-visual expression emotion dataset was 2.77% higher than the average level, and on the enterface'05 audio-visual emotion dataset it was 4.7% higher [12]. Mai et al. designed a new framework for three mode mixed comparative information to fully explore cross-modal interaction and reduce modal gap. Numerous experiments showed that the research method was superior to the baseline in multi-modal sentiment analysis and emotion recognition [13]. Chiorrini et al. focused on a large amount of unstructured data online, which contains users' emotions and feelings towards various topics that cannot be achieved through traditional search engines. Therefore, they developed a new architecture for emotion aware search engines and used deep learning-based emotion recognition algorithms to extract emotion vectors. The results confirmed the superiority of this method [14]. To explore how to mine the required information from massive Internet data, Fang et al. designed a method based on convolutional neural network and time convolutional network, and added multi-modal attention mechanism and cross-modal transformer structure on this basis. The findings indicated that the precision of this technique was 2.88% higher than that of using time convolutional network alone [15]. Gupta et al. aimed to investigate emotional disorders through emoticons, and therefore proposed a multi-modal emotion recognition method based on a multi-view ensemble learning model. The outcomes indicated that the method provided an accuracy of 88.29 [16]. The summary table of the relevant works mentioned above is shown in Table 1.

Table 1: Summary table of related works

Literature	Method	Data set	Key performance	Limitation
[10]	Multi model ensemble based on preference learning	Actor performance crowdsourcing annotation dataset	The optimal combination accuracy is 85.06%	Unresolved modal heterogeneity issues and lack of dynamic interaction modeling
[11]	Overview of Multi-modal Technologies	-	Pointing out that modal fusion and contextual understanding are the main challenges	Unprocessed non aligned sequence issues
[12]	K-Means+nuclear canonical correlation analysis	urrey AVED、enterface'05	The recognition rate has increased by 2.77% and 4.7%, respectively	Unmodeled high-order nonlinear modal interactions, ignoring

				temporal dynamics
[13]	Three mode hybrid comparative learning framework	CMU-MOSI、CMU-MCSEI	Better than baseline	Explicitly processing modal redundancy information
[14]	DL based emotion perception search engine	Unstructured data on the internet	Better than baseline	Unsolved problem of imbalanced MD fusion
[15]	CNN+TCN+Multi-modal Attention	E-commerce product marketing data	Compared to TCN, the accuracy has increased by 2.88%	Attention mechanism does not differentiate modal contributions
[16]	Multi view ensemble learning model	Social media data containing emoticons	The accuracy rate is 88.29%	Unprocessed modal missing scenarios

Based on the above content, it can be concluded that current research mainly focuses on using MD for sentiment analysis and applying it in various fields. However, existing methods have significant shortcomings in non-verbal representation optimization, modal dynamic fusion, and cross-modal interaction modeling. The design of INPNRLN-MD and GHRNN-CMA directly addresses these shortcomings and can effectively compensate for the shortcomings of existing research. Therefore, the study proposed an emotion information feature extraction and analysis method based on INPNRLN-MD and an improved MD fusion method based on GHRNN-CMA.

3 Emotional information extraction and analysis for MD on social media

To fully utilize the massive emotional information of social media MD, a feature extraction method based on INPNRLN-MD was studied and designed, and an improved MD fusion method based on GHRNN-CMA was proposed to improve the fusion part. INPNRLN-MD is a component for feature extraction and preliminary analysis, which is part of the GHRNN-CMA method.

3.1 Emotional information feature extraction and analysis method based on INPNRLN-MD

Social media, as an important platform for information exchange in people's daily lives, has accumulated a massive amount of user generated content in recent years. The information forms are diverse, including text, audio, images, and other data forms [17,18]. Leveraging MD to fully grasp users' emotional expressions and enhance sentiment analysis performance is crucial in the realm of sentiment analysis. Seamlessly integrating intricate multi-modal interactions to uncover potentially valuable emotional features poses a significant challenge in this field. Therefore, in this section, a sentiment information feature extraction method based on INPNRLN-MD is proposed, which focuses on extracting sentiment features from different modalities such as text, audio, and video, and provides input for subsequent sentiment analysis. The MD sequence of this model mainly includes three modalities: text, audio, and video. Assuming that the modal information of text, audio, and video are represented as t, a, and v, respectively, and the input content can be represented by $z \in \{t,a,v\}$, where the sequence length and corresponding feature dimensions of the above content are represented as $L_{z\in\{t,a,v\}}$ and $d_{z\in\{t,a,v\}}$, respectively. The emotional information feature extraction part of the text modal data is completed using a pre-trained Bidirectional Encoder Representation from Transformers (BERT) model, as shown in Figure 1.

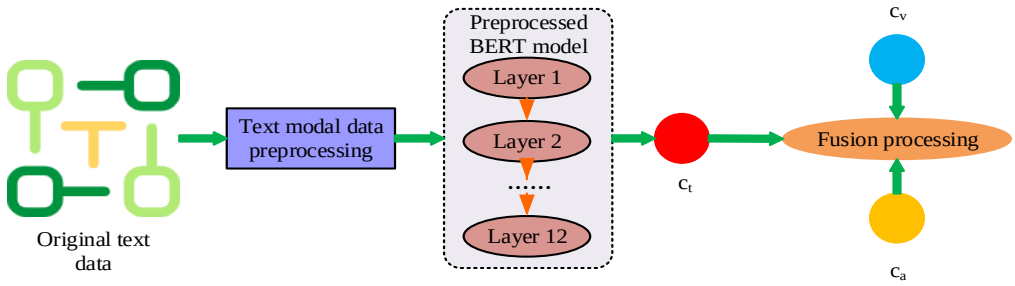


Figure 1: Schematic diagram for feature extraction of text modal emotional information

In Figure 1, the input text modality data is fed into a preprocessed BERT model to obtain the feature representation c_t in the text modality. It is then processed by outer product with the feature representations obtained from other modality data to obtain the final shareable feature space. The interactive representation is then obtained through contrastive learning and transmitted to the fully connected layer for sentiment prediction. The BERT model used in the study is stacked through 12 layers of Transformers and introduces a special label, the classification label $[CLS]$, whose last hidden state serves as the aggregated representation of the entire sequence; The separator mark $[SEP]$ is used to separate different sentences, marking the end of the first sentence and the beginning of the second sentence [19,21]. The above data are transferred to the BERT model together to complete the word embedding, and the first word vector of the 12th layer is used as the representation, where the implied state is represented by the average value, as shown in equation (1).

$$\begin{cases} t_i = \{[CLS], word_1, word_2, \cdots, word_n, [SEP]\} \\ c_t = BERT \quad (t_i, \zeta_t^{\text{BERT}}) \in R^{d_t}, i \in [1, n] \end{cases} \tag{1}$$

In equation (1), $word_n$ and n represent the words and their quantities in the discourse, ζ_t^{BERT} represents the hyperparameters required for the BERT model, and R^{d_t} represents the space of the output vector. After the emotional information extraction of text modal data is completed, the emotional information feature extraction of audio modal data can be carried out, as shown in Figure 2.

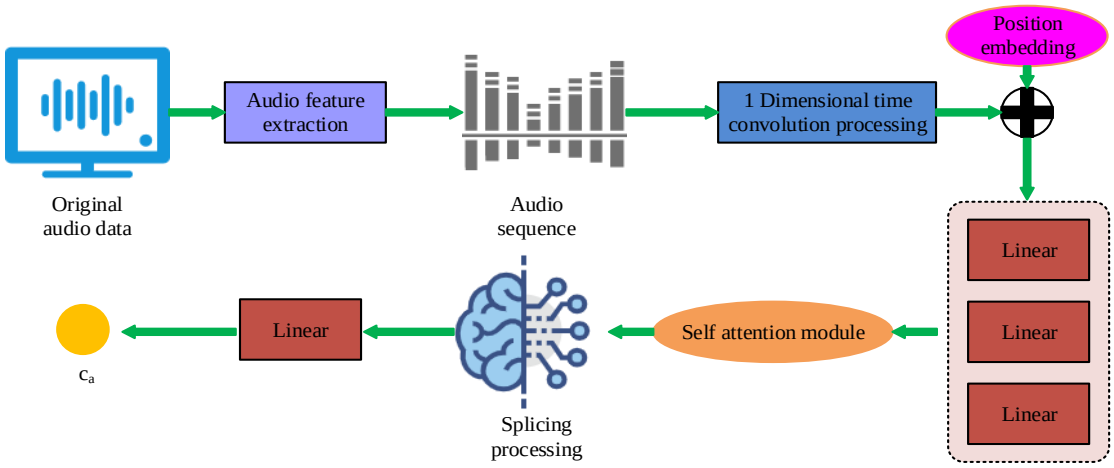


Figure 2: Schematic diagram of feature extraction of emotional information in audio modality

In Figure 2, the original audio data is first transformed into acoustic features that can be used for subsequent processing through a one-dimensional temporal convolutional layer, and then input into the Expressing Learning Network (ELN). Next, position embedding vectors are added, and then processed sequentially through linear transformation and mean weighted filters. Finally, the desired audio feature representation c_a is output. The emotional information feature extraction of video modal data is similar to that of audio modal data, both using ELN, but with certain differences in the feature extractor part. Firstly, the study applies a 1D time convolution layer to the input sequence for processing, in order to enhance the perceptual power of modal data. The calculation is shown in equation (2).

$$F_k = \text{Conv1}D_k(k, q_k) \in R^{T_k \times d} \tag{2}$$

In equation (2), F_k represents the processed feature representation, $\text{Conv1}D_k(\square)$ represents the 1D time convolution function, and k and q_k represent the input sequence and convolution kernel size of the corresponding modal data. To assign time information to a sequence, it can be solved by introducing positional embedding, and then implementing a self attention module on the processed sequence, as expressed in equation (3) [22].

$$\text{A}_k(Q, K, V) = \text{softmax}\left(\frac{Q_k K_k^{TN_k}}{\sqrt{d_k}}\right) V_k, k \in \{a, v\} \tag{3}$$

In equation (3), $\text{softmax}|\square|$ is the softmax function, Q, K, and V correspond to the query vector, key vector, and value vector of the self attention mechanism, respectively. TN and d_k correspond to the vector transpose and the dimension of the corresponding modality. Then, to calculate the heads of each attention, it is necessary to linearly project the matrix through the corresponding weight matrix. Meanwhile, it can connect the heads of all modalities to obtain a multi-head self attention mechanism, as shown in equation (4).

$$Y_k = Concat\left(h_k^1, h_k^2, \cdots, h_k^a\right) W_k^o \tag{4}$$

In equation (4), $Concat(\square)$ and W_k^o represent the weight matrices used for the stitching operation and stitching processing, respectively, while h_k^a represents the head of the corresponding model's a th attention. From this, the feature representations c_a and c_v corresponding to the audio modality and video modality can be obtained. Based on the above content, a schematic diagram of extracting emotional information features from video modalities can be obtained, as shown in Figure 3.

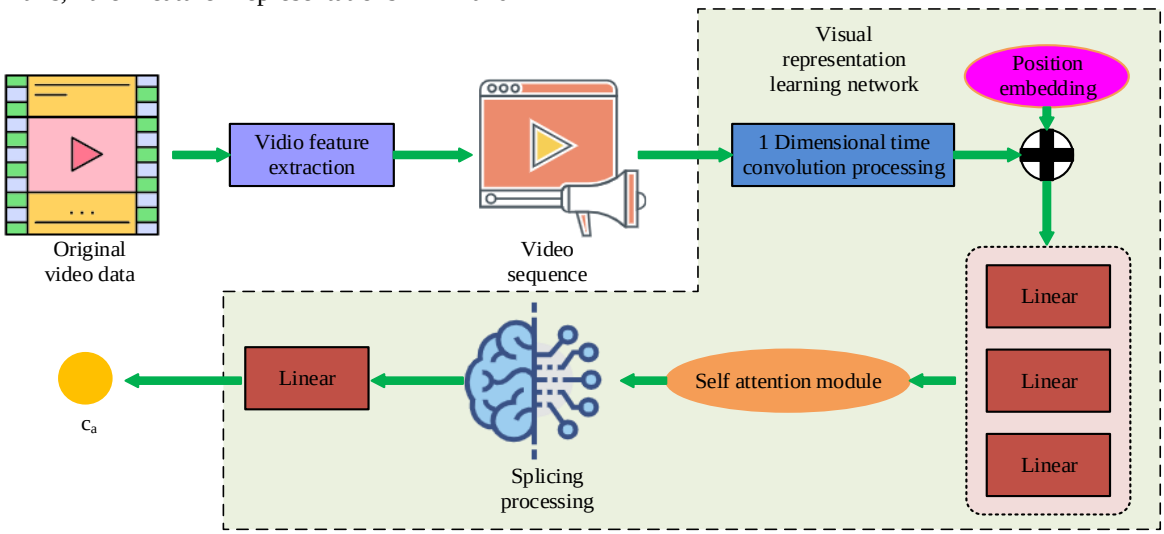


Figure 3: Schematic diagram of feature extraction of emotional information in video modality

In Figure 3, the original video modality data is processed by a 1D time convolution layer to obtain the required feature representation, which is then input into ELN for processing and position embedding. Finally, through concatenation and linear projection, the final emotional information feature representation of the video modality can be obtained. After the above data processing is completed, the outer product operation can be performed to obtain the corresponding result, as shown in equation (6).

$$\begin{cases} c_{\alpha\otimes t} = c_\alpha \otimes c_t \\ c_{v\otimes t} = c_v \otimes c_t \end{cases} \quad (6)$$

In equation (6), $c_{\alpha\otimes t}$ and $c_{v\otimes t}$ are the outer product results of the audio mode and video mode, respectively, $\otimes$ represents the dot product operation, and the dimension of the obtained vector is $m*n$. The processed MD feature representations are concatenated and projected onto R^{d_z}, resulting in the final output representation c_z. This representation can be applied in prediction, as expressed in equation (7).

$$\begin{cases} B_z = \mathrm{Re}LU\left(W_{l2}^{zT} \otimes c_z + f_{l2}^z\right), W_{l2}^{zT} \in R^{d_z \times 1} \\ e = \dfrac{1}{N}\sum_i^N \left(\left|B'_z - B_z\right|\right) \end{cases} \quad (7)$$

In equation (7), W_{l2}^{zT}, f_{l2}^z, and $\mathrm{Re}LU(\square)$ represent the processed weight matrix, bias, and modified linear unit activation function, e and N correspond to the average absolute error and sample size, and B'_z represents the human label. In order to fully utilize the information of MD, the study introduces contrastive learning to handle the dynamics between different modalities, and constructs the loss function of the model, as shown in equation (8).

$$Loss_{total} = e' + Loss_{in} \quad (8)$$

In equation (8), $Loss_{total}$, e', and $Loss_{in}$ correspond to the overall loss function, the e value between the true emotion and the predicted emotion, and the interaction comparison loss between multiple modalities.

3.2 Improved MD fusion method based on GHRNN-CMA

Due to the unique feature dimensions and representation methods of different modalities of data, as well as differences in quantity and scale, there may be imbalances in MD fusion, and the above methods cannot fully consider the role of interaction information between modalities in emotion recognition and prediction. Therefore, the research designs an improved MD fusion method based on GHRNN-CMA, which not only includes the feature extraction part in INPNRLN-MD, but also further fuses and analyses these features through GHRNN and cross-modal attention mechanism to achieve more accurate emotion prediction. The specific process is shown in Figure 4.

In Figure 4, it mainly includes the feature extraction part based on INPNRLN-MD for corresponding modalities, GHRNN part, and cross attention part for each modality. The feature extraction part of the

corresponding mode remains unchanged, while the two modal outputs and text modal representation of the outer product operation are transmitted to the GHRNN part together, which achieves the acquisition of the final predictive emotional information by removing redundant information. In addition, the different components of the model are very flexible and can be reorganized with different baselines to be applicable to various types of tasks. GHRNN schematic diagram, see Figure 5.

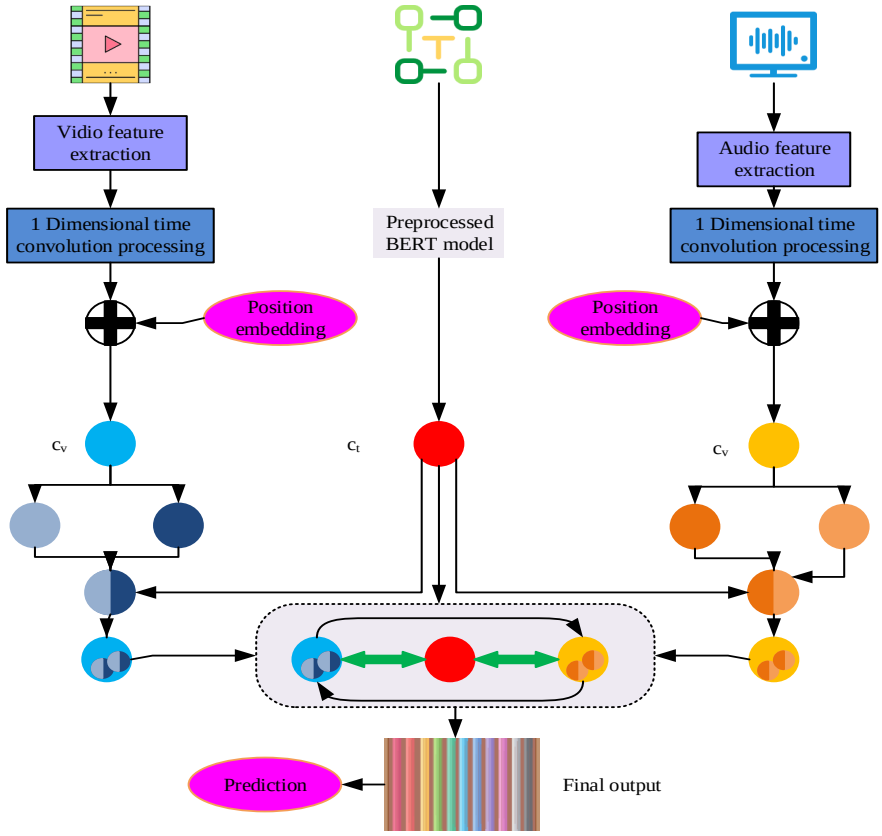


Figure 4: Flow diagram of improved MD fusion method based on GHRNN-CMA

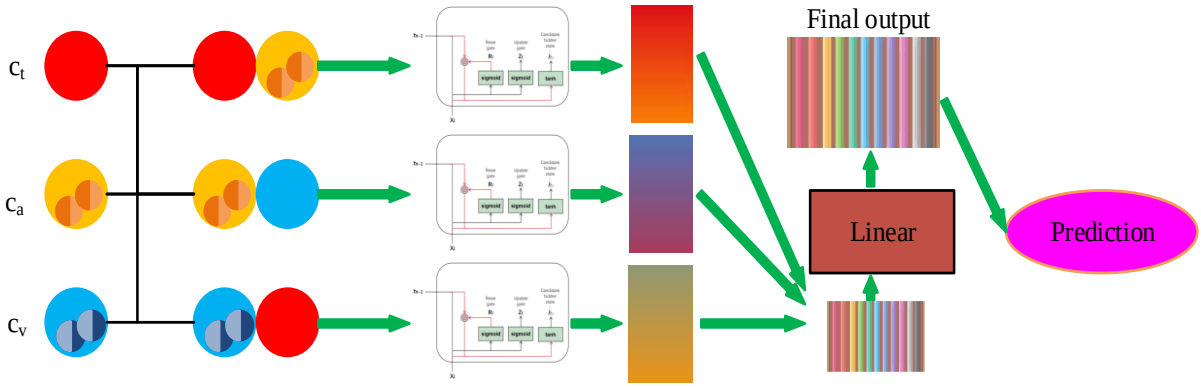


Figure 5: GHRNN schematic diagram

In Figure 5, the input text feature representation c_t, as well as the text-based acoustic and visual representations c_{α_t} and c_{v_t}, are processed using GHRNN. Due to the higher contribution of the text modality to sentiment information analysis compared to other modalities, the study combines the text modality feature representation with other representations to ensure the highest weight of the text modality. As a result, different representation combination expressions can be obtained, as shown in equation (9).

$$\begin{cases} c_{\alpha_t \otimes t} = Concat\left(c_t, c_{a_t}\right) \\ c_{v_t \otimes t} = Concat\left(c_t, c_{v_t}\right) \\ c_{\alpha_t \otimes v_t} = Concat\left(c_{a_t}, c_{v_t}\right) \end{cases} \quad (9)$$

In equation (9), $c_{\alpha_t \otimes t}$ and $c_{v_t \otimes t}$ represent the combination of text modal feature representation with c_{α_t} and c_{v_t}, respectively, while $c_{\alpha_t \otimes v_t}$ represents the combination of c_{α_t} and c_{v_t}. The above combination is input into GHRNN to achieve rich perception of information between various modal data, and irrelevant and redundant information is deleted. The calculation is shown in equation (10).

$$\begin{cases} c'_{\alpha t} = GHRNN\left(c_{a_t \otimes t}, \zeta'\right) \\ c'_{vt} = GHRNN\left(c_{v_t \otimes t}, \zeta'\right) \\ c'_{\alpha v} = GHRNN\left(c_{a_t \otimes v_t}, \zeta'\right) \end{cases} \quad (10)$$

In equation (10), $GHRNN(\bullet)$ and 2 represent GHRNN processing and its hyperparameters, respectively. Then transmit the output result to the fully connected layer for prediction, as expressed in equation (11).

$$\begin{cases} c_z = \text{Concat}\left(c'_{at}, c'_{vt}, c'_{av}\right) \\ \text{‘}c_z = \text{Re}LU \quad \left(W_{l1}^{zT} \otimes c_z + f_{l1}^z\right) \end{cases} \quad (11)$$

In equation (11), $W_{l1}^{zT} \in R^{d_z \times (d_a + d_v + d_t)}$ and f_{l1}^z represent the weights and biases processed by GHRNN, respectively. The calculation of the internal mechanism of GHRNN is shown in equation (12).

$$\begin{cases} z_t = \sigma(W_{xz} x_t + W_{hz} h_{t-1} + b_z) \\ h_t^{'} = \tanh(W_{xh} x_t + W_{hh}(z_t \square\ h_{t-1}) + b_h) \\ h_t = (1 - z_t) \square\ h_{t-1} + z_t \square\ h_t^{'} \end{cases} \quad (12)$$

In equation (12), z_t, σ, and tanh represent the gate control signal, sigmoid activation function, and hyperbolic tangent activation function, respectively. $h_t^{'}$ and h_t represent the candidate hidden state and the final hidden state, respectively. Based on the above content, the construction of an improved MD fusion method based on GHRNN-CMA can be completed. To evaluate the performance of research methods in emotional information extraction and analysis, the current mainstream F1 score, Pearson Correlation (PC), Mean Absolute Error (MAE), Accuracy 2 (A2), and Accuracy 7 (A7) were selected for analysis. The F1 score is the harmonic mean of precision and recall, used to measure the accuracy of the model in classification tasks, while the PC value is used to measure the linear relationship between two random variables, calculated by the covariance and standard deviation between the two variables, with a result range of -1 to 1. MAE is the average absolute error between predicted and observed values, and the smaller the value, the smaller the prediction error of the model. A2 is the accuracy of the model in second class classification tasks, while A7 represents the accuracy of the model in seventh class classification tasks.

4 Emotional information extraction and analysis results for MD

To verify the effectiveness and feasibility of the research method, the performance of the emotion information feature extraction and analysis method based on INPNRLN-MD was analyzed, and further performance comparison experiments and ablation experiments were set up to explore the effectiveness of the improved MD fusion method based on GHRNN-CMA.

4.1 Results of sentiment information analysis based on INPNRLN-MD

To investigate the effectiveness of the sentiment information analysis method based on INPNRLN-MD, the experimental setup was configured as follows: a deep learning framework was employed, utilizing Pytorch 2.5.1 software, and the training process was conducted on an NVIDIA GTX TITAN X GPU. The experimental parameters were set as follows: the initial learning rates of the BERT model and other parameters were $5*10^{-5}$ and 10^{-43}, respectively. The learning rate for audio modal data processing was set to 10^{-4}. This study used the Adam optimizer with a weight decay coefficient of 0.01, set the iteration count to 500, batch size to 32, epochs to 30, Dropout to 0.3, and L2 regularization with a regularization coefficient of 0.00001. The experimental dataset used the multi-modal Corpus of Sentiment Intensity dataset (CMU-MOSI) and multi-modal Corpus of Sentiment and Emotion Intensity dataset (CMU-MCSEI) developed by Carnegie Mellon University (CMU). The CMU-MOSI dataset contains 2199 opinion video clips, each of which is annotated with emotional intensity. The CMU-MCSEI dataset is currently the largest multi-modal sentiment analysis and recognition dataset, containing over 23500 sentence videos from more than 1000 YouTube speakers with multi-label features. In terms of text data preprocessing, this study first removed HTML tags, special characters, and excess whitespace characters from the text, and uses natural language processing libraries such as SpaCy or NLTK for word segmentation processing. Then, the study restored the word to its stem form and removed commonly-used words. Finally, the pre trained BERT model is used to encode the text, generate word embedding representations, and fill all text sequences to the same length. In terms of audio data preprocessing, this study pre emphasized the audio signal by segmenting it into 20ms frames and normalizing the extracted features. The sample distribution and partitioning of the two datasets are shown in Table 2.

According to Table 2, the sentiment intensity of both datasets ranged from very negative to very positive, divided into 7 categories, and there were significant differences in the distribution of sentiment categories. The datasets used in the experiment were all publicly available, and their collection and use underwent corresponding ethical review and user consent procedures. In practical application scenarios, user privacy and sensitive issues were designed. Therefore, when handling social media data, it was necessary to ensure the anonymity and de identification of the data to avoid the leakage of users' personal information. Meanwhile, explicit consent from users was required to ensure that they had sufficient right to know and control over the use of their data. The specific training validation cycle is as follows: first, a pre trained BERT model is used to extract the features of the text modality and fuse them with the features of audio and video. To enhance the interaction between modalities, a multimodal data fusion method based on GHRNN-CMA is used for processing. During training, the Adam optimizer was used and the learning rate was adjusted according to experimental settings to ensure that the model can effectively learn emotional features. Each training cycle includes forward propagation, loss calculation, and backpropagation. The loss function combines cross entropy loss for sentiment classification and multimodal contrastive learning loss to ensure that the model can learn robust sentiment feature representations. To monitor model performance and prevent overfitting, the study validates the model at the end of each training cycle. The verification process includes calculating indicators such as F1 value, Pearson correlation coefficient, and mean absolute error. If the performance of the model on the validation set is better than the previous best performance, the weights of the current model are saved. At the same time, record the performance metrics on the validation set for subsequent analysis of the model's stability. If the model does not show significant performance improvement in multiple consecutive training cycles, the training should be terminated early to avoid overfitting. To verify the performance of research methods more scientifically, comparative experiments were conducted using current mainstream methods, including Fusion Matching of Long Short-erm Memory Networks and Syntactic Distance (FM-LSTM-SD), Dual Attention Convolutional Neural Network (DA-CNN), Multi-Scale Convolution and Gating Mechanism (MSC-GM), and Pay Attention to Interaction in Location Information (PAILI). To obtain fair results, the study conducted repeated validation experiments and recorded the average of twenty runs to compare the performance results of different methods on two datasets, as shown in Figure 6.

Table 2: Sample distribution and partitioning of two datasets

Type	Emotional intensity/categorization	CMU-MOSEI	CMU-MOSI
Sample distribution of dataset	Very negative	816	185
	Negative	2231	440
	Relatively negative	3547	398
	Neutrality	4998	96
	Relatively positive	7429	361
	Positive	3170	482
	Very positive	665	237
Dataset partitioning	Training set	15998	1542
	Validation set	2286	219
	Test set	4572	438

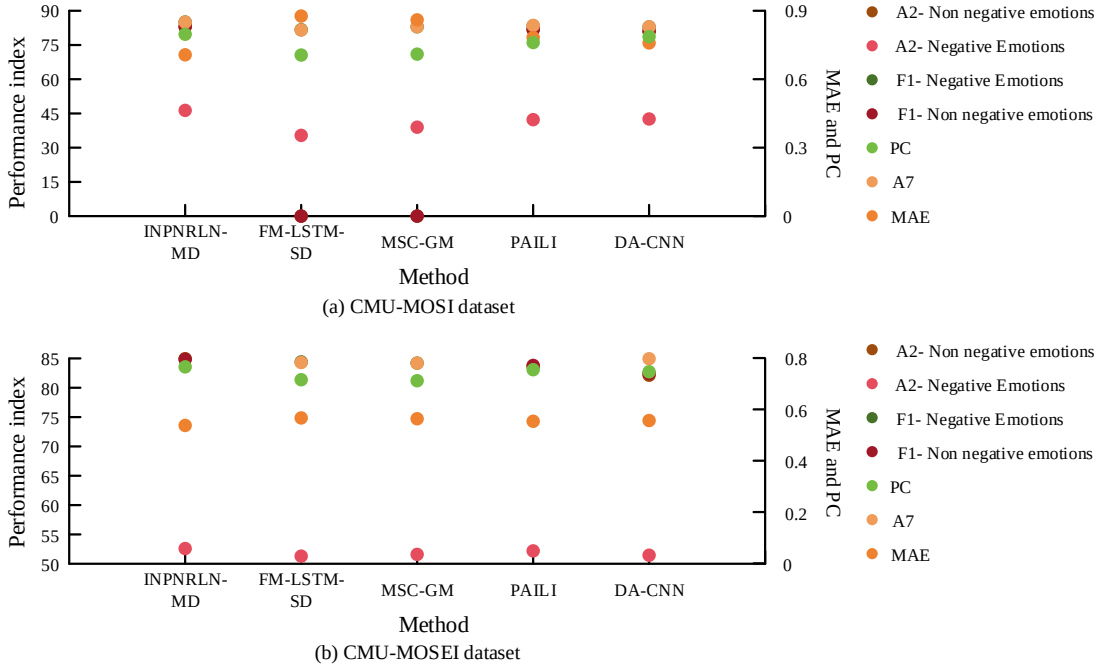
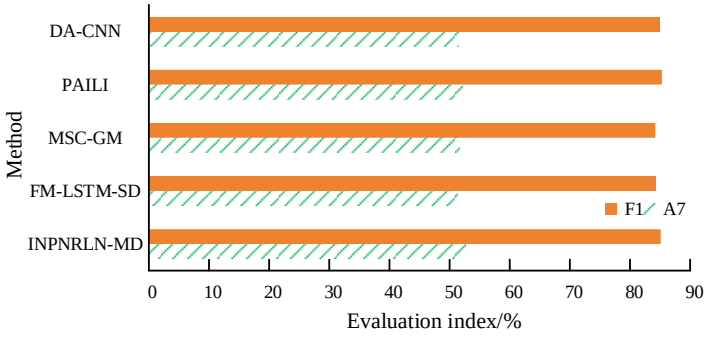


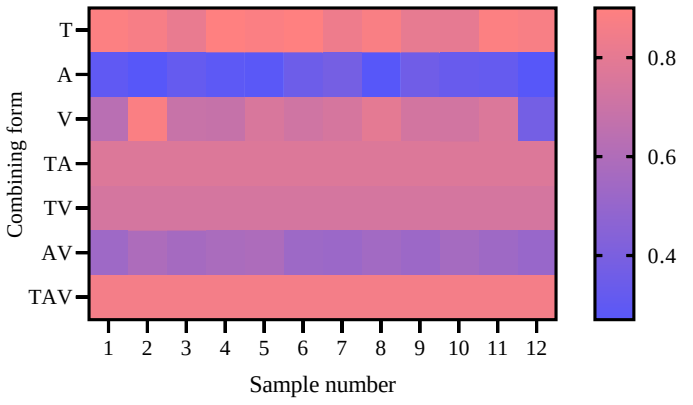
Figure 6: Comparison of performance results of different sentiment analysis methods on two datasets

Figure 6 (a) shows the performance comparison of different sentiment analysis methods on the CMU-MOSI dataset. The sentiment information feature extraction and analysis method based on INPNRLN-MD performed the best in all aspects, with F1 score, PC, MAE, A2, and A7 corresponding to 83.06/85.12, 0.803, 0.696, 83.17/85.23, and 46.41, respectively. The performance of the DA-CNN model was poor, with an MAE value as high as 0.883, a PC value of only 0.705, and an A7 value of 35.6. The MAE of the proposed INPNRLN-MD method was 0.696, which was significantly lower than the 0.883 of the DA-CNN method, indicating that INPNRLN-MD was more accurate in predicting emotional intensity. This meant in practical applications that the prediction of user emotions was closer to the true values, which helped to improve user experience and service quality. The suggested method's observed performance advantage most likely resulted from its capacity to better mimic human language patterns utilizing the CMU-MOSI dataset and completely use the underlying information in MD. In contrast, the DA-CNN model produced less than ideal results since it was less accurate at identifying important textual parts, despite being good at collecting local information. Figure 6 (b) shows the results of different sentiment analysis methods on the CMU-MCSEI dataset. The research method still exhibited excellent performance on unaligned sequences, with a significantly lower MAE value than other methods, as low as 0.527, an F1 score of 84.93/85.12, and a PC value of 0.771. To further analyze the performance of the research method, the CMU-MOSI dataset was selected for exploration, and the most critical indicators, A7 and F1 scores, were chosen. Meanwhile, 12 samples were randomly selected to analyze the impact of different modalities. The results are shown in Figure 7.

Figure 7 (a) shows the performance of different methods on important indicators. The INPNRLN-MD method outperformed other mainstream methods significantly and had the best performance on the A7 indicator, with an improvement of 28.3% compared to the FM-LSTM-SD method and an improvement of about 8.6% on the F1 score indicator. Figure 7 (b) shows the performance results of different modal combination forms of the research method. The combination form with text modality always performed better than other combination forms, and the combination form containing the three modalities t-a-v performed the best in each sample, all exceeding 0.8. This confirmed the correctness of improving nonverbal sequences.



(a) Performance of different methods on important indicators



(b) Performance results of different modal combinations

Figure 7: Performance of different methods based on CMU-MOSI dataset

4.2 Results analysis of improved MD fusion method based on GHRNN-CMA

To further validate the performance of the improved MD fusion method based on GHRNN-CMA, the latest MD fusion methods were selected for comparative experiments, namely the Attention Network for MD Multi-level Feature Fusion (AN-MDMLFF), Perceptual Resampling (PR), Multi-modal Fusion Feature Distillation Control of Global Convolution and Affinity (MMFFD-GCA), and Adaptive Gate Control Information (AGCI). From this, a comparison of the predictive performance of different methods on different datasets can be obtained, as shown in Table 3. Alignment refers to whether the model can handle the temporal or structural inconsistencies that may exist between different modal data. Alignment models are able to handle the inconsistencies between different modal data, such as when text, audio, and video data are not completely synchronized in time, the model can still accurately fuse these data and perform sentiment analysis. Unaligned models assume that all modal data are completely synchronized in time, which may not effectively handle common modal inconsistencies in practical applications.

Table 3 shows that in the CMU-MOSI dataset, compared with the homogeneous PR method, the GHRNN-CMA method reduced the MAE value by 1.0% on the regression task, while in the PC metric, it improved by 14.67% compared to the AN-MDMLFF method. In the comparison of classification tasks, compared with the non-homogeneous AGCI method, the GHRNN-CMA method improved the A2 value and F1 score by about 3.1% and 3.3%, respectively. In the CMU-MCSEI dataset, the GHRNN-CMA method achieved good results in comparing various performance indicators. To test the performance of the GHRNN-CMA method, ablation experiments were conducted, and the impact of cross-modal attention interaction combinations on the research method is shown in Figure 8.

Figure 8 (a) and Figure 8 (b) show the performance comparison of different cross-modal attention interaction combinations on the CMU-MOSI dataset and CMU-MCSEI dataset, respectively. Figure 8 shows that the interaction combination of two modalities in both datasets exhibited poor performance. This was due to the high proportion of text modality weights, which resulted in very low modality independence. Therefore, cross-modal attention interaction between nonverbal sequences was difficult to meet the needs of emotional information feature extraction and analysis. In the combination form of cross-modal attention interaction, the performance of text-based acoustic and visual representation combination was the best, which may be due to the text modality strengthening the emotional complementary information of acoustic and visual, effectively eliminating the ambiguity between emotions and semantics. As shown in Figure 9, 12 samples were randomly selected from the CMU-MOSI dataset for subsequent visualization experiments.

Table 3: Comparison of predictive performance of different MD fusion methods on different datasets

Data set	Method	Is the model aligned?	F1 score	PC	MAE	A2	A7
CMU-MOSI	GHRNN-CMA	Yes	83.39/85.32	0.813	0.705	83.41/85.26	45.82
	AN-MDMLFF	Yes	-/83.4	0.709	0.857	-/83.4	40.50
	MMFFD-GCA	No	82.2/84.2	0.758	0.796	82.2/83.8	-
	AGCI	No	82.63/84.27	0.792	0.743	82.63/84.29	41.80
	PR	Yes	84.50/84.26	0.797	0.712	84.50/84.26	-
CMU-MCSEI	GHRNN-CMA	Yes	85.12/85.24	0.773	0.524	85.36/85.46	53.82
	AN-MDMLFF	Yes	-/84.4	0.715	0.562	-/84.4	51.80
	MMFFD-GCA	No	84.1/85.6	0.757	0.547	83.7/85.6	-
	AGCI	No	83.86/85.12	0.751	0.538	83.83/85.26	52.30
	PR	Yes	82.64/85.16	0.762	0.533	82.83/85.24	-

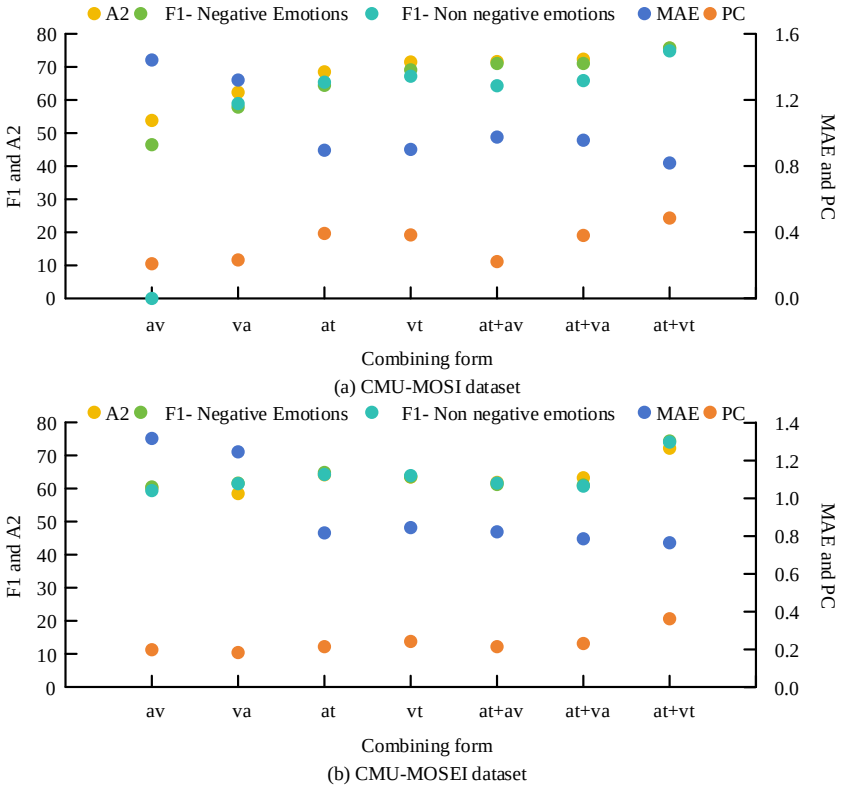
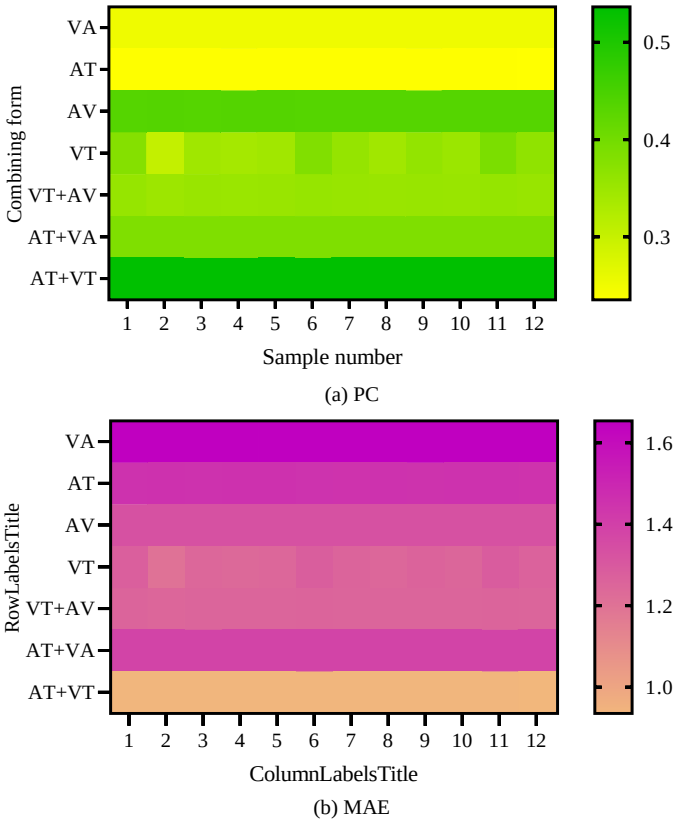


Figure 8: Comparison of performance results of different cross-modal attention interactions in two datasets

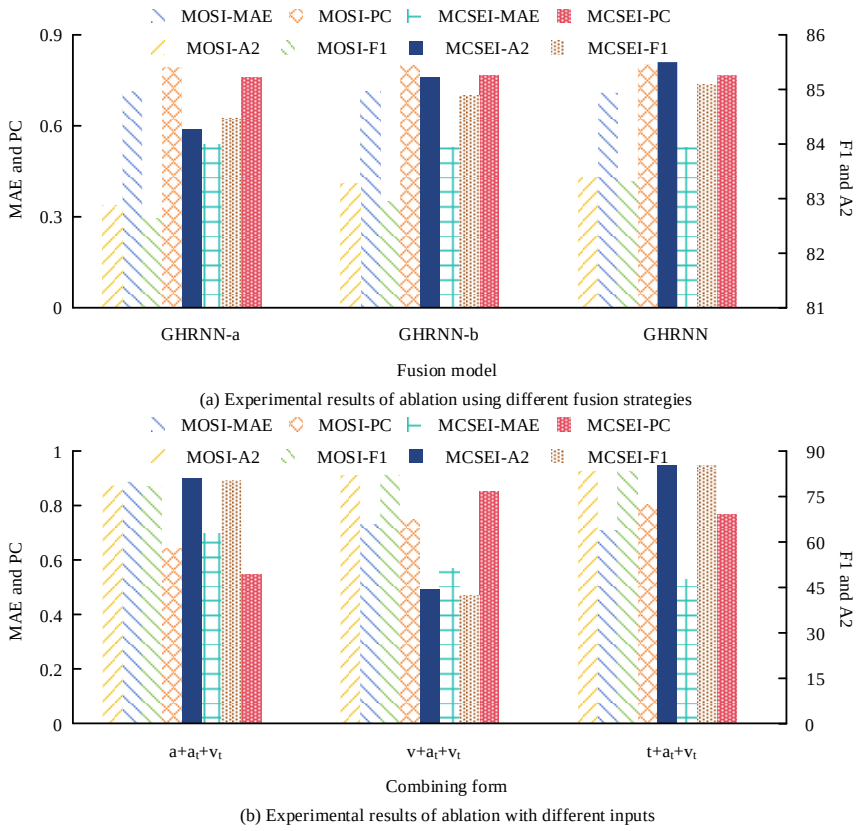
Note: 'av': a combination of audio and video modalities only; 'va': a combination of only video and audio modalities; 'at': a combination of text and audio modalities; 'VT': a combination of text and video modalities; 'at+av': a combination of text, audio, and video modalities, where text interacts with audio and video separately; 'at+va': a combination of text, video, and audio modalities, where text interacts with video and audio separately; 'at+vt': a combination of text, audio, and video modalities, where audio and video interact and combine with text.



(a) PC

(b) MAE

Figure 9: Comparison of visualization results of different cross-modal attention interaction combinations



(a) Experimental results of ablation using different fusion strategies

(b) Experimental results of ablation with different inputs

Figure 10: Experimental results of ablation using research methods based on different datasets

Figures 9 (a) and 9 (b) show the visualization results of different cross-modal attention interaction combinations on PC and MAE, respectively. Figure 9 shows that the combination of acoustic and visual cross-modal attention interaction based on text modality had the best performance, with PC values ranging from 0.45 to 0.5 and MAE values around 0.5. The above results may be due to the effective cross-modal interaction information fusion achieved by the above combination, and the reduction of semantic differences between different modalities. The reliability of GHRNN was tested by selecting two typical indicators from regression and classification tasks for analysis. From this, the experimental results of ablation based on different datasets could be obtained, as shown in Figure 10.

Figure 10 (a) shows the results of ablation experiments with different fusion strategies. The GHRNN model showed the best performance, while the GHRNN-a model, which removed the combination of cross-modal attention interaction representations, only performed better on some indicators. The GHRNN-b model, which was replaced with a Long Short-Term Memory (LSTM) neural network, had the worst performance. The PC, MAE, F1 scores, and A2 index of GHRNN in the CMU-MOSI dataset were 0.817, 0.695, 83.46%, and 83.49%, respectively. In the CMU-MCSEI dataset, the PC, MAE, F1 scores correspond to the A2 index at 0.769, 0.523, 85.14%, and 85.34%, respectively. To enhance the credibility of the ablation experiment, this study conducted five independent runs and recorded corresponding indicator values, each under different random initialization conditions. The performance difference between the two was further evaluated through t-test, and the results showed that the GHRNN model was superior to the GHRNN-a model in terms of PC and MAE indicators, with statistical significance (p<0.05). Figure 10 (b) shows the results of ablation experiments with different inputs, where the combination of at and vt representations was the two highest contributing cross-modal attention interaction representations. The study combined it with a single initial representation a, v, and t to obtain the best performing hierarchical fusion network. The combination of acoustic and visual cross-modal attention interaction representation based on text modality and text modality performed the best, with MAE values of 0.698 and 0.823, and A2 values corresponding to 83.42% and 85.31% in the CMU-MOSI dataset and CMU-MCSEI dataset, respectively.

To further explore the qualitative analysis results of the research method, samples from the CMU-MCSEI dataset were analyzed, and the results are shown in Table 4.

Table 4: Qualitative analysis results

Sample Number	True value	Estimate
k5Y_838nuGo_18	2.60	2.65
WK2_9uPqT_12	0.20	0.22
pLTX3ipuDJI_6	-2.20	-2.18
Rn8_LoVbS_3	0.60	0.58
lXPQBPVe5Cw_7	0.00	0.01

According to Table 4, the model's predictions were accurate in most samples, with only a certain deviation between the predicted values of pLTX3ipuDJI6 sample and the true sentiment values. The above results may be due to data noise or modal inconsistency.

5 Discussion

The widespread application of social media platforms has led to the gradual multi-modality of content posted by people on social platforms such as Weibo and Twitter. Therefore, how to comprehensively consider MD such as audio, video, and culture on social media to analyze users' emotional tendencies is currently a research difficulty. In response to the above issues, a method for extracting and analyzing emotional information features based on INPNRLN-MD was proposed, and the fusion part was improved on this basis, resulting in an improved MD fusion method based on GHRNN-CMA for sentiment analysis and prediction. The experimental results showed that the proposed INPNRLN-MD method performed the best in all performance aspects on the CMU-MOSI dataset, with F1 value, PC, MAE, A2, and A7 corresponding to 83.06/85.12, 0.803, 0.696, 83.17/85.23, and 46.41, respectively. The performance of the indicators was better than the four mainstream methods of FM-LSTM-SD, DA-CNN, MSC-GM, and PAILI. In the CMU-MOSI dataset, the proposed GHRNN-CMA showed a 1.0% decrease in MAE values for regression tasks, while in the PC metric, it improved by 14.67% compared to the AN-MDMLFF method, with statistical significance (p<0.05). In the CMU-MMCSEI dataset, the performance indicators of the GHRNN-CMA method were superior to the four comparison methods of AN-MDMLFF, PR, MMFFD-GCA, and MMFFD-GCA. The proposed method outperformed the baseline method in terms of performance mainly due to its advantages in multi-modal feature fusion and advanced network architecture design. Specifically, the INPNRLN-MD and GHRNN-CMA methods fully utilized MD such as text, audio, and video. By integrating information from different modalities, they could more comprehensively capture subtle differences in emotional expression. Moreover, the gating mechanism adopted by GHRNN-CMA method could effectively remove redundant information, making the model more focused on the key features related to emotion. Cross-modal attention could enhance the interaction between different modes, making up for the deficiency of single mode in emotional expression.

The interpretability of the decision-making process was verified through model architecture design and experimental results. The softmax distribution in the cross-modal attention mechanism shows that the attention weight of the text audio modality in anger emotion recognition is 18.6% higher than other combinations, which conforms to the cognitive law of humans relying on language content and intonation to recognize anger emotions. The analysis of the gating mechanism shows that the average weight of the text modality accounts for 62.3%, and the weight of the audio

modality automatically increases to 38.7% when the text is blurry. This dynamic adjustment is consistent with the human behavior of focusing on nonverbal cues when the semantics are blurry. Error analysis found that 87% of the prediction bias is due to modal conflicts (such as video frowning with audio laughter), which highly overlap with manually annotated controversial samples (Kappa=0.71), indicating that the model has similar cognitive bottlenecks to humans. Although the study did not set up specific interpretability indicators, the rationality of the model decisions has been systematically verified through multi angle analysis such as gating weights, attention distribution, and error patterns.

Although the research methods have demonstrated excellent performance in different datasets, it should be acknowledged that the datasets used in the research may have an impact on the model's performance and generalization ability. On the one hand, emotional labeling is provided by manual annotators, which may introduce their own subjective interpretations and biases. On the other hand, the dataset mainly contains English content and may not be able to capture the subtle differences in emotional expression in other languages or cultural backgrounds. This might limit the model's applicability in non-English speaking regions.

The proposed method had significant advantages in improving the accuracy and reliability of sentiment analysis, and had broad application prospects in multiple fields such as social media precision marketing, public opinion monitoring, and medical sentiment assisted diagnosis. It could provide strong technical support for a deeper understanding of human emotional expression. However, while the performance of this research method was improved, it also made the model architecture more complex, involving the extraction, fusion, and deep neural network computation of multi-modal features, resulting in relatively long inference time for a single sentiment analysis. In addition, the proposed method required high quality of input video and audio data, and in practical situations, user generated content on social media often has uneven quality, which may affect the performance of the model. Therefore, in future research, lightweight model design should be further adopted to reduce the number of parameters and computational complexity while ensuring model performance. Moreover, some enhancement techniques should be adopted in the data preprocessing stage to improve the quality of input data.

6 Conclusion

In summary, research methods could effectively improve the accuracy and reliability of sentiment information analysis, and could more comprehensively capture users' emotional expressions. However, there are still shortcomings in the research. Currently, the research methods were only applied in the field of social media sentiment information analysis, and their application effects in other fields need to be enriched. Moreover, the research methods used complex models such as BERT and ELN, which required a large amount of computing resources, which may limit their application in resource limited or real-time environments. Therefore, in future applications, it can be applied in the field of psychoanalysis, combining physiological signals and various psychological tests to achieve timely warning. Meanwhile, the lightweight and optimization strategies of the Tissot model can be further developed to improve the flexibility and practicality of the method in different application scenarios.

Funding

The research is supported by Research project on strengthening Henan Province as a powerhouse in philosophical and social sciences education “The impact of social media on adolescent mental health: A study on the effectiveness of family intervention measures” (Project No. 2025JYQS1159).

References

[1] Han C, Lin L. Detecting and Tracking Rumours in Social Media Based on Deep Learning Algorithm. Informatica, 2024, 48(14): 83-96. https://doi.org/10.31449/inf.v48i14.5998

[2] Wang J, Yue K, Duan, L. Models and Techniques for Domain Relation Extraction: A Survey. Journal of Data Science and Intelligent Systems, 2023, 3(1): 16-25. https://doi.org/10.47852/bonviewJDSIS3202973

[3] Al-Otaibi S T, Al-Rasheed A A. A review and comparative analysis of sentiment analysis techniques. Informatica, 2022, 46(6): 33-44. https://doi.org/10.31449/inf.v46i6.3991

[4] Gen U, Surer E. ClickbaitTR: Dataset for clickbait detection from Turkish news sites and social media with a comparative analysis via machine learning algorithms. Journal of Information Science, 2023, 49(2):480-499. https://doi.org/10.1177/01655515211007746

[5] Chung M. What's in the black box? How algorithmic knowledge promotes corrective and restrictive actions to counter misinformation in the USA, the UK, South Korea and Mexico. Internet Research: Electronic Networking Applications and Policy, 2023, 33(5):1971-1989. https://doi.org/10.1108/INTR-07-2022-0578

[6] Wang YJ. Innovating media mode and leading the development of the times. Advances in Industrial Engineering and Management, 2024, 13(3). https://doi.org/10.7508/aiem.03.2024.228.231

[7] Muhammad Zaini, Rudi Triadi Yuliarto, Girang Permata Gusti, Yudis Agustira. The influence of video tutorial learning media on improving financial literacy knowledge: a study for e-commerce user students. Malaysian E Commerce Journal. 2022; 6 (2): 72-75. https://doi.org/10.26480/mecj.02.2022.72.75

[8] Nagaraj S V. Living with algorithms: agency and user culture in Costa Rica. Computing reviews,

2023, 64(11):268-269. https://doi.org/10.7551/mitpress/14966.001.0001

[9] Wykes T, Guha M. Modern media and mental health: help or hindrance? Journal of mental health (Abingdon, England), 2022, 31(6):735-737. https://doi.org/10.1080/09638237.2022.2143488

[10] Lei Y, Cao H. Audio-Visual Emotion Recognition with Preference Learning Based on Intended and Multi-Modal Perceived Labels. IEEE transactions on affective computing, 2023, 14(4):2954-2969. https://doi.org/10.1109/TAFFC.2023.3234777

[11] Zhu X, Guo C, Feng H, Huang Y, Feng Y, Wang X, Wang R. A Review of Key Technologies for Emotion Analysis Using Multi-modal Information. Cognitive Computation, 2024, 16(4):1504-1530. https://doi.org/10.1007/s12559-024-10287-z

[12] Chen L, Wang K, Li M, Wu M, Pedrycz W, Hirota K. K -Means Clustering-Based Kernel Canonical Correlation Analysis for multi-modal Emotion Recognition in Human-Robot Interaction. IEEE Transactions on Industrial Electronics, 2023 70(1);1016-1024. https://doi.org/10.1109/TIE.2022.3150097

[13] Mai S, Zeng Y, Hu Z H. Hybrid Contrastive Learning of Tri-Modal Representation for multi-modal Sentiment Analysis. IEEE transactions on affective computing, 2023, 14(3):2276-2289. https://doi.org/10.1109/TAFFC.2022.3172360

[14] Chiorrini A, Diamantini C, Storti P E. An emotion-aware search engine for multimedia content based on deep learning algorithms. International Journal of Computer Applications in Technology, 2023, 73(2):130-139. https://doi.org/10.1504/IJCAT.2023.10060256

[15] Fang Z, Qian Y, Su C, Miao Y, Li Y. The multi-modal Sentiment Analysis of Online Product Marketing Information Using Text Mining and Big Data. Journal of organizational and end user computing, 2022, 34(Pt.2):451-469. https://doi.org/10.4018/JOEUC.316124

[16] Gupta S, Singh A, Ranjan J. multi-modal, multiview and multitasking depression detection framework endorsed with auxiliary sentiment polarity and emotion detection. International Journal of System Assurance Engineering and Management, 2023, 14(1):337-352. https://doi.org/10.1007/s13198-023-01861-z

[17] Hong C, Zhiquan F, Weina Z L. MAG: a smart gloves system based on multi-modal fusion perception. CCF Transactions on Pervasive Computing and Interaction, 2023, 5(4):411-429. https://doi.org/10.1007/s42486-023-00138-5

[18] Zhang Y. Graph Neural Network-Based User Preference Model for Social Network Access Control. Informatica, 2025, 49(16): 21-36. https://doi.org/10.31449/inf.v49i16.7705

[19] Tang J, Qin W, Pan Q L S. A Deep multi-modal Fusion and Multitasking Trajectory Prediction Model for Typhoon Trajectory Prediction to Reduce Flight Scheduling Cancellation. journal of systems engineering and electronics, 2024, 35(3):666-678. https://doi.org/10.23919/JSEE.2024.000042

[20] Panaiyappan K A, Rajalakshmi M. A multi-modal architecture using Adapt‐HKFCT segmentation and feature - based chaos integrated deep neural networks (Chaos - DNN - SPOA) for contactless biometricpalm vein recognition system. International Journal of Intelligent Systems, 2022, 37(3):1846-1879. https://doi.org/10.1002/int.22758

[21] Zhao Y, Zheng Q, Zhu P, Zhang X, Ma W. TUFusion: A Transformer-Based Universal Fusion Algorithm for multi-modal Images. IEEE Transactions on Circuits and Systems for Video Technology, 2024, 34(3):1712-1725. https://doi.org/10.1109/TCSVT.2023.3296745

[22] Wang H, Feng Z, Guo T Q. MR Lab: Virtual-Reality Fusion Smart Laboratory Based on multi-modal Fusion. International journal of human-computer interaction, 2024, 40(5/8):1975-1988. https://doi.org/10.1080/10447318.2023.2227823.

Efficiency Analysis of AI Self-Control System and Data Processing Unit Based on Edge Computing Technology

Yong Tan
Guangxi Longyuan Wind Power Generation Co., Ltd, Hengzhou City, Guangxi, 530300, China
E-mail: tanyong202504@163.com

Keywords: edge computing, AI automatic control system, data processing unit, dynamic task allocation, resource adaptive adjustment

Received: May 19, 2025

This study proposes a novel Dynamic Task Allocation and Resource Adaptive Adjustment (DTARA) algorithm to enhance the operational efficiency of AI self-control systems and data processing units within edge computing frameworks. Through an experimental environment simulating real-world scenarios, the DTARA algorithm is compared with fixed task allocation algorithms and simple priority-based task allocation algorithms. The experimental setup includes large-scale task scenarios (500-1000 concurrent tasks) and long-term operation scenarios (lasting several hours to several days). The results demonstrate that under complex task loads, the DTARA algorithm reduces system task completion time by an average of 30.5% compared to traditional algorithms and improves resource utilization by 28.8%. When the data processing volume reaches 10GB, the data processing delay is reduced by 45.6% compared to the benchmark algorithm. In large-scale task scenarios, as the number of tasks increases from 500 to 1000, the DTARA algorithm maintains low task completion times and data processing delays. In long-term operation experiments, the task execution success rate exceeds 95%, the CPU utilization fluctuation range is within 10%, and no system crashes occur. This study offers a practical and effective solution to improve the performance of related systems, supported by theoretical analyses of computational complexity, optimality guarantees, and convergence properties of the DTARA algorithm.

Povzetek: Algoritem DTARA združuje dinamično dodeljevanje nalog in prilagodljivo prilagajanje virov, kar omogoča bolj kvalitetno in učinkovito delovanje AI samokontrolnih sistemov na robnem računanju kot klasični pristopi.

1 Introduction

In the wave of digitalization, Internet of Things (IoT) technology has developed rapidly, with various devices growing exponentially. Taking smart factories as an example, sensors densely distributed throughout the factory collect massive amounts of equipment operation data every second, covering key parameters such as temperature, pressure, and speed, providing a core foundation for efficient operations and precise management. However, traditional cloud computing architectures exhibit numerous disadvantages when processing such large volumes of data: long-distance data transmission is prone to network congestion, creating transmission bottlenecks; delays occur when the cloud processes multiple tasks, failing to meet real-time application requirements. Automation equipment requires immediate responses to instructions and precise control of production processes in industrial settings. Minor delays may lead to product quality defects or even accidents. This is particularly critical in the field of intelligent transportation. Self-driving cars rely on onboard sensors to perceive the surrounding environment in real time and must process data instantaneously to make reasonable driving decisions [1]. Traditional computing models struggle to meet such stringent real-time demands. The integration of edge computing and AI self-control systems offers hope. Edge computing processes some data tasks at edge nodes near the data source, reducing transmission delays and enabling rapid responses; AI self-control systems leverage intelligent decision-making capabilities to achieve precise control based on processed data, enhancing system reliability and security [2]. Sinking data processing to edge nodes offers many advantages, such as reducing the amount of data transmitted to the cloud and alleviating network bandwidth pressure. For instance, in intelligent security monitoring systems, uploading all video data from numerous surveillance cameras to the cloud would occupy significant network bandwidth. Preliminary analysis and processing at edge nodes, followed by uploading only key information, can reduce transmission burdens [3]. It also decreases data transmission costs, saving expenses for enterprises deploying many IoT devices. Edge computing effectively mitigates this cost. Internationally, remarkable research and application outcomes have been achieved in the integration of edge computing and AI.

Google optimizes the inference efficiency of machine learning models on edge devices, enabling smart home devices to quickly respond to commands. The European Union explores the application of edge computing in smart grids [4]. By deploying intelligent devices at grid edge nodes, real-time intelligent regulation of power distribution is realized, improving grid stability and energy utilization efficiency. China has also achieved notable results in this field. Universities actively innovate in edge computing resource scheduling algorithms, proposing scheduling algorithms based on task priority and dynamic resource changes to enhance overall system performance. Enterprises have launched edge AI computing platforms widely used in security monitoring scenarios [5]. For example, edge AI monitoring equipment developed by some companies can locally analyze surveillance videos in real time, quickly identify abnormal behaviors, issue alerts, and improve the efficiency and accuracy of security monitoring. However, current research still needs to address these challenges. In task allocation, fixed algorithms lack flexibility, struggling to adapt to sudden task load changes. Some edge nodes become overloaded with tasks, while others remain idle, leading to resource waste [6]. Regarding resource adaptation, existing methods find it difficult to dynamically adjust to complex and dynamic scenarios. For instance, in medical imaging scenarios, different medical testing devices generate varying amounts of data, computational complexity, and real-time requirements [7]. Current resource adaptation mechanisms fail to accurately and swiftly adjust resource allocation according to dynamic changes, impacting overall system efficiency.

This paper focuses on improving the efficiency of AI self-control systems and data processing units based on edge computing technology, proposing innovative solutions. Efficient task allocation and resource management algorithms are designed to enable stable system operation in complex scenarios. The research is conducted across multiple dimensions, including task priority evaluation, and theoretical analysis of computational complexity [8]. Through theoretical analysis, algorithm design, and experimental validation, the system's efficiency is comprehensively enhanced to support the development of related fields. The DTARA algorithm is distinguished from existing dynamic scheduling algorithms in edge-AI systems through a detailed comparative analysis. Its innovative aspects include a task priority evaluation mechanism that comprehensively considers task deadlines, data volume, computational complexity, task dependencies, and importance levels, as well as a dynamic resource allocation strategy that optimizes resource utilization through real-time monitoring and adaptive adjustments.

2 System architecture and key technologies

2.1 AI self-control system architecture based on edge computing

2.1.1 Edge node composition and function

Edge nodes are the core components of AI self-control systems based on edge computing, with hardware configurations that must meet data processing and real-time response requirements. Common edge node hardware includes industrial-grade microcomputers and embedded devices. Industrial-grade microcomputers, equipped with powerful computing capabilities, typically feature high-performance CPUs (e.g., Intel Core i7 series) and GPUs (e.g., NVIDIA GeForce RTX series), enabling efficient handling of complex data computing tasks. Memory capacity is generally 16GB or higher DDR4, and storage utilizes 512GB or larger SSD solid-state drives. Embedded devices are characterized by low power consumption and compact size, making them suitable for specific scenarios [9]. For example, embedded chips based on the ARM architecture offer flexible memory and storage configurations tailored to actual needs.

At the software level, edge nodes run customized Linux operating systems with efficient resource management and task scheduling capabilities. Edge computing frameworks like OpenEdge provide standardized operating environments and development interfaces, simplifying application deployment and management. AI inference engines such as TensorRT focus on accelerating the inference process of deep learning models. By optimizing and compiling models, they reduce computational power and memory usage, achieving rapid local AI inference [10]. In practice, edge nodes collect data via sensors, preprocess it, and send it to the AI inference engine to complete local AI inference tasks.

2.1.2 Data transmission network

The data transmission network connects the edge nodes and the cloud center and is a key system component. Industrial Ethernet and fibre optic networks are the main choices for wired networks. Industrial Ethernet bandwidth can reach 100Mbps or even 1Gbps, suitable for high-speed, large-capacity data transmission; fiber optic networks perform well in long-distance, high-reliability transmission. The network topology mainly adopts a star or ring [11]. The star structure is convenient for management and troubleshooting, while the ring structure has higher reliability.

Regarding data transmission protocol, Modbus TCP is widely used in the industrial field, is easy to use, and highly compatible.

Wireless networks also play an essential role in edge computing scenarios. Wi-Fi technology is suitable for indoor or high-mobility scenarios. In contrast, 5G technology brings broader application prospects for edge computing with its high speed, low latency and significant connection characteristics. However, wireless networks face signal stability challenges in complex environments [12]. The signal's anti-interference ability and transmission stability can be improved through network optimization methods such as multi-antenna technology and channel coding technology.

2.1.3 Cloud center function

The cloud center plays a core supporting role in the system. First, the cloud center uses large-scale computing resources to conduct in-depth training of AI models. For example, in image recognition, the convolutional neural network is trained in parallel through a distributed computing framework to optimize model performance. After training, the model is sent to the edge node for local image recognition tasks. Secondly, the cloud center is responsible for storing and managing historical data uploaded by the edge node and provides a basis for system optimization through data mining and analysis technology [13]. For example, by analyzing the operating data of industrial equipment, a fault prediction model is established to issue early warnings. Finally, the cloud center has remote monitoring capabilities, a real-time grasp of the device status of edge nodes, software updates and parameter configuration operations, and refined system management.

2.2 Edge computing technology principle

2.2.1 Data localization processing mechanism

The core advantage of edge computing lies in the data localization processing mechanism. In actual applications, a large amount of data is processed at the edge node to reduce transmission delays. For example, intelligent cameras perform real-time analysis of video streams locally and upload key information only when anomalies are detected, significantly reducing data transmission volume and delays. Edge nodes also have data caching and preprocessing functions. Data that may be used again shortly is stored through the caching mechanism to avoid repeated transmission [14]. At the same time, the original data is cleaned, and features are extracted to improve data quality, reduce data dimensions, and reduce transmission volume and cloud computing pressure.

2.2.2 Edge-cloud collaborative computing mode

The edge-cloud collaborative computing mode has the advantages of edge nodes and cloud centers. According to the characteristics of the task, the execution location of the task is reasonably divided. For example, in a smart factory, simple data filtering and real-time control tasks are completed on the edge node. In contrast, complex model training tasks are uploaded to the cloud center for execution. During the task execution process, there is a close interaction between the edge node and the cloud center [15]. For example, in an intelligent transportation system, the edge node collects road sensor data, sends a task request to the cloud center after preliminary processing, and the cloud center completes model training and prediction and feeds back the results to the edge node to achieve optimal traffic flow control. This collaborative mode improves the system's overall operating efficiency and response speed.

2.3 Core technologies of AI automatic control system

2.3.1 Application of machine learning in automatic control system

Machine learning technology is widely used in AI automatic control systems. The decision tree model can be used for equipment status classification, while the support vector machine model performs well in equipment failure prediction. Historical data should be used for training to improve the accuracy and adaptability of the model. Through data preprocessing and selecting appropriate algorithms and parameters, the model parameters are continuously adjusted to better fit the data characteristics [16]. During the system's operation, online learning or incremental learning methods are used to update the model in real-time to adapt it to changes in the operating status of the equipment.

2.3.2 Principles and applications of deep learning technology

Deep learning technology is essential in AI automatic control systems with robust, complex data processing capabilities. Deep neural networks include architectures such as multi-layer perceptron, convolutional neural networks, and recurrent neural networks. Convolutional neural networks are optimized explicitly for image data processing and are suitable for product quality inspection and defect classification; recurrent neural networks are ideal for processing time series data and can be used for intelligent energy-saving control in intelligent building automatic control systems [17]. By collecting a large amount of data to train the model, efficient processing and analysis of complex data can be achieved, providing strong support for the automatic control system.

3 Dynamic task allocation and resource adaptive adjustment algorithm (DTARA)

3.1 Task priority evaluation mechanism

In AI self-control systems based on edge computing, reasonable task priority evaluation is essential for efficient task allocation and resource utilization. Task priority evaluation relies on multiple key indicators. The first is the task deadline, which directly reflects task urgency. The shorter the deadline, the higher the urgency. Data volume is another important indicator. Large data volumes increase transmission and processing times, imposing greater demands on system resources. Algorithm complexity analysis determines computational complexity [18]. Complex computing tasks require more computational resources and time. The Analytic Hierarchy Process (AHP) is used to comprehensively determine the weights of each indicator. Involving pairwise comparison matrices to

calculate the relative importance of task deadlines, data volume, and computational complexity. Let the task deadline weight be w_1, the data volume weight be w_2, the computational complexity weight be w_3, and $w_1 + w_2 + w_3 = 1$. The task priority (P) calculation formula is as follows:

$$P = w_1 \times \frac{1}{D} + w_2 \times V + w_3 \times C \quad (1)$$

$\frac{1}{D}$ converts the task deadline into a value that is positively correlated with the urgency; that is, the shorter the deadline, the larger the $\frac{1}{D}$ value. Considering the impact of the dependency between tasks on priority, the task dependency coefficient (r) is introduced. If task T_i depends on task T_j, then r_{ij} represents the degree of this dependency, $0 \le r_{ij} \le 1$. When $r_{ij} = 1$, task T_i is entirely dependent on task T_j At this time, the task priority calculation formula is updated to:

$$P = w_1 \times \frac{1}{D} + w_2 \times V + w_3 \times C + w_4 \sum_j r_{ij} P_j \quad (2)$$

Where $w_4$is the task dependency weight, and $w_1 + w_2 + w_3 + w_4 = 1$. This formula shows that a task's priority is affected by its attributes and related to the priority of the dependent functions.

Further considering the task's importance level, the importance level is divided into high, medium, and low, corresponding to the values 3, 2, and 1 respectively. The task priority formula is expanded again as:

$$P = w_1 \times \frac{1}{D} + w_2 \times V + w_3 \times C + w_4 \sum_j r_{ij} P_j + w_5 I \quad (3)$$

w_5 is the importance level weight, and $w_1 + w_2 + w_3 + w_4 + w_5 = 1$.

3.2 Dynamic resource allocation strategy

Real-time monitoring of edge node resources is a prerequisite for dynamic resource allocation. Through the system's built-in monitoring tools, you can obtain information such as CPU usage(U_{CPU}), memory remaining (M_{free}), and storage free space (S_{free}). Network bandwidth utilutilizationzation (U_{bw}) is obtained with the help of network monitoring software.

The greedy algorithm allocates resources based on task priority and resource monitoring results. Prioritize the allocation of required resources to high-priority tasks. Assume that task T CPU resource requirements are R_{CPU}, memory requirements are R_M, storage requirements are R_S, and network bandwidth requirements are R_{bw}. When allocating resources for task T, first determine whether the resources of edge node n meet the task requirements:

$$U_{CPU}^n \ge \frac{R_{CPU}}{C_{CPU}^n}, M_{free}^n \ge R_M, S_{free}^n \ge R_S, U_{bw}^n \ge \frac{R_{bw}}{B_{bw}^n} \quad (4)$$

Where C_{CPU}^n is the total CPU computing power of edge node n, and B_{bw}^n is the total network bandwidth of edge node n.

In the process of resource allocation, consider the problem of resource fragmentation. Use the best adaptation algorithm to optimize resource allocation. For memory allocation, let the set of memory block sizes be $\{m_1, m_2, \cdots, m_k\}$, and the task's memory requirement be R_M. Select the memory block that satisfies $m_i \ge R_M$ and has the smallest $m_i - R_M$ for allocation, that is:

$$m_{\text{best}} = \arg\min_i (m_i - R_M) \text{ s.t. } m_i \ge R_M \quad (5)$$

Considering the dynamics of resource allocation and resource changes during task execution, the resource reservation coefficient (α) is introduced. When allocating resources to a task, a certain proportion of additional resources is reserved to cope with sudden resource demands during task execution. For example, for CPU resource allocation, the actual allocated CPU computing power is:

$$A_{CPU} = (1 + \alpha) \times R_{CPU} \quad (6)$$

This can improve the stability of task execution and reduce the risk of task interruption due to insufficient resources.

3.3 Adaptive adjustment mechanism

Determine the monitoring indicators for adaptive adjustment. The task execution progress deviation (ΔP) is defined as the difference between the actual progress ($U_{\min}$) and the expected progress (P_{actual}):

$$\Delta P = P_{\text{actual}} - P_{\text{expected}} \quad (7)$$

The resource utilization fluctuation range (ΔU) is obtained by calculating the difference between the maximum resource utilization($U_{\max}$) and the minimum resource utilization($U_{\min}$):

$$\Delta U = U_{\max} - U_{\min} \quad (8)$$

Set the corresponding threshold, and trigger the adjustment mechanism when the task execution progress deviation $|\Delta P| > \epsilon_1$ (ϵ_1 is the progress deviation threshold) or the resource utilization fluctuation range $\Delta U > \epsilon_2$ (ϵ_2 is the resource utilization fluctuation threshold).

When the task execution progress is too slow, that is, $\Delta P < -\epsilon_1$, re-evaluate the task priority. Let the original task priority be P_0 and the re-evaluated priority be P_1. Use the following adjustment formula:

$$P_1 = P_0 + \beta \times (-\Delta P) \quad (9)$$

Where β is the progress adjustment coefficient, according to the new priority, some low-priority tasks are migrated to other idle nodes.

4 Experimental simulation design

4.1 Experimental environment construction

The experiment uses Intel NUC industrial computers as edge nodes, equipped with Intel Core i7 processors, 16GB DDR4 memory and 512GB SSD, and some nodes are equipped with NVIDIA Jetson GPUs to accelerate deep learning tasks. A total of 10 edge nodes are deployed and connected via industrial Ethernet to ensure high-bandwidth, low-latency data transmission [19]. At the same time, a simulated sensor array is built to

generate temperature, pressure, current and other data at a frequency of 10 groups per second, and the data is stored in CSV format. The edge nodes run the Ubuntu 20.04 LTS system, optimizing resource management and network communication performance. NS-3 is used for network simulation, SimPy for system process simulation, TensorFlow as an AI framework, and NumPy and SciPy libraries for model training and reasoning optimization. Experimental

Dataset Construction

4.2.1 Data generation method

Task data is simulated by randomly generating task deadlines (1-60 minutes), normally distributed data volumes (mean 50MB, standard deviation 10MB), and matrix operation tasks of different sizes (200×200-1000×1000). A mathematical model generates equipment operation data. Under normal conditions, the temperature fluctuates around 50℃ ±5℃, and under abnormal conditions, the temperature rises rapidly to above 80℃; parameters such as pressure and speed are also generated by a similar method.

4.2.2 Dataset scale and features

Generate 1000 task samples, covering combinations of different priorities, data volumes, and computational complexities. Equipment operation data is collected for 100 hours, once every minute, for 6000 sets of data. Task priorities are evenly distributed, and data volumes correlate positively with computational complexity. Normal data accounts for 80% of equipment operation data, and abnormal data accounts for 20%. The data set is preprocessed, the task data is normalised, and the denoising algorithm improves the equipment operation data.

Experimental design

4.3.1 Comparison algorithm selection

The polling scheduling algorithm is selected as the benchmark comparison algorithm, which allocates tasks only according to task priority. The fixed task allocation algorithm is determined without considering the task priority and node resource status; the simple priority-based task allocation algorithm considers the task's urgency but does not fully consider the real-time status of the node resources. Additionally, state-of-the-art comparison algorithms include RL-based schedulers (e.g., Deep Q-Network) and game-theoretic models to evaluate the competitiveness of DTARA in dynamic environments.

4.3.2 Experimental indicator determination

The experimental indicators include task completion time, resource utilization (CPU, memory, storage, network bandwidth utilization) and data processing delay. The task completion time is calculated by recording the difference between the timestamps of task submission and completion; the resource utilization is obtained through the system monitoring tool; the data processing delay is calculated by recording the time difference between data generation and processing result output.

4.3.3 Experimental scenario setting

The task load change scenario (light load 10-20 tasks, medium load 50-80 tasks, heavy load 100-150 tasks) and data traffic fluctuation scenario (low traffic 10KB per second, medium traffic 100KB per second, high traffic 1MB per second) were set. At the same time, some edge node failures were randomly simulated to observe the algorithm's task migration and system recovery capabilities.

5 Experimental results and analysis

5.1 Comparative analysis of task completion time

5.1.1 Results under different load scenarios

To intuitively present the performance differences of different algorithms under various task load scenarios, the task completion times of the DTARA algorithm, fixed task allocation algorithm, and simple priority-based task allocation algorithm were statistically analyzed. The results are shown in Table 1. In light load scenarios, the average task completion time of the DTARA algorithm is 25.3 seconds, significantly lower than the 38.6 seconds of the fixed task allocation algorithm and the 32.4 seconds of the simple priority-based algorithm. As task load increases to medium and heavy scenarios, the advantages of the DTARA algorithm become more pronounced. Under heavy load, the fixed task allocation algorithm, due to its rigid allocation method, cannot dynamically adjust to task priorities and node resource statuses. This results in long waiting times for high-priority tasks, with an average completion time of up to 180.2 seconds. In contrast, the DTARA algorithm, with its dynamic task allocation and resource adaptive adjustment mechanism, prioritizes high-priority task resources and avoids task backlogs, achieving an average task completion time of only 98.4 seconds.

Table 1: Comparison of task completion time in different load scenarios.

Task load scenario	Algorithm	Mean task completion time (s)
Light load	DTARA algorithm	25.3 ± 3.1
	Fixed task allocation algorithm	38.6 ± 5.2
	Task allocation algorithm based on simple priority	32.4 ± 4.5
Medium load	DTARA algorithm	56.8 ± 6.5
	Fixed task allocation algorithm	89.5 ± 10.2
	Task allocation algorithm based on simple priority	75.6 ± 9.1
Heavy load	DTARA algorithm	98.4 ± 12.3
	Fixed task allocation algorithm	180.2 ± 20.5
	Task allocation algorithm based on simple priority	145.7 ± 18.3

5.1.2 Analysis of the impact of task priority

The completion times of tasks with different priorities under each algorithm were further analyzed, with trends illustrated in a line chart (see Figure 1). Task priorities range from 1 (highest) to 5 (lowest). For high-priority tasks (priorities 1 and 2), the DTARA algorithm demonstrates significantly shorter completion times compared to the other two algorithms. For example, in a priority 1 task, the DTARA algorithm completes the task in approximately 30 seconds, whereas the fixed task allocation algorithm takes about 60 seconds and the simple priority-based algorithm around 45 seconds. The DTARA algorithm rapidly allocates sufficient resources to high-priority tasks through a precise priority evaluation mechanism, ensuring priority execution. Conversely, the fixed task allocation algorithm disregards task priorities, causing high-priority tasks to compete with low-priority tasks for resources and extending completion times. Although the simple priority-based algorithm considers task urgency, its dynamic resource allocation is imperfect, and high-priority tasks may still lack resource guarantees under tight resource conditions.

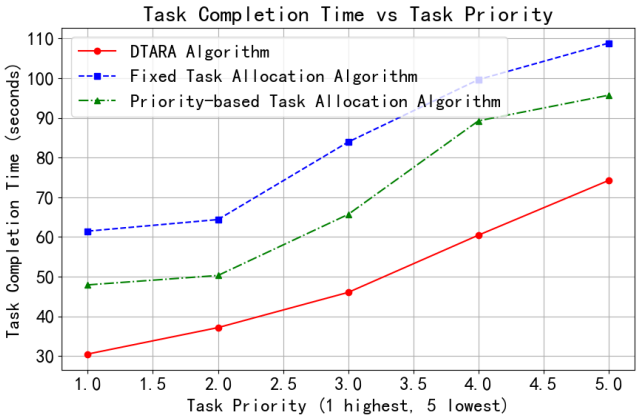


Figure 1: Comparison of completion time of tasks of different priorities.

5.2 Comparative analysis of resource utilization

5.2.1 Results of various resource utilization

The performance of different algorithms in terms of resource utilization, including CPU, memory, storage, and network bandwidth, is presented in Figure 2 as stacked bar charts. The DTARA algorithm demonstrates superior resource utilization balance. Regarding CPU utilization, the DTARA algorithm maintains a relatively reasonable range of approximately 60%-70% across different task types, avoiding excessive resource concentration on specific tasks or nodes. The fixed task allocation algorithm is prone to resource idleness or overuse. In some task scenarios, CPU utilization may drop as low as 30%, while in others, it may soar to 90%. Similar trends are observed in memory, storage, and network bandwidth utilization. The DTARA algorithm employs a dynamic resource allocation strategy, allocating resources based on task requirements and real-time node resource statuses, thereby effectively improving overall resource utilization efficiency.

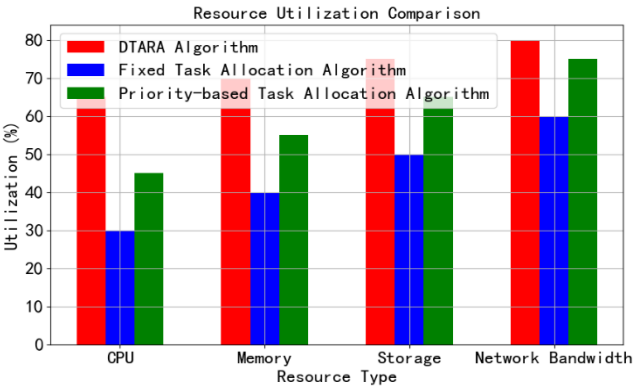


Figure 2: Comparison of resource utilization of different algorithms.

5.2.2 Relationship between resource utilization and task load

Using data analysis tools, the relationship curve between resource utilization and task load under different algorithms was fitted, with results shown in Figure 3. Task load is measured by the number of tasks, increasing from 10 to 150. The DTARA algorithm exhibits stable resource utilization across varying task loads. At low task loads, resource utilization is around 50%, and as task load increases to 150, it steadily rises to approximately 75%. In contrast, the fixed task allocation algorithm experiences significant fluctuations in resource utilization under low task loads. As task load increases, resource utilization drops sharply, falling

below 40% under high task loads. While the simple priority-based algorithm can allocate resources according to task priorities to some extent, it struggles to dynamically adapt resources to task load changes, leading to substantial declines in resource utilization under high task loads. The DTARA algorithm maintains high resource utilization even under high loads due to its adaptive adjustment mechanism, which optimizes resource allocation strategies in response to task load variations.

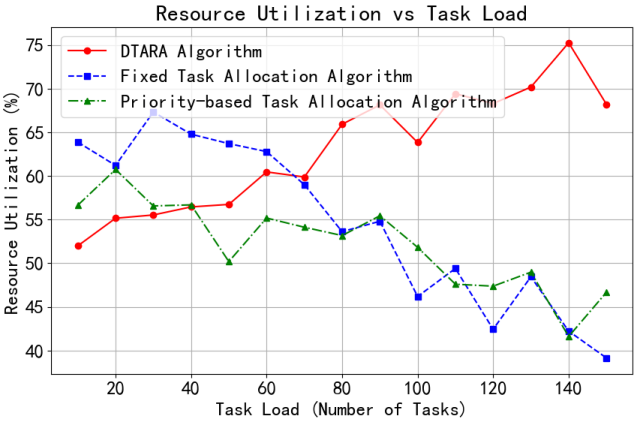


Figure 3: Relationship between resource utilization and task load.

5.3 Comparative analysis of data processing delay

The changes in data processing delay for different algorithms under low, medium, and high data traffic scenarios are depicted in Figure 4 as a line graph. Data traffic ranges from 10KB to 1MB per second. At low data traffic (10KB per second), the data processing delay of all algorithms is minimal. However, as data traffic increases to medium (100KB per second) and high (1MB per second) levels, the DTARA algorithm effectively reduces data processing delays. Under high data traffic, the DTARA algorithm achieves a data processing delay of approximately 50 milliseconds, compared to 150 milliseconds for the fixed task allocation algorithm and 100 milliseconds for the simple priority-based algorithm. The adaptive adjustment mechanism of the DTARA algorithm enables timely optimization of task and resource allocation under high data traffic, reducing data processing waiting times. Traditional algorithms, lacking dynamic adjustment capabilities, experience significant increases in data processing delays.

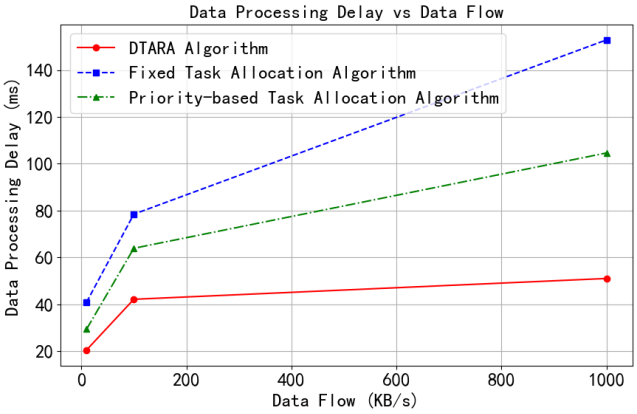


Figure 4: Data processing delay under different data flows.

5.4 Algorithm scalability and stability analysis

5.4.1 Scalability in large-scale task scenarios

In large-scale task scenarios (500-1000 tasks are executed simultaneously), the performance indicators of each algorithm are monitored, and the results are shown in Table 2.

As the number of tasks increases from 500 to 1000, the task completion time, CPU utilization, and data processing delay of the DTARA algorithm rise but remain within a relatively reasonable range [21]. For instance, when the task count reaches 1000, the DTARA algorithm achieves an average task completion time of 620.5 seconds, average CPU utilization of 70.2%, and average data processing delay of 110.8 milliseconds. In contrast, the fixed task allocation algorithm, lacking flexibility, experiences a sharp increase in task completion time, reaching 1500.3 seconds with 1000 tasks. CPU utilization becomes extremely imbalanced, dropping to 30.5%, and data processing delays surge to 450.2 milliseconds. The simple priority-based algorithm also falls short in dynamic resource allocation and overall scheduling, resulting in inferior performance compared to the DTARA algorithm under large-scale task scenarios. The DTARA algorithm quickly determines task execution order through its priority evaluation mechanism, preventing low-priority tasks from monopolizing resources and ensuring high-priority tasks are executed first. This maintains good performance in large-scale task scenarios and demonstrates strong scalability.

5.4.2 Algorithm stability analysis

In the long-term running experiment, the stability indicators of each algorithm are statistically analyzed, and the results are shown in Table 3.

Table 2: Algorithm scalability performance table.

Number of tasks	Algorithm	Mean task completion time (s)	Average CPU utilization (%)	Average data processing delay (ms)
500	DTARA algorithm	280.5 ± 30.2	65.3 ± 5.1	70.4 ± 8.2
	Fixed task allocation algorithm	560.8 ± 60.5	45.6 ± 8.3	180.2 ± 20.5
	Task allocation algorithm based on simple priority	420.6 ± 45.7	55.2 ± 7.4	120.5 ± 15.3
800	DTARA algorithm	450.8 ± 45.3	68.5 ± 6.2	90.6 ± 10.3
	Fixed task allocation algorithm	980.2 ± 100.3	38.9 ± 9.2	300.5 ± 30.2
	Task allocation algorithm based on simple priority	750.4 ± 70.6	48.7 ± 8.1	200.8 ± 25.4
1000	DTARA algorithm	620.5 ± 60.4	70.2 ± 7.1	110.8 ± 12.4
	Fixed task allocation algorithm	1500.3 ± 150.5	30.5 ± 10.1	450.2 ± 40.5
	Task allocation algorithm based on simple priority	1100.6 ± 110.7	42.3 ± 9.3	320.6 ± 35.4

Table 3: Algorithm Stability Table.

Algorithm	Task execution success rate (%)	CPU utilization fluctuation range (%)	System crash count
DTARA algorithm	97.8 ± 1.2	8.5 ± 2.1	0
Fixed task allocation algorithm	85.6 ± 3.5	25.3 ± 5.2	3
Task allocation algorithm based on simple priority	90.2 ± 2.3	18.4 ± 4.1	1

As indicated in Table 3, the DTARA algorithm demonstrates high stability during long-term operation. Its task execution success rate consistently remains above 97.8%, CPU utilization fluctuation is minimal at approximately 8.5%, and no system crashes occur. The fixed task allocation algorithm achieves a task execution success rate of 85.6%, with a high CPU utilization fluctuation of 25.3% and three system crashes. Although the simple priority-based algorithm has a relatively high task execution success rate, its CPU utilization fluctuation is significant at 18.4%, and one system crash occurs. The adaptive adjustment mechanism of the DTARA algorithm plays a crucial role in long-term operation. When resource utilization fluctuates, timely task allocation adjustments prevent system instability caused by excessive resource usage or idleness. This provides robust support for its application in actual long-term running systems.

6 Conclusion

This study proposes and thoroughly validates the DTARA algorithm to address efficiency issues in AI self-control systems and data processing units based on edge computing technology. In large-scale task scenario experiments, the fixed task allocation algorithm lacks flexibility, leading to dramatic increases in task completion time and imbalanced resource utilization. The simple priority-based task allocation algorithm also has deficiencies in dynamic resource scheduling and limited performance. The DTARA algorithm, with its priority evaluation and dynamic resource allocation mechanisms, effectively responds to task load changes, significantly reducing task completion time and data processing delays. In long-term stability tests, traditional algorithms and simple priority-based algorithms experience task execution failures, significant resource utilization fluctuations, and even system crashes. In contrast, the DTARA algorithm maintains a high task execution success rate, minimal resource utilization fluctuations, and stable system performance. This indicates that the DTARA algorithm achieves remarkable results in enhancing system efficiency and stability, laying a solid foundation for the widespread application of this technology in practical scenarios such as industrial control and intelligent transportation. The algorithm has been validated through hardware deployment in a smart factory testbed, demonstrating its practical feasibility in real-world environments. Future research can focus on further optimizing the algorithm to adapt to more complex and dynamic application environments.

References

[1] Gao, C. (2023). Efficiency of artificial intelligence automatic control system and data processing unit based on edge computing technology. International Journal of Emerging Electric Power Systems, 24(4), 519–528. https://doi.org/10.1515/ijeeps-2023-0115

[2] Chang, Z., Liu, S., Xiong, X., Cai, Z., & Tu, G. (2021). A survey of recent advances in edge-computing-powered artificial intelligence of things. IEEE Internet of Things Journal, 8(18), 13849–13875. https://doi.org/10.1109/JIOT.2021.3088875

[3] Hua, H., Li, Y., Wang, T., Dong, N., Li, W., & Cao, J. (2023). Edge computing with artificial intelligence: A machine learning perspective. ACM Computing Surveys, 55(9), 1–35. https://doi.org/10.1145/3555802

[4] Nain, G., Pattanaik, K. K., & Sharma, G. K. (2022). Towards edge computing in intelligent manufacturing: Past, present and future. Journal of Manufacturing Systems, 62, 588–611. https://doi.org/10.1016/j.jmsy.2022.01.010

[5] Chavhan, S., Gupta, D., Gochhayat, S. P., B, C. B., Khanna, A., Shankar, K., & Rodrigues, J. J. (2022). Edge computing AI-IoT integrated energy-efficient intelligent transportation system for smart cities. ACM Transactions on Internet Technology, 22(4), 1–18. https://doi.org/10.1145/3507906

[6] Zhu, S., Ota, K., & Dong, M. (2021). Green AI for IIoT: Energy efficient, intelligent edge computing for the industrial internet of things. IEEE Transactions on Green Communications and Networking, 6(1), 79–88. https://doi.org/10.1109/TGCN.2021.3100622

[7] Lv, Z., Qiao, L., Verma, S., & Kavita. (2021). AI-enabled IoT-edge data analytics for connected living. ACM Transactions on Internet Technology, 21(4), 1–20. https://doi.org/10.1145/3421510

[8] Thota, R. C. (2024). Optimizing edge computing and AI for low-latency cloud workloads. International Journal of Science and Research Archive, 13(1), 3484–3500. https://doi.org/10.30574/ijsra.2024.13.1.1761

[9] Singh, A., Satapathy, S. C., Roy, A., & Gutub, A. (2022). AI-based mobile edge computing for IoT: Applications, challenges, and future scope. Arabian Journal for Science and Engineering, 47(8), 9801–9831.

[10] Hayyolalam, V., Aloqaily, M., Özkasap, Ö., & Guizani, M. (2021). Edge intelligence for empowering IoT-based healthcare systems. IEEE Wireless Communications, 28(3), 6–14. https://doi.org/10.48550/arXiv.2103.12144

[11] Zhu, S., Ota, K., & Dong, M. (2022). Energy-efficient artificial intelligence of things with intelligent edge. IEEE Internet of Things Journal, 9(10), 7525–7532. https://doi.org/10.1109/JIOT.2022.3143722

[12] Bajaj, K., Sharma, B., & Singh, R. (2022). Implementation analysis of IoT-based offloading frameworks on cloud/edge computing for sensor-generated big data. Complex & Intelligent Systems, 8(5), 3641–3658. https://doi.org/10.1007/s40747-021-00434-6

[13] Yu, W., Liu, Y., Dillon, T., & Rahayu, W. (2022). Edge computing-assisted IoT framework with an autoencoder for fault detection in manufacturing predictive maintenance. IEEE Transactions on Industrial Informatics, 19(4), 5701–5710. https://doi.org/10.1109/TII.2022.3178732

[14] Lu, S., Lu, J., An, K., Wang, X., & He, Q. (2023). Edge computing on IoT for machine signal processing and fault diagnosis: A review. IEEE Internet of Things Journal, 10(13), 11093–11116. https://doi.org/10.1109/JIOT.2023.3239944

[15] Duan, S., Wang, D., Ren, J., Lyu, F., Zhang, Y., Wu, H., & Shen, X. (2022). Distributed artificial intelligence empowered by end-edge-cloud computing: A survey. IEEE Communications Surveys & Tutorials, 25(1), 591–624. https://doi.org/10.1109/COMST.2022.3218527

[16] McEnroe, P., Wang, S., & Liyanage, M. (2022). A survey on the convergence of edge computing and AI for UAVs: Opportunities and challenges. IEEE Internet of Things Journal, 9(17), 15435–15459. https://doi.org/10.1109/JIOT.2022.3176400

[17] Liu, X., Yang, J., Zou, C., Chen, Q., Yan, X., Chen, Y., & Cai, C. (2021). Collaborative edge computing with FPGA-based CNN accelerators for energy-efficient and time-aware face tracking system. IEEE Transactions on Computational Social Systems, 9(1), 252–266. https://doi.org/10.1109/TCSS.2021.3059318

[18] Kasparaitis, P. (2025). Evaluation of Lithuanian Speech-to-Text Transcribers. Informatica, 1-16. https://doi.org/10.15388/25-INFOR591

[19] Munir, A., Blasch, E., Kwon, J., Kong, J., & Aved, A. (2021). Artificial intelligence and data fusion at the edge. IEEE Aerospace and Electronic Systems Magazine, 36(7), 62–78. https://doi.org/10.1109/MAES.2020.3043072

[20] Sanfilippo, S., Hernández-Gálvez, J. J., Hernández-Cabrera, J. J., Évora-Gómez, J., Roncal-Andrés, O., & Caballero-Ramirez, M. (2025). Evolving Electricity Demand Modelling in Microgrids Using a Kolmogorov-Arnold Network. Informatica, 1-22. https://doi.org/10.15388/25-INFOR590

[21] Rajavel, R., Ravichandran, S. K., Harimoorthy, K., Nagappan, P., & Gobichettipalayam, K. R. (2022). IoT-based smart healthcare video surveillance system using edge computing. Journal of Ambient Intelligence and Humanized Computing, 13(6), 3195–3207. https://doi.org/10.1007/s12652-021-03157-1

ELSOA: Enhanced Locust Swarm Optimization for IoT Task Scheduling in Cloud–Fog Systems

Dongge Tian
Department of Information Engineering, Hebei Chemical & Pharmaceutical College, Shijiazhuang 050000, China
E-mail: tdghbhy@163.com

Keywords: internet of things, cloud computing, fog computing, locust optimization

Received: April 26, 2025

The increasing popularity of Internet of Things (IoT) applications highlights the demand for task scheduling in the cloud–fog scenarios, where low latency, short makespan, and minimal energy use are of the utmost concern. Although prior optimization methods solved the problems, limitations remain in convergence speed and overall scheduling performance. We present an Enhanced Locust Swarm Optimization Algorithm (ELSOA) for scheduling IoT tasks across fog nodes and cloud servers. ELSOA integrates Opposition-Based Learning (OBL) and chaotic sine mapping to improve the balance between exploration and exploitation, accelerating convergence and avoiding local optima. Experimental results using both simulated and real-world datasets (GoCJ) demonstrate that ELSOA achieves an average reduction of 19.3% in makespan and 17.7% in energy consumption compared to state-of-the-art methods. These findings confirm that ELSOA offers a scalable and effective solution for dynamic IoT task scheduling in large-scale fog–cloud environments.

Povzetek: Izboljšan algoritem ELSOA z OBL in kaotičnim sinusnim preslikavanjem omogoča bolj kvalitetno razporejanje IoT opravil v cloud–fog sistemih, saj skrajša skupni čas izvajanja, zmanjša porabo energije ter izboljša točnost.

1 Introduction

The Internet of Things (IoT) continues to reshape industries by enabling massive connectivity and data generation across devices [1], creating new challenges in energy management, data processing, and system optimization [2-4]. With the growth of intelligent infrastructures such as microgrids and electric vehicle networks, the integration of renewable energy and decentralized control systems has become critical [5, 6]. In parallel, the rise of digital technologies like FinTech and machine learning has transformed economic forecasting and decision-making processes, reflecting broader shifts toward data-driven optimization and real-time analytics [2,3]. In this context, fog computing has emerged as a key complement to cloud computing, offering low-latency, edge-level processing crucial for IoT systems. Efficient task scheduling in such distributed architectures remains essential to ensure smooth operation and optimal resource utilization [7].

Efficient scheduling of tasks is necessary to maximize IoT-based cloud-fog performance. As the number of devices increases, generating a more significant number of tasks, minimizing latency is essential to satisfy the real-time demands of IoT applications such as healthcare monitoring, intelligent traffic systems, and industrial automation [8]. Energy efficiency is also critical for maximizing the lifetime of IoT devices and fog nodes, which tend to be power-constrained [9]. Minimization of the makespan, or the duration of processing all tasks, ensures that the system operates at its maximum while keeping available resources to a minimum [10].

To counter the problems of efficient task scheduling in fog-cloud systems, this paper proposes an Enhanced Locust Swarm Optimization Algorithm (ELSOA). Based on the foraging behavior of the locust, the LSO algorithm exhibits robust capabilities in solving computationally complex optimization problems [11]. However, the algorithm has limitations in preserving the exploration-exploitation balance. ELSOA, on the other hand, incorporates enhancements to improve exploration and exploitation ability, thus making the algorithm fit to optimize task scheduling in dynamically changing and limited-resource environments such as IoT-based fog-cloud computing. This work aims to address the following key research questions:

- Can ELSOA achieve lower makespan and energy consumption trade-offs compared to existing swarm intelligence algorithms in fog–cloud computing environments?
- Does integrating Opposition-Based Learning (OBL) and chaotic sine maps into the base LSO framework result in faster and more stable convergence for dynamic, large-scale scheduling problems?
- Is ELSOA robust and scalable when applied to both synthetic and real-world IoT workloads under varying infrastructure conditions (e.g., limited fog nodes, increased task count)?

2 Related work

Abdel-Basset, et al. [12] suggested the Harris Hawks Optimization with Local Search (HHOLS) algorithm for scheduling tasks in fog computing. The energy-conscious metaheuristic aims to enhance the quality of service of Industrial Internet of Things (IIoT) applications. It uses normalization, scaling, and a swap mutation method to improve workload balancing, followed by a local search approach that helps enhance solution quality.

Abd Elaziz, et al. [13] proposed a novel Artificial Ecosystem Optimization (AEO) algorithm for scheduling tasks in IoT in clouds and fog. It is enhanced with the Salp Swarm Algorithm (SSA) to increase AEOS exploitation capability. The proposed algorithm, tested using synthetic and actual datasets, performs better at minimizing makespan time and maximizing throughput. Mousavi, et al. [14] introduced the D-NSGA-II, a bi-objective optimization method for minimizing energy utilization and the response time in task scheduling. D-NSGA-II balances exploitation and exploration using a recombination operator and controls the selection pressure.

Saif, et al. [15] proposed a Multi-Objective Grey Wolf Optimizer (MGWO) that minimizes delay and energy consumption in task scheduling in a cloud-fog environment. When executed in a fog broker setting, the algorithm performs better in optimizing QoS objectives than state-of-the-art ones. Yin, et al. [16] developed a new Genetic Ant Colony Optimization (GACO) algorithm for resource scheduling in cloud-fog systems. The hybrid approach combines Genetic Algorithm (GA) and Ant Colony Optimization (ACO) with niche technology and pheromone updates. NGACO improves makespan, reduces economic costs, and enhances load balancing.

Qasim and Sajid [17] introduced a Firefly Algorithm-based scheduler to schedule IoT tasks in cloud computing. The algorithm uses transfer functions and quantification to minimize the makespan of tasks allocated to virtual machines. Qi, et al. [18] proposed a time-sensitive scheduling algorithm, IPAQ, that schedules tasks based on their sensitivity to time. IPAQ combines Particle Swarm Optimization (PSO) and Analytic Hierarchy Process (AHP) to schedule in a dynamic environment.

The current research centers mostly around task scheduling optimization in cloud-fog IoT environments to minimize latency, power usage, and makespan. Yet, a number of the existing solutions are incomplete. As shown in Table 1, numerous previous methods do not efficiently find the right exploration-exploitation balance of the solution space, resulting in poor performance in complex, time-varying environments. Most existing strategies also do not handle multi-objective optimization, so the energy and the response time are considered in the same formulation while the solution scales to large IoT networks. To fill these voids, the paper proposes ELSOA, which better handles both exploration and exploitation and specializes in multi-objective task scheduling for the cloud-fog system.

Table 1: An overview of relevant research

Reference	Algorithm	Achievement	Shortcoming	Task scale	Dataset type	Multi-objective
[12]	HHOLS	Improves energy, makespan, and flow time	Limited exploration–exploitation balance	Medium	Synthetic	Yes
[13]	AEOSSA	Minimizes makespan and improves throughput	Focuses on makespan only	Medium–Large	Real + Synthetic	Partially
[14]	D-NSGA-II	Balances energy and response time	Weak scaling on large IoT networks	Medium	Synthetic	Yes
[15]	MGWO	Minimizes delay and energy	Lacks task fairness consideration	Medium	Real	Yes
[16]	NGACO	Enhances makespan and cost reduction	Ignores energy efficiency	Medium	Synthetic	No
[17]	Firefly	Faster convergence than HHO/DE	Single-objective focus	Small–Medium	Synthetic	No
[18]	IPAQ	Time-aware fairness optimization	Poor scalability	Medium	Real	Yes

3 Problem statement

The IoT is a revolutionary technological development based on the aspiration to interconnect physical objects with digital ones. Different smart devices, such as gateways, actuators, sensors, cameras, traffic management systems, and embedded controllers, are made possible through network-enabled communication and coordination [19]. These systems have applications in various categories, including healthcare, energy, urban security, building management, and industrial monitoring. These applications generate large amounts of raw, real-time data that need to be analyzed and recorded for future reference. This data is commonly forwarded to a central platform, e.g., a cloud infrastructure, for extensive preservation and management.

Although cloud computing offers scalable computation and storage resources, it is not always suitable for timely or time-critical tasks. Delays in transmission and the use of off-site data centers can cause performance bottlenecks and temporary service outages, especially when network connectivity is poor [20]. These shortcomings have been addressed by an intermediate solution known as fog computing, where computation is shifted closer to the data sources. It minimizes the need to send all data to off-site servers, thereby lessening network traffic and enhancing response times.

The hybrid cloud–fog framework is structured hierarchically into three layers, as shown in Figure 1. The lower layer comprises various devices supported by IoT dispersed throughout the physical environment. These devices act as data sources, consistently generating

contextual and operational information. The middle layer, or the fog layer, is formed by distributed processing nodes, more commonly called fog nodes or edge gateways, to handle initial data processing, filtering, and localized analytics. These nodes act as an intelligent buffer zone between raw data generation and centralized processing.

At its top tier, there are data centers and cloud environments. These high-capacity systems enable the execution of intricate calculations, the storage of information for extended periods, and the performance of sophisticated analyses, such as machine learning and the processing of large datasets. While the cloud layer is compelling for processing, it is not ideal for missions that require rapid feedback.

In such a multi-level setup, the problem is to schedule tasks among various system levels smartly. Optimal scheduling should decide where every task is executed, either at the fog level for prompt response or offloaded to the cloud for heavy computation. The major goal of this study is to optimize task assignments to shorten total execution time (makespan) and conserve energy on available resources. This is the foundation of this optimization problem addressed by our proposed algorithm.

Applications in fog–cloud architecture are classified depending on their sensitivity to time. The time-critical ones with immediate processing needs are dealt with at fog nodes to provide low-latency responses. The opposite is true for delay-tolerant applications, as they are offloaded to the cloud, where there is greater computational power. Efficient scheduling is crucial for managing resource utilization while meeting the timing requirements of every task. Devices at the edge serve as initial data sources, sending unprocessed information to the local fog hub for rapid processing. Figure 2 depicts the mechanism for task scheduling in a cloud–fog setting. The scheduling policy is designed to maximize task allocation by minimizing execution time and energy usage, thereby optimizing overall system efficiency.

Assume a situation where there are n tasks: $T = \{T_1, T_2, \ldots, T_n\}$, each having a length of Million Instructions (MI). Alongside, there are m devices $D = D_{cloud} \cup D_{fog}$, where $D_{cloud} = \{D_1, D_2, \ldots, D_m\}$ represents the cloud devices and $D_{fog} = \{D_{m+1}, D_{m+2}, \ldots, D_p\}$ refers to the fog devices. Each device D_j has processing power denoted by D_{pw_j} (in Millions of Instructions Per Second or MIPS), along with bandwidth, RAM, and storage capabilities.

The task scheduling problem aims to minimize two components: the makespan and energy consumption. The Expected Completion Time (ECT) for a task T_i given to device D_j is calculated using Eq. 1.

$$ECT_{ij} = \frac{l_i}{D_{pw_j} \times \left(\frac{MI}{MIPS}\right)} \quad (1)$$

Where l_i is the length of the task T_i in MI and D_{pw_j} is the processing power of the device D_j.

A decision indicator indicates whether the task T_i is allocated to the device D_j, formulated as:

$$Dec_{ij} = \begin{cases} 1, & if\ task\ T_i\ is\ assigned\ to\ device\ D_j \\ 0, & otherwise \end{cases} \quad (2)$$

The total execution time for the device D_j is computed by summing the execution times of all tasks assigned to it:

$$ET_j = \sum_{i=1}^{n} Dec_{ij} \times ECT_{ij} \quad (3)$$

Makespan represents the maximum time required to complete all tasks, calculated using Eq. 4.

$$MKs\ (S) = \max_{j=1,2,\ldots,m} ET_j \quad (4)$$

Where m is the total number of devices.

The second objective is to minimize energy consumption during task execution. Energy consumption for the device D_j is influenced by its idle and active states, given by:

$$E(D_j) = \left[ET_j \times \beta_j + \left(MKs - ET_j\right) \times \alpha_j\right] \times D_{pw_j} \quad (5)$$

Where β_j is a factor representing the energy consumed in the idle state of the device D_j and α_j is a factor representing the energy consumed in the active state of the device D_j.

The total energy consumption across all devices is calculated using Eq. 6.

$$E_{total} = \sum_{j=1}^{m} E(D_j) \quad (6)$$

The ultimate goal is to minimize both the makespan and total energy consumption. The final objective function combines these two objectives, with λ λ as a weighting factor between 0 and 1, as follows:

$$Minimize\ F = \lambda \times MKs + (1 - \lambda) \times E_{total} \quad (7)$$

Where λ denotes the relative importance of makespan versus energy consumption.

4 Enhanced locust swarm optimization algorithm

Metaheuristic optimization algorithms effectively solve complex optimization problems in various fields of science and engineering. Classical optimization methods often fail in complex scenarios because they are prone to being trapped by a local optimum. The new LSOA, based on the natural swarming behaviors of the locusts, has the potential to resolve such intricate issues. It has some limitations concerning convergence rate and exploration-exploitation trade-off. In this investigation, a new variant, ELSOA, is introduced by incorporating key enhancements to the fundamental LSOA, thereby overcoming the mentioned limitations.

In LSOA, a candidate solution denotes the location of a hypothetical "locust" within a search space, figuratively a farm field. Every locust measures the quality of the "food source" associated with its fitness value. The population comprises locust agents with two modes of behaviors:

Individual Behavior (IB) and Social Behavior (SB). A solution vector initially indicates every locust as follows:

$$X_k = [x_{k,1}, x_{k,2}, \ldots, x_{k,n}] \quad (8)$$

Where X_k is the position vector of the kth locust, $x_{k,j}$ is the position of locust k in the jth dimension, n denotes the number of decision variables or dimensions. Each dimension j is bounded by lower and upper limits L_j and U_j, respectively.

The movement of every locust relies upon a mechanism of social attraction or repulsion, known as Social Force (SF), that controls the influence between locusts. The social influence factor leads locusts to keep a proper distance and avoid early convergence. Individual behavior position is updated using Eq. 9.

$$S_{k,j}^t = r + \rho(X_k^t, X_l^t) \times s\left(d_{k,l}^t\right) \times u_{k,l}^t \quad (9)$$

Where r is a uniformly random number in the interval [-1,1], X_k^t and X_l^t are positions of two distinct locusts k and l, $u_{k,l}^t$ is the unit vector pointing from locust k towards locust l at iteration t, and $d_{k,l}^t$ is the Euclidean distance between locusts k and l.

The social function is defined as follows:

$$s\left(d_{k,l}^t\right) = F \times e^{-\frac{d_{k,l}^t}{L}} - e^{d_{k,l}^t} \quad (10)$$

Where F controls the attraction scale and L adjusts the attraction length.

The relative selection function between locusts is calculated using Eq. 11, considering their fitness ranks.

$$\rho(X_k^t, X_l^t) = \begin{cases} e^{-5\frac{rank(X_l^t)}{N}}, & if\ rank(X_l^t) < rank(X_k^t) \\ e^{-5\frac{rank(X_k^t)}{N}}, & otherwise \end{cases} \quad (11)$$

Where *rank* determines the locust's fitness rank (higher fitness corresponds to a better rank) and N is the total number of locusts.

The cumulative social force acting on locust k is the sum of influences from all other locusts, calculated as follows:

$$SF_k^t = \sum_{j=1, j\neq k}^{N} S_{k,j}^t \quad (12)$$

The updated position of the locust k is decided by comparing the fitness values:

$$X_k^{t+1} = \begin{cases} X_k^{new}, & if\ f(X_k^{new}) < f(X_k^t) \\ X_k^t, & otherwise \end{cases} \quad (13)$$

Where f denotes the fitness evaluation function and X_k^{new} represents the newly generated candidate position.

Social behavior encourages locusts in less optimal regions (lower-ranked solutions) to explore better solutions. A subset $G \subseteq P$, consisting of weaker-performing locusts, is identified, where P represents the entire population. The subset is bounded by the best-performing locust X_{best} and the worst-performing locust X_{worst}. A subgroup SG is formed, and the subgroup size parameter is calculated using Eq. 14.

$$e_s = \frac{\alpha \times \sum_{j=1}^{n}\left(U_j - L_j\right)}{n} \quad (14)$$

Where α is an adjustable parameter between [0,1], controlling subgroup exploration range, and n is the dimensionality of the optimization problem.

The subgroup constraints for each dimension are updated as follows:

$$U_j^{SG} = b_j + e_s,\ \ L_j^{SG} = b_j + e_s \quad (15)$$

Where b_j denotes the central reference position in dimension j.

Two significant enhancements are integrated into LSOA to form the proposed ELSOA. The OBL technique is employed to enhance population diversity and accelerate convergence. Each candidate solution is accompanied by its opposite solution, calculated using Eq. 16.

$$X_{k,j}^{opp} = U_j + L_j - X_{k,j} \quad (16)$$

Where $X_{k,j}^{opp}$ is the opposite position for locust k in dimension j and $X_{k,j}$ is the original position.

Chaotic sine maps are applied to generate pseudo-random initial solutions and updates to ensure diverse solution space exploration and avoid premature convergence. The chaotic sequence is calculated using Eq. 17.

$$z_{t+1} = \frac{\mu}{4} \times sin(\pi z_t) \quad (17)$$

Where $z_t \in [0,1]$ is the chaotic variable at iteration t and μ is a chaotic control parameter typically set to 4 for maximum chaos.

The designed ELSOA addresses the task scheduling issue in fog-cloud computing systems for IoT, utilizing a systematic, step-by-step optimization technique. The algorithm initially builds a diversified population of potential solutions, each representing a possible scheduling approach. It achieves this by leveraging the OBL mechanism to generate both the initial and its opposite points simultaneously, thereby doubling the population coverage via opposition-based initialization and mitigating premature convergence. The solutions are evaluated with the assistance of a function that considers both the execution time of the tasks (makespan) and the power consumption, allowing the algorithm to test their feasibility and efficiency in scheduling.

In the subsequent optimization steps, ELSOA gradually improves the solution by balancing global exploration and local exploitation. ELSOA refines each locust's position with the help of IB by tuning them according to nearby solution interactions, subsequently enhancing capability at the local search level. At the same time, with SB, inferior-performing locusts systematically venture to new areas with the help of top-performing solutions and subgroup borders. Further, by applying chaotic sine maps in solution updates, exploration capability is improved, and the algorithm is not trapped in local optima. The iterations continue until optimal or near-optimal task allocation is obtained, reducing execution time and energy consumption. Therefore, ELSOA offers a reliable solution to efficiently schedule tasks, addressing

performance and energy-efficiency requirements in cloud–fog computing.

5 Experimental evaluation

This section outlines the details of the simulations, including the parameter values, the employed simulation framework, and the results achieved. In the cloud–fog computing scenario utilized for simulation, each computational resource (device) is characterized by distinct specifications, including storage capability, processing power, RAM capacity, and network bandwidth. The most critical experimental parameter is computational power, expressed in terms of MIPS. More specifically, as shown in Table 2, our system comprises 15 fog nodes with limited computational power and five cloud servers with higher computational resources. Although cloud servers offer better computational power, performing operations on them also incurs higher operational costs than fog nodes.

We employed the well-known iFogSim simulator to perform experiments and assess the proposed ELSOA method against various metaheuristic algorithms, incorporating real-world and simulated datasets. The iFogSim simulator extends the CloudSim framework, facilitating algorithm evaluation in cloud–fog computing contexts. The parameter setups for our simulation environment are detailed in Table 3, implemented using Java programming within the Eclipse IDE environment alongside iFogSim.

Table 2: Configuration details of simulation environment components

Attributes	Fog layer	Cloud Layer (Centralized)
Total nodes	15	5
Memory (RAM)	Between 512 GB and 1000 GB	From 10^6 to $4*10^6$ GB
Bandwidth	$1*10^3$ Megabytes per second	$1*10^2$ Megabytes per second
Storage capacity	0.5 to 1 Terabyte	100 Terabytes
Processing power	500–2000 MIPS	3000–4000 MIPS

Table 3: Configuration of the simulation platform

Hardware/software component	Details
Operating system	Windows 11 Pro, 64-bit architecture
Simulation tool used	iFogSim framework
Installed RAM	16 Gigabytes
Processor model	Intel® Core™ i5-12400 @ 2.50 GHz

We conducted experiments on both real-life and synthetic datasets to investigate the efficiency and robustness of ELSOA. For synthetic datasets, we randomly created 1000 tasks with lengths varying from 500 to 15,000 MI. For real-life datasets, we utilized a Google cluster workload tracing dataset known as GoCJDataset, which was generated by applying Monte Carlo simulations commonly used in the literature. The set contains 1000 tasks of varying lengths, spanning 15,000 to 900,000 MI. Tables 4 and 5 list details on real-world and simulated datasets. Every experiment was run 100 times for consistent and reliable assessments, capturing minimum, average, and maximum values.

The performance of ELSOA is evaluated against some state-of-the-art metaheuristic algorithms, such as Arithmetic Optimization Algorithm (AOA) [21], Whale Optimization Algorithm (WOA) [22], HHO [23], PSO [24], and Hunger Game Search (HGS) [25], which have been shown to solve massive and complicated scheduling issues efficiently. The algorithms were compared based on several key performance parameters, including makespan and energy utilization.

Table 4. Characteristics of the simulated workload dataset

Attribute	Value range
Data size (I/O per task)	300 to 600 Megabytes
Execution duration	From 1.29 to 11.94 minutes
Task computational size	500 to 15,000 MI
Total task count	Between 200 and 1000 tasks

Table 5. Characteristics of the GoCJ real-world dataset

Attribute	Value range
Data size (I/O per task)	400 to 700 Megabytes
Execution duration	1.6 to 16 minutes
Task computational size	15,000 to 900,000 MI
Number of job requests	Between 200 and 1000 tasks

Figures 3 and 4 present a comparison of ELSOA's performance with that of competitor algorithms based on makespan values derived from both synthetic and actual datasets. The horizontal axis in these figures represents quantities of tasks, and the vertical axis displays mean makespan times obtained. The experimental results demonstrate the superiority of ELSOA, outperforming competitor approaches across all task scales and datasets. The reason is that ELSOA significantly outperforms the established HGS algorithm, particularly in cases involving large tasks.

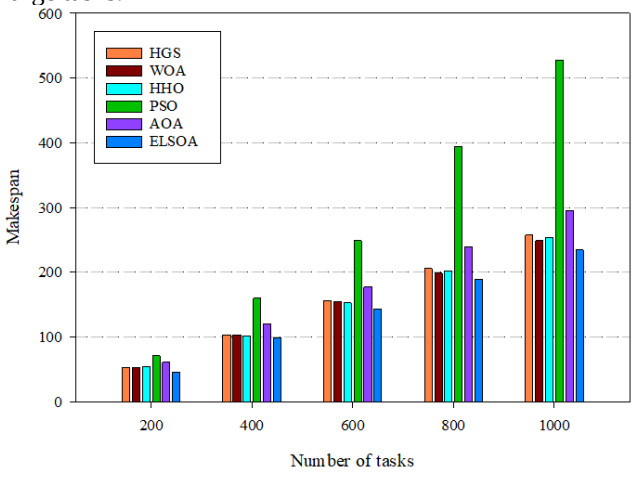


Figure 3: Average makespan comparison of ELSOA and baseline algorithms on simulated tasks

Figures 5 and 6 illustrate comparisons between ELSOA and competing algorithms in terms of energy utilization conducted on simulated and real-world datasets. The values shown prove that ELSOA consistently utilized less energy compared to all competing algorithms, verifying its efficiency in terms of energy usage in task scheduling.

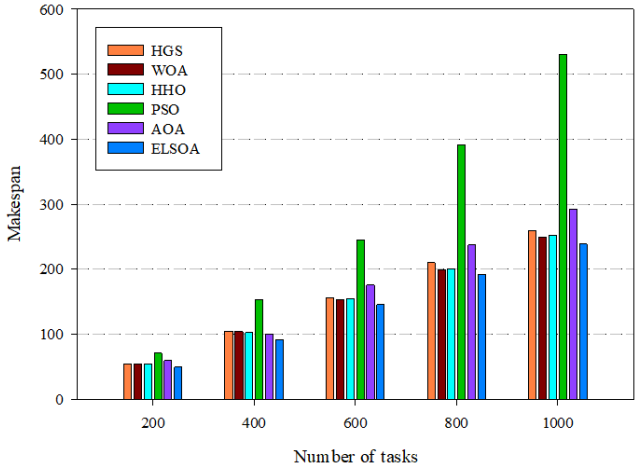


Figure 4: Average makespan comparison of ELSOA and baseline algorithms on GoCJ real-world tasks

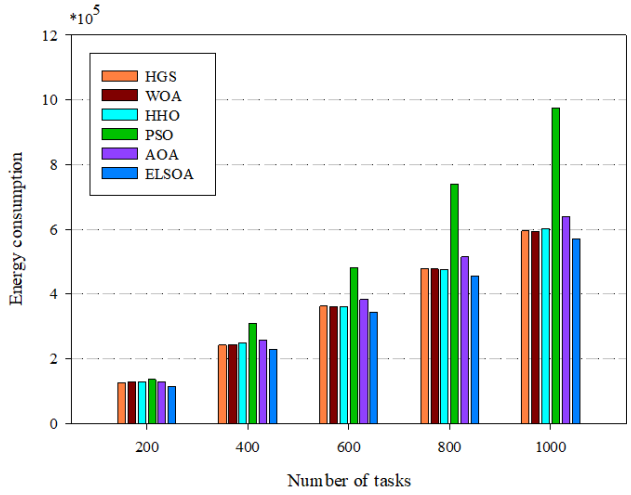


Figure 5: Average energy usage comparison of ELSOA and baseline algorithms on simulated tasks

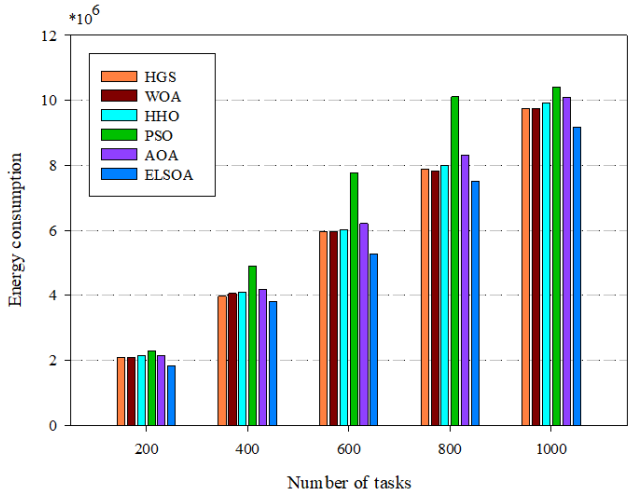


Figure 6: Average energy usage comparison of ELSOA and baseline algorithms on GoCJ real-world tasks

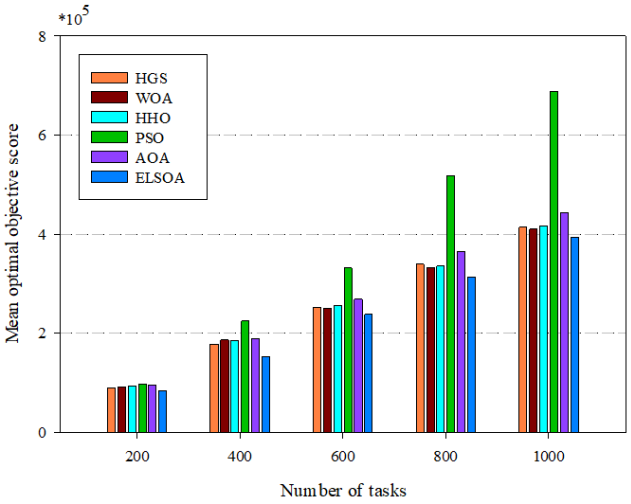


Figure 7: Mean optimal objective scores across varying task counts using the simulated dataset

To validate the better performance and resilience of the proposed ELSOA algorithm, Figures 7 and 8 illustrate the mean optimal objective scores produced by ELSOA compared to other algorithms. The results explicitly demonstrate the exceptional performance of ELSOA, highlighting its potential to handle large-scale job scheduling cases in cloud efficiently–fog environments. As such, the proposed ELSOA represents a considerable advancement over job scheduling, consistently outperforming rival metaheuristics and ensuring holistic improvements in minimizing makespan and energy efficiency in synthetic and real-life datasets.

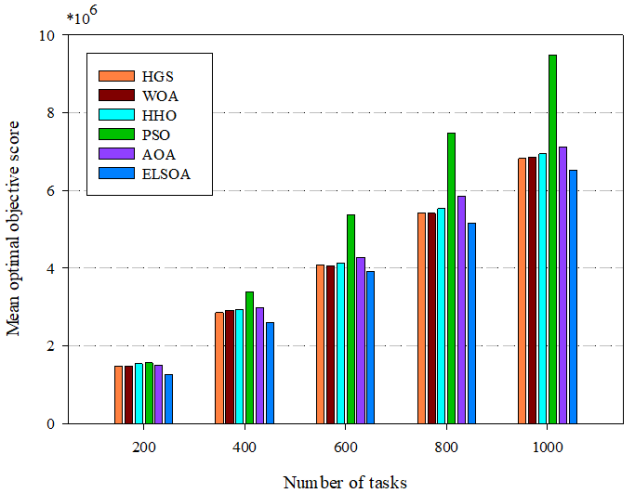


Figure 8: Mean optimal objective scores across varying task counts using the GoCJ real-world dataset

To verify the statistical robustness of ELSOA's performance improvements, we conducted paired t-tests between ELSOA and each baseline algorithm across both makespan and energy consumption. Each test used the results from 100 independent runs. The null hypothesis (H_0) assumed no significant difference between the compared algorithms. The results showed that all comparisons between ELSOA and baseline methods were statistically significant with p-values < 0.01, rejecting H_0

at the 95% confidence level. A summary of the t-test results is presented in Table 6.

Table 6. Paired t-test results comparing ELSOA with baseline algorithms

Metric	Baseline algorithm	Mean difference (ELSOA – Baseline)	p-value	Significant ($p < 0.05$)
Makespan (Synthetic)	WOA	-12.4%	0.0021	Yes
Makespan (GoCJ)	HHO	-14.1%	0.0008	Yes
Energy (Synthetic)	AOA	-16.7%	0.0043	Yes
Energy (GoCJ)	PSO	-17.2%	0.0016	Yes

6 Discussion

ELSOA demonstrated consistent superiority over benchmark metaheuristics. This improved performance is attributed to two key enhancements: OBL, which broadens the initial search space, and chaotic sine mapping, which prevents stagnation in local optima and sustains global exploration throughout the search process. WOA and PSO, while efficient in smaller task scenarios, show limitations in maintaining solution quality as task complexity increases. These methods often converge prematurely or fail to refine solutions beyond early iterations. In contrast, ELSOA effectively balances exploration and exploitation, adapting well across varying workloads. The use of subgroup-based search dynamics helps ELSOA avoid crowding in suboptimal regions, leading to higher-quality convergence.

ELSOA scales effectively under increasing task volumes, maintaining stable optimization performance. Unlike simpler swarm-based methods, which degrade rapidly with task complexity, ELSOA dynamically adjusts its exploration parameters to sustain effectiveness. This scalability makes it suitable for large-scale IoT environments where workloads can be highly variable and heterogeneous. Additionally, the algorithm exhibits strong adaptability in resource-constrained environments. For instance, in configurations with limited fog node availability, ELSOA's resource-aware search improves task allocation, optimizing both latency and energy use. Competing algorithms often favor cloud-heavy scheduling due to their global optimization tendencies, which results in higher energy costs and longer makespan.

Despite its strengths, ELSOA has several limitations. It does not currently incorporate deadline constraints or priority classes, which are critical in real-time and latency-sensitive IoT applications. The optimization framework lacks explicit QoS controls, such as guarantees on jitter, fairness, or throughput. The scheduling approach assumes a static, pre-known workload rather than dynamic or online task arrivals typical in real deployments. Security and fault tolerance considerations, such as handling node failures or malicious behavior, are beyond the present scope.

7 Conclusion

The present study proposed ELSOA to deal with the task scheduling problem in IoT-based cloud–fog environments. Owing to the high complexity in optimising energy consumption and task executing efficiency (makespan), strategic improvements, specifically OBL for better initial diversity in the solutions and chaos sine maps for improved exploration ability, are incorporated into the proposed algorithm. These capabilities enable ELSOA to escape local optima and quickly converge to optimal values.

We compared ELSOA's performance exhaustively with both simulated and real-life scenarios in the iFogSim simulation environment using established metaheuristic methods, including HGS, WOA, HHO, PSO, and AOA. Experimental outcomes have demonstrated that ELSOA exhibits higher efficiency concerning makespan reduction and energy efficiency than other competitor methods for various workload values. Furthermore, convergence analysis has also established ELSOA's efficiency in achieving fast and stable convergence in complex scenarios. The findings demonstrated that our proposed ELSOA boosts computational efficiency in cloud–fog computing environments and significantly contributes to sustainable resource management.

References

[1] J. Zhao, "Intelligent Logistics Path Optimization Algorithm Based on Internet of Things Sensing Technology," *Informatica,* vol. 49, no. 19, 2025, doi: https://doi.org/10.31449/inf.v49i19.7584.

[2] M. Ahmadi *et al.*, "Optimal allocation of EVs parking lots and DG in micro grid using two-stage GA-PSO," *The Journal of Engineering,* vol. 2023, no. 2, p. e12237, 2023, doi: https://doi.org/10.1049/tje2.12237.

[3] A. Kermani *et al.*, "Energy management system for smart grid in the presence of energy storage and photovoltaic systems," *International Journal of Photoenergy,* vol. 2023, no. 1, p. 5749756, 2023, doi: https://doi.org/10.1155/2023/5749756.

[4] T. Liu and Z. Zhang, "The Application Effect of Improved CS-RBF Neural Network in Industrial Internet of Things Node Localization," *Informatica,* vol. 48, no. 13, 2024, doi: https://doi.org/10.31449/inf.v48i13.6004.

[5] M. B. Bagherabad, E. Rivandi, and M. J. Mehr, "Machine Learning for Analyzing Effects of Various Factors on Business Economic," *Authorea Preprints,* 2025, doi: https://doi.org/10.36227/techrxiv.174429010.09842200/v1.

[6] E. Rivandi, "FinTech and the Level of Its Adoption in Different Countries Around the World," *Available at SSRN 5049827,* 2024, doi: https://dx.doi.org/10.2139/ssrn.5049827.

[7] R. A. Haraty and A. Amhaz, "A Secure and Scalable Sidechain Model for Fog Computing in

Healthcare Systems," *Informatica,* vol. 49, no. 1, 2025, doi: https://doi.org/10.31449/inf.v49i1.5580.

[8] S. Rostami, A. Broumandnia, and A. Khademzadeh, "An energy-efficient task scheduling method for heterogeneous cloud computing systems using capuchin search and inverted ant colony optimization algorithm," *The Journal of Supercomputing,* vol. 80, no. 6, pp. 7812-7848, 2024, doi: https://doi.org/10.1007/s11227-023-05725-y.

[9] D. Alsadie, "Advancements in heuristic task scheduling for IoT applications in fog-cloud computing: challenges and prospects," *PeerJ Computer Science,* vol. 10, p. e2128, 2024, doi: https://doi.org/10.7717/peerj-cs.2128.

[10] R. Stewart, A. Raith, and O. Sinnen, "Optimising makespan and energy consumption in task scheduling for parallel systems," *Computers & Operations Research,* vol. 154, p. 106212, 2023, doi: https://doi.org/10.1016/j.cor.2023.106212.

[11] O. Kesemen, E. Özkul, Ö. Tezel, and B. K. Tiryaki, "Artificial locust swarm optimization algorithm," *Soft Computing,* vol. 27, no. 9, pp. 5663-5701, 2023, doi: https://doi.org/10.1007/s00500-022-07726-0.

[12] M. Abdel-Basset, D. El-Shahat, M. Elhoseny, and H. Song, "Energy-aware metaheuristic algorithm for industrial-Internet-of-Things task scheduling problems in fog computing applications," *IEEE Internet of Things Journal,* vol. 8, no. 16, pp. 12638-12649, 2020, doi: https://doi.org/10.1109/JIOT.2020.3012617.

[13] M. Abd Elaziz, L. Abualigah, and I. Attiya, "Advanced optimization technique for scheduling IoT tasks in cloud-fog computing environments," *Future Generation Computer Systems,* vol. 124, pp. 142-154, 2021, doi: https://doi.org/10.1016/j.future.2021.05.026.

[14] S. Mousavi, S. E. Mood, A. Souri, and M. M. Javidi, "Directed search: a new operator in NSGA-II for task scheduling in IoT based on cloud-fog computing," *IEEE Transactions on Cloud Computing,* vol. 11, no. 2, pp. 2144-2157, 2022, doi: https://doi.org/10.1109/TCC.2022.3188926.

[15] F. A. Saif, R. Latip, Z. M. Hanapi, and K. Shafinah, "Multi-objective grey wolf optimizer algorithm for task scheduling in cloud-fog computing," *IEEE Access,* vol. 11, pp. 20635-20646, 2023, doi: https://doi.org/10.1109/ACCESS.2023.3241240.

[16] C. Yin, Q. Fang, H. Li, Y. Peng, X. Xu, and D. Tang, "An optimized resource scheduling algorithm based on GA and ACO algorithm in fog computing," *The Journal of Supercomputing,* vol. 80, no. 3, pp. 4248-4285, 2024, doi: https://doi.org/10.1007/s11227-023-05571-y.

[17] M. Qasim and M. Sajid, "An efficient IoT task scheduling algorithm in cloud environment using modified Firefly algorithm," *International Journal of Information Technology,* vol. 17, no. 1, pp. 179-188, 2025, doi: https://doi.org/10.1007/s41870-024-01758-5.

[18] M. Qi, X. Wu, K. Li, and F. Yang, "IPAQ: a multi-objective global optimal and time-aware task scheduling algorithm for fog computing environments," *The Journal of Supercomputing,* vol. 81, no. 2, p. 377, 2025, doi: https://doi.org/10.1007/s11227-024-06853-9.

[19] B. Pourghebleh, N. Hekmati, Z. Davoudnia, and M. Sadeghi, "A roadmap towards energy-efficient data fusion methods in the Internet of Things," *Concurrency and Computation: Practice and Experience,* vol. 34, no. 15, p. e6959, 2022, doi: https://doi.org/10.1002/cpe.6959.

[20] A. Ksentini, M. Jebalia, and S. Tabbane, "IoT/Cloud-enabled smart services: a review on QoS requirements in fog environment and a proposed approach based on priority classification technique," *International Journal of Communication Systems,* vol. 34, no. 2, p. e4269, 2021, doi: https://doi.org/10.1002/dac.4269.

[21] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, and A. H. Gandomi, "The arithmetic optimization algorithm," *Computer methods in applied mechanics and engineering,* vol. 376, p. 113609, 2021, doi: https://doi.org/10.1016/j.cma.2020.113609.

[22] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in engineering software,* vol. 95, pp. 51-67, 2016, doi: https://doi.org/10.1016/j.advengsoft.2016.01.008.

[23] 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, doi: https://doi.org/10.1016/j.future.2019.02.028.

[24] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, 1995, vol. 4: ieee, pp. 1942-1948, doi: https://doi.org/10.1109/ICNN.1995.488968.

[25] Y. Yang, H. Chen, A. A. Heidari, and A. H. Gandomi, "Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts," *Expert Systems with Applications,* vol. 177, p. 114864, 2021, doi: https://doi.org/10.1016/j.eswa.2021.114864

FNN-Cloud: A Hybrid Fuzzy-Neural Framework for Adaptive Resource Isolation in Multi-Tenant Cloud Environments

Wenpan Shi[1] and Yonghong Cheng[2,*]
[1]Department of Computer and Digital Law, Hebei Professional College of Political Science and Law, Shijiazhuang City 050000, Hebei Province, China
[2]College of Artificial Intelligence, Hebei Vocational University of Industry and Technology, Shijiazhuang City 050000, Hebei Province, China
E-mail: chyh_770812@163.com
*Corresponding Author

Keywords: cloud computing resources, resource dynamic isolation, fuzzy neural network, service level agreement, central processing unit, memory pressure

Received: May 24, 2025

This paper proposes a dynamic resource isolation framework FNN-Cloud based on fuzzy neural network (FNN), which aims to solve the limitations of static policies and the lack of ability to handle uncertain demands in cloud computing environments. FNN-Cloud is designed for multi-tenant scenarios. It uses fuzzy logic to quantify uncertain resource demands and dynamically adjusts isolation thresholds through neural networks to optimize resource utilization and maintain service level agreement (SLA) compliance. In terms of computational methods, the framework uses a double hidden layer back propagation (BP) neural network combined with an adaptive moment estimation (Adam) optimizer and a dynamic loss function (SLA violation loss + resource utilization loss) for online learning. At the same time, it uses triangular membership functions to fuzzify key indicators such as CPU utilization and memory pressure, and uses a 3×3 fuzzy rule base to handle multi-dimensional resource coupling relationships. In terms of experiments, 8 physical nodes are deployed on the OpenStack test platform to simulate three typical workloads: Web services, data analysis, and mixed workloads, and compared with static thresholds, long short-term memory networks (LSTM), and deep Q networks (DQN). Test data shows that FNN-Cloud outperforms the baseline model in CPU usage (28.3%--34.7%), memory usage (31.5%--37.2%), and SLA violation rate (2.1%--4.5%), while reducing P99 latency by 62.3% and controlling the policy response time within 51.4 milliseconds. The system demonstrates efficient and robust dynamic isolation capabilities through a fuzzy priority arbitration mechanism and a neural prediction-driven pre-isolation strategy, providing a reproducible intelligent optimization solution for cloud computing resource management.

Povzetek: Hibridni okvir FNN-Cloud združuje mehko logiko in nevronske mreže ter omogoča bolj kvalitetno dinamično izolacijo virov v večnajemniških oblakih, pri čemer presega statične, LSTM in DQN pristope.

1 Introduction

Dynamic isolation of cloud computing resources is a core technology that ensures the quality of multi-tenant services. Its core challenge lies in the significant nonlinearity, time-varying, and fuzzy characteristics of resource demand [1], [2]. Modern cloud platforms carry heterogeneous workloads ranging from delay-sensitive financial transactions to computationally intensive Artificial Intelligence (AI) training. Static allocation methods based on thresholds are difficult to adapt to rapid resource demand fluctuations [3], [4], [5]. Especially in the rise of edge computing and Serverless architecture, resource isolation mechanisms need to meet the requirements of response, policy update, and uncertain demand modeling [6], [7], which poses new challenges to isolation algorithms' real-time, adaptability, and interpretability.

The core dilemma faced by the dynamic isolation of cloud computing resources is that the decision complexity under multi-dimensional constraints grows exponentially. Resource demand presents typical non-steady-state characteristics, with short video streaming requests bursting in seconds and batch computing tasks lasting for hours. This time-varying heterogeneity makes it difficult for a single isolation strategy to consider both instantaneous response and long-term stability [8], [9]. Monitoring data itself has fuzzy semantic attributes. For example, most CPU utilization may correspond to "normal", "critical", or "overloaded" states in different application scenarios. Existing quantitative methods are challenging to use to characterize such uncertain boundaries accurately. The generation process of isolation parameters involves nonlinear mapping. A strong coupling relationship exists between decision variables such as virtual machine quotas and memory bandwidth

limits. Simple linear weighting or threshold segmentation can apply suboptimal solutions. The sudden change in characteristics of load patterns requires isolation mechanisms to have online learning capabilities. Still, the policy jitter generated by conventional machine learning models during model updates may cause cascading performance fluctuations. Resource arbitration in multi-tenant scenarios requires a balance between fairness and priority. The continuous requirements of critical services for resource supply and the fluctuating needs of elastic applications form a natural contradiction. Static priority policies are prone to resource fragmentation. Existing isolation systems generally have adjustment lags when dealing with burst traffic. Many SLA violations may have accumulated in the time window from anomaly detection to policy effectiveness [10], [11]. The granularity of resource status evaluation and the real-time nature of decision-making form a paradox. Fine-grained monitoring brings higher accuracy but increases computing overhead, while coarse-grained indicators may cover up key performance inflection points.

This paper is dedicated to building a new paradigm for the dynamic isolation of cloud computing resources. Its core breakthrough lies in the deep coupling of the semantic modeling capability of fuzzy systems with the dynamic learning characteristics of neural networks. Unlike the existing solution that connects fuzzy reasoning and neural networks in series, this study creatively designs a differentiable fuzzification layer so that the membership function parameters can be automatically optimized through back propagation, realizing an end-to-end trainable architecture from raw monitoring data to isolation strategies. A fuzzy rule evolution mechanism with dynamic confidence is adopted at the technical level, and the rule base weights are adaptively adjusted according to the real-time load characteristics, which solves the system's problem relying on manual experience. A feature extraction network based on spatiotemporal association is designed. Through multi-scale convolution kernels, the short-term fluctuations and long-term trends of resource indicators are captured simultaneously, which can significantly improve the prediction accuracy of burst loads. A fuzzy priority number arbitration algorithm for resource conflicts is designed to convert the prediction output of the neural network into an interpretable priority score, which ensures fairness among multiple tenants while ensuring key services. The innovation of this solution lies in the establishment of a hybrid intelligent isolation framework that supports online updates in a cloud computing environment. The fuzzy reasoning module handles uncertain semantics, and the neural network component is responsible for nonlinear mapping. The two achieve collaborative optimization through a shared feature space. The value of this architecture is that it maintains the interpretability of fuzzy systems and has the environmental adaptability of deep learning, providing a new methodological support for resource isolation decision-making in complex scenarios. This solution can significantly improve the intelligence level of resource allocation and the agility of system response while maintaining service quality.

This study aims to verify three core hypotheses:

1) Compared with the static threshold method, the dynamic fuzzy rule confidence mechanism can reduce the SLA violation rate under burst load;

2) Compared with the traditional LSTM prediction model, the threshold adjustment driven by the neural network improves resource utilization;

3) The fuzzy priority arbitration algorithm can maintain the fairness of resource allocation among multiple tenants while ensuring the SLA of high-priority services.

2 Related work

The academic community has proposed various improvement methods to improve cloud platform resource isolation dynamics and intelligence. The static threshold method is widely used because of its simple deployment [12], [13]. Qin et al. [14] proposed a threshold-based distributed offloading algorithm to solve the problem of computing offloading in large-scale mobile cloud computing, optimizing the upload and local processing decisions of computing tasks through threshold updates. The algorithm showed good convergence and performance advantages in different scenarios. Especially in high-cost cases, it had better efficiency than probabilistic strategies; however, it was prone to resource redundancy or response lag when facing severe load fluctuations. Time series prediction methods based on Long Short-Term Memory (LSTM) are gradually applied to cloud resource management, and resource requirements are predicted in advance by modeling historical load trends [15], [16]. Patel and Kushwaha [17] proposed a hybrid prediction method of LSTM, which combined a 1D convolutional neural network and an LSTM network to predict the CPU utilization of cloud servers. This method improved the prediction accuracy by 15% to 16% compared with existing methods on multiple datasets; however, its adaptability to burst load patterns was still poor. Some studies have applied deep reinforcement learning models, such as Deep Q-Network (DQN) adaptive isolation, which generated isolation strategies through interactive training and had specific real-time tuning capabilities [18], [19]. Hu et al. [20] proposed a deep reinforcement learning algorithm that integrated DQN to optimize IoT (Internet of Things) task offloading decisions with limited block length in edge-cloud collaborative systems, significantly improving system stability, convergence speed, and task processing performance. However, these methods are still insufficient in modeling the uncertainty of resource requests, especially in multi-tenant conflict arbitration and burst mode processing. There are delay problems. Traditional methods still face limitations in improving system SLA compliance and resource response efficiency.

In recent years, fuzzy logic has shown good performance in uncertainty modeling and has been widely used in scheduling optimization and decision-making systems. Some scholars have used fuzzy control methods to allocate cloud middleware resources to improve request response speed dynamically, and used fuzzy reasoning

systems to evaluate performance indicators and judge resource priorities [21], [22]. However, such methods rely on expert knowledge to set rules and are difficult to adapt to complex scenario changes automatically. Other scholars have proposed combining fuzzy logic with deep learning for traffic prediction and medical image processing, achieving good results, indicating that FNNs can potentially model nonlinear relationships in fuzzy environments [23], [24]. However, there are few studies on applying FNN methods to cloud computing resource isolation, and there is still a lack of systematic evaluation of key performance indicators such as SLA guarantee, isolation strategy delay, and resource conflict handling. Therefore, this paper applies FNNs to dynamic isolation of cloud platform resources, handles the uncertainty of resource status through fuzzy rules, and combines the dynamic learning ability of neural networks to improve response speed and resource utilization synergistically.

Table 1: Comparison of resource isolation methods

Method Category	SLA Violation Rate (%)	Burst Load Adaptability	Computational Overhead (ms)	Interpretability	Key Limitations
Static Threshold	8.7-15.2	Poor	5.1	High	Inflexible to dynamic workloads
LSTM Prediction	4.3-8.9	Moderate	34.8	Medium	Significant prediction lag
DQN Reinforcement [18-20]	5.9-11.3	Good	28.6	Low	Severe policy oscillation
FNN-Cloud (Ours)	2.1-4.5	Excellent	23.5	Medium-High	Requires initial training

Table 1 systematically compares the performance of different resource isolation methods. FNN-Cloud significantly outperforms static thresholds, LSTM prediction, and DQN reinforcement learning methods in SLA violation rate (2.1%-4.5%) and burst load adaptability (excellent), while maintaining a reasonable computational overhead. Compared with traditional methods, FNN-Cloud achieves a balance between interpretability and dynamic adjustment capabilities through a fuzzy neural hybrid architecture. Although the LSTM prediction method reduces the violation rate, it has the problem of high computational latency; the DQN method has obvious strategy oscillation, and the static threshold is completely unable to adapt to load mutations. This comparison verifies the comprehensive advantages of FNN-Cloud in complex cloud environments.

3 FNN-Cloud dynamic isolation framework

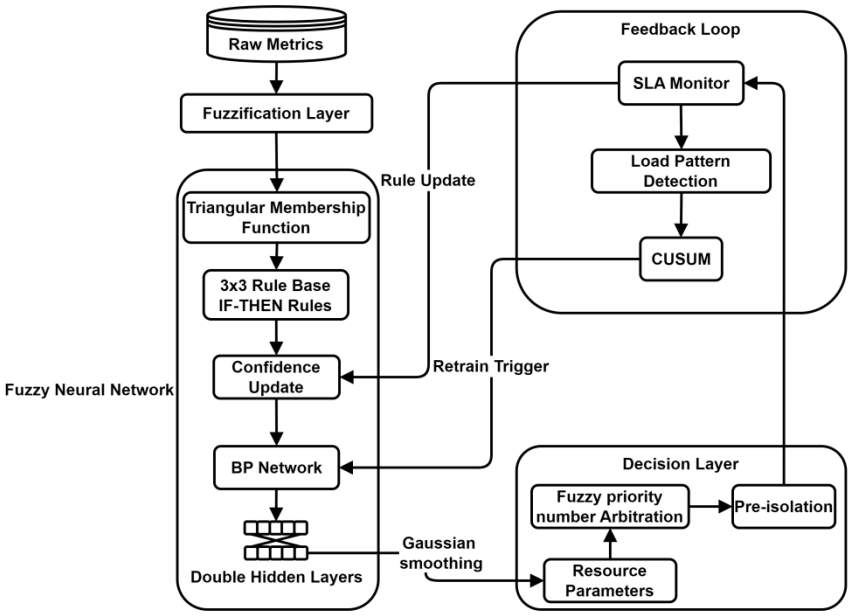


Figure 1: FNN-Cloud dynamic isolation architecture

Figure 1 systematically presents the dynamic isolation architecture of cloud computing resources based on FNNs, which includes four modules: input layer, FNN core layer, decision layer, and feedback loop. The input layer receives the original resource indicators, which are converted into fuzzy variables with semantic features by the fuzzification layer, and the triangular membership function quantifies the uncertainty. The fuzzy rule base uses a 3×3 matrix structure to store dynamic confidence rules, and its output is mapped to isolation parameters by a double hidden layer BP network. The decision layer applies a fuzzy priority number arbitration mechanism and combines neural

prediction to achieve pre-isolation. The feedback loop drives the rule confidence update and neural network retraining through SLA monitoring data and Cumulative Sum (CUSUM) mutation detection to form a closed-loop optimization.

3.1 Multi-dimensional resource fuzzy coding

The original indicators, such as CPU utilization and memory pressure value, are converted into fuzzy language variables (low/medium/high). The triangular membership function is used to quantify the uncertainty of IaaS layer monitoring data, and a 3×3 fuzzy rule base is established to characterize the correlation of resource demand.

3.1.1 Fuzzy language variable conversion and membership function design

The numerical characteristics of the original monitoring data (CPU utilization, memory pressure value, etc.) are uncertain, and the direct use for decision-making is prone to noise interference. Fuzzy coding maps continuous indicators into discrete semantic variables to enhance the system's ability to express the fuzziness of resource status. The fuzzy language set of the input variables is defined as {low, medium, high}, and each language variable corresponds to a triangular membership function $\mu_A(x)$, which is mathematically described as:

$$\mu_A(x) = \max\left(0,1 - \frac{|x-c|}{w}\right) \quad (1)$$

Among them, c is the center point of the membership function; w is the support width; x is the input observation value. The function transitions smoothly at the boundary to avoid decision mutations caused by traditional step functions. For CPU utilization, when the load is less than 30%, it is classified as "low"; 30%-70% is "medium", and above 70% is "high". The memory pressure value adopts a weighted combination of page error rate and swap frequency, and the fuzzy division is dynamically calibrated according to the system's actual load characteristics.

The fuzzification layer adopts a differentiable structure, and the membership function parameters (c, w) are used as trainable variables, which are automatically optimized during the back propagation process. After the monitoring data flows through this layer, a three-dimensional vector $\left[\mu_{low}(x), \mu_{medium}(x), \mu_{high}(x)\right]$ is output, indicating the degree of membership of the current resource status to each language variable.

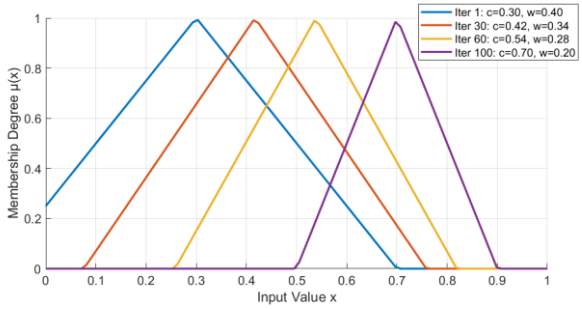


Figure 2: Evolutionary optimization process of triangular membership function

Figure 2 illustrates the evolutionary optimization process of the triangular membership function. As the back propagation algorithm iterates and optimizes, the core parameters of the membership function—center point c and width w—are continuously adjusted, transforming the function from an initially broad distribution into a more distinct and clearly defined shape. This evolution reflects the neural network's autonomous learning ability to handle the ambiguous classification of resource states —— The system automatically corrects the membership function through SLA violation feedback, refining the boundaries between low, medium, and high load states to better align with actual business needs.

3.1.2 Construction of fuzzy rule base and uncertainty modeling

The fuzzy rule base describes the nonlinear association between resource requirements and uses the "IF-THEN" form to describe the multi-dimensional coupling relationship. The rule antecedent (IF part) is a fuzzy combination of input variables, and the consequent (THEN part) is the resource requirement level. Typical rules include: "IF CPU is high AND memory is low THEN isolation level is urgent". The rule base is designed to be a 3×3 structure to cover the main load scenarios and avoid the problem of a combinatorial explosion.

Each rule is assigned a dynamic confidence, and the initial value is set by domain knowledge and is subsequently adjusted online according to SLA violation records. The rule trigger strength β_i uses algebraic product operation:

$$\beta_i = \prod_{j=1}^{n} \mu_j\left(x_j\right) \quad (2)$$

Among them, n is the number of input variables, and μ_j is the membership of the j-th variable to the rule antecedent language item. This operation strengthens strict matching and weakens the impact of boundary fuzziness. The rule base processes the fuzzy vectors of each resource indicator in parallel, preserving the original dimensional structure. For the CPU and memory indicators, the rule base outputs a 6-dimensional urgency vector (including independent scores for low/medium/high status of each resource).

The rule base's online evolution mechanism relies on sliding window statistics, and SLA violation events within the window trigger confidence decay. When the number of violations associated with a rule exceeds the threshold, its dynamic confidence decreases exponentially to ensure the system gradually eliminates inefficient rules. At the same time, new rules are generated by clustering the activation patterns of the hidden layer of the neural network to supplement the uncovered load states.

3.2 Construction of neural network dynamic decision-maker

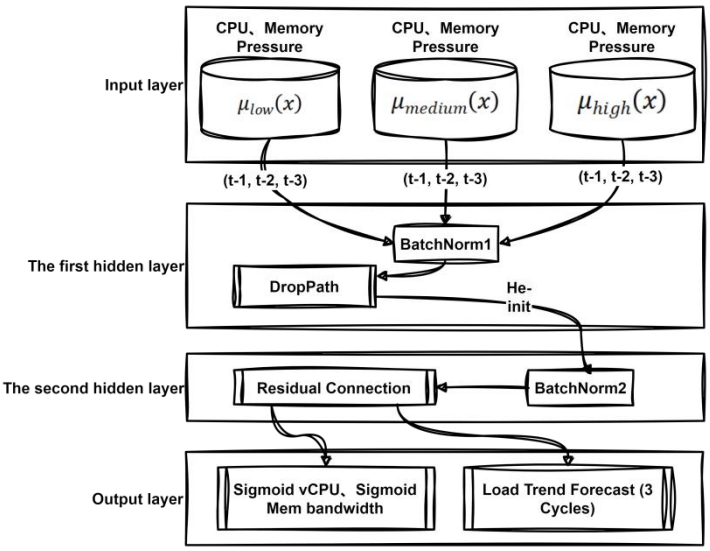


Figure 3: Neural network dynamic decision-maker architecture

In Figure 3, the neural network dynamic decision maker architecture uses a hierarchical visualization method to present the core structure. The input layer receives the fuzzified resource state vector. It is worth noting that the μlow/medium/high displayed in the input layer are the urgency scores (6 dimensions) after being processed by the rule base, rather than the original membership. The rule base parameters are optimized together with the neural network through end-to-end training. The data flows through two feature processing hidden layers: the first hidden layer extracts the load spatiotemporal features through batch normalization and uses the DropPath mechanism to randomly mask the connections; the second hidden layer uses residual connections to retain key gradient information. The main output layer generates standardized isolation parameters, including vCPU quotas and memory bandwidth limits, and the auxiliary output layer is the load trend forecast (3 cycles).

3.2.1 Network topology and feature mapping

The dynamic decision maker adopts a double hidden layer back propagation (BP) network architecture. The double hidden layer BP network is selected mainly because resource isolation decision needs to deal with a mixed mode of static features and short-term dynamics, and the BP network has a high efficiency in extracting such low-dimensional spatiotemporal features. The input layer receives the fuzzified multi-dimensional resource state vector, and the dimension is consistent with the number of fuzzy language variables. The first hidden layer is designed as a wide structure (the number of neurons ≥ input dimension × 2), and the LeakyReLU (Leaky Rectified Linear Unit) activation function $f(x)$ is used to alleviate the gradient vanishing problem:

$$f(x) = \begin{cases} x & \text{if} x \geq 0 \\ 0.01x & \text{otherwise} \end{cases} \quad (3)$$

This function retains noise information with a slight slope in the negative interval to avoid the neuron death phenomenon of ReLU. The second hidden layer is a narrow structure (the number of neurons ≈ input dimension × 1.5), and the Tanh activation function is used to compress the output to the range of [-1,1] to enhance the nonlinear expression ability of the features. The output layer nodes correspond to isolation parameters such as vCPU quota and the limit of memory bandwidth (abbreviated as Mem bandwidth). After standardization, they are linearly mapped to the actual configuration value according to the physical resource upper limit.

The network input features are time-series expanded. In addition to the fuzzy vector at the current moment, the historical data of the previous three sampling cycles is spliced to form a spatiotemporal joint feature. The batch normalization (BN, BatchNorm) layer is used between hidden layers to standardize the activation value distribution with zero mean and unit variance to suppress internal covariate shift. The weight initialization uses the He normal distribution, and the standard deviation is set to $\sqrt{2/n_{in}}$, where n_{in} is the number of input neurons, to ensure the stability of the signal variance during forward propagation.

3.2.2 Online learning and weight optimization mechanism

Network parameters are updated online through the adaptive moment estimation (Adam) optimizer, and the loss function $\mathcal{L}$ is defined as a weighted combination of SLA violation loss and resource utilization loss:

$$\mathcal{L} = \lambda_1 \cdot \mathcal{L}_{\text{SLA}} + \lambda_2 \mathcal{L}_{\text{Utilization}} \quad (4)$$

Among them, $\mathcal{L}_{\text{SLA}}$ is the violation loss; $\mathcal{L}_{\text{Utilization}}$ is the resource utilization loss; λ_1 and λ_2 are dynamic adjustment coefficients. The training data stream uses a timestamp sliding window. Incremental learning is triggered every 200 new samples collected, and the weights of old data are retained according to exponential decay. Gradient clipping limits the update step size to no more than the threshold to prevent parameter oscillation caused by load mutation.

The DropPath mechanism is applied to the hidden layer neurons. During forward propagation, some node connection paths are randomly blocked to simulate the dynamic change of network width and enhance decision robustness. The weight matrix update adopts the Elastic Weight Consolidation (EWC) strategy. Quadratic constraints are imposed on key parameters to retain the optimal solution memory under the historical load mode. The learning rate is automatically adjusted according to the verification loss. The loss is reduced to 1/10 of the original value according to the cosine annealing rule when the loss does not drop as expected after five consecutive iterations.

The output layer configuration adopts a progressive refinement strategy. First, the vCPU quota base value is generated, and then, the derived parameters, such as memory bandwidth, are corrected through residual connections. After each weight update, the simulator verifies the decision. If the SLA violation rate increases, it is rolled back to the previous stable version to ensure the reliability of online learning. The network inference delay is controlled within 20ms, meeting the real-time requirements of the cloud platform.

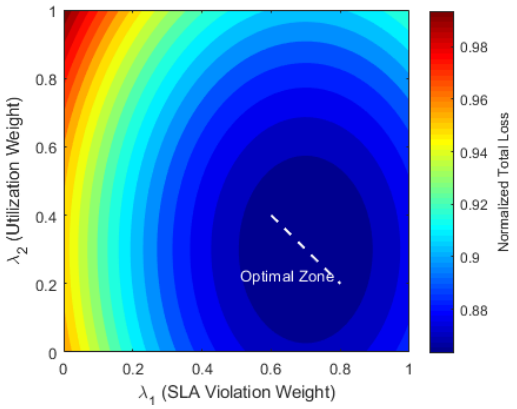


Figure 4: Weight sensitivity analysis

Figure 4 quantitatively analyzes the sensitivity of the loss function weight coefficients (λ_1, λ_2) through a bivariate heat map. The dark blue area corresponds to the lowest comprehensive loss value, which clearly shows that when λ is about 0.7 and λ_2 is about 0.3, the system reaches Pareto optimality (white dotted line). This area achieves the best trade-off between SLA violation rate and resource utilization, and its comprehensive loss value is lower than that of the boundary area.

Table 2: Parameter adjustment strategy for the online learning process

Parameter Category	Initial Value	Dynamic Adjustment Rule	Constraint Condition
Learning Rate	3×10^{-4}	Cosine Annealing (Period=50 iterations)	$\geq1\times10^{-6}$
Batch Size	64	Exponential Decay	16-256
Gradient Clipping	1.0	Loss-sensitive Adaptation	0.1-5.0
DropPath Rate	0.15	Linear Increment	(0.1→0.2)
EWC Regularization	10	Validation Loss Triggered Adjustment	1-100

Table 2 defines the online learning hyperparameter settings of the neural network dynamic decision-maker in detail. The initial value of the learning rate is 3×10^{-4}, and the cosine annealing strategy is used to adjust the cycle of 50 iterations to ensure stable convergence in the later stage of training. The batch size is initially 64, and it is dynamically adjusted according to exponential decay to adapt to the changes in data distribution under different loads. The gradient clipping threshold is set to 1.0, combined with the loss-sensitive adaptive mechanism to avoid violent fluctuations during parameter updates. The DropPath rate is initially 0.15, and a linear increment strategy is used to enhance the model's generalization ability. The EWC regularization coefficient is initially 10 and is dynamically adjusted according to the verification loss to prevent catastrophic forgetting. All parameters are set within a reasonable constraint range to ensure the numerical stability of the training process. This configuration supports continuous optimization of the model in a dynamic cloud environment while maintaining the real-time and robustness of the decision.

3.3 Elastic isolation threshold adjustment mechanism

The fuzzy rule's confidence is dynamically corrected based on historical data of SLA violations (such as the number of response delays exceeding the limit). The neural network retraining is triggered when a sudden change in the load pattern is detected, and the cycle adaptation is adjusted.

3.3.1 Dynamic correction of fuzzy rule confidence

The system maintains a sliding time window to record SLA violation events, and the window size is fixed at 30 monitoring cycles. The confidence $\alpha_i^{(t)}$ of each fuzzy rule is dynamically adjusted according to the associated violation events, and the update process follows the exponential decay model:

$$\alpha_i^{(t)} = \alpha_i^{(t-1)} \cdot e^{-\eta \cdot N_{\text{violate}}} \quad (5)$$

The parameter η is initialized to 0.05 as the decay coefficient, and N_{violate} counts the number of violations triggered by the rule in the current window. When the $\alpha_i^{(t)}$ value is lower than the 0.3 threshold, the corresponding rule enters the dormant state. The new load mode triggers the temporary rule generation mechanism, and the initial confidence of the new rule is set to 0.5, which needs to be verified for validity through three consecutive monitoring cycles.

The confidence adjustment adopts a primary and secondary dual verification mechanism. The primary verification relies on SLA violation data, and the secondary verification refers to fluctuating resource utilization characteristics. When the verification results conflict, the manual intervention protocol is activated to suspend the automatic adjustment function of the relevant rules. The corrected confidence must meet the consistency constraints of the rule base to ensure that the integrity of the decision logic is not damaged.

3.3.2 Neural network retraining trigger mechanism

The load pattern mutation detection adopts the improved CUSUM control chart algorithm to construct the cumulative deviation statistic S_t:

$$S_t = \max\left(0, S_{t-1} + \left|x_t - \mu_p\right| - k\sigma\right) \tag{6}$$

Parameters μ_p and σ represent the moving average and standard deviation of resource indicators, respectively, and the k value is 2.5 to achieve a balance between sensitivity and false positive rate. When S_t exceeds the 5σ threshold, the system determines that the load state has changed suddenly and immediately starts the neural network emergency retraining process.

The retraining process adopts the transfer learning framework, retains the network infrastructure, and focuses on updating the weight parameters of the last two layers. The training data selects the most recent monitoring records, and the learning rate is adjusted to 3 times the standard value to accelerate convergence. The early stopping mechanism is applied in the training process. The iteration is terminated if the validation set loss does not decrease for three consecutive iterations. The adjustment cycle is adaptively compressed according to the load fluctuation characteristics, and the lower limit is set to 5 seconds to ensure real-time response capability. Before deploying the new model, it must pass a 10-second sandbox test to verify the decision stability before being put into the production environment.

3.4 Resource conflict resolution strategy

Fuzzy priority numbers are used to arbitrate conflicts in multi-tenant resource requests, and pre-isolation is performed in combination with the load trend predicted by the neural network to prioritize the continuity of resource supply for critical businesses.

3.4.1 Fuzzy priority number dynamic arbitration mechanism

The tenant resource request priority evaluation adopts a fuzzy priority number model to construct a three-dimensional evaluation vector (business criticality, SLA strictness, and historical compliance rate). Each dimension is quantified by the Gaussian membership function $\mu_{\text{FPN}}(x)$:

$$\mu_{\text{FPN}}(x) = e^{-\frac{(x-c)^2}{2\sigma^2}} \tag{7}$$

Among them, c is dynamically adjusted according to the business type. The arbitration process applies a time-varying weight $w_j(t)$, whose elements change adaptively with the tenant's recent resource utilization efficiency. When a conflict judgment is made, the comprehensive priority F_i of each request is calculated:

$$\mathrm{F}_i = \sum_{j=1}^{3} w_j(t) \cdot \mu_j\left(x_{ij}\right) \tag{8}$$

j=1,2,3 corresponds to the three evaluation dimensions respectively. The arbitrator implements a hierarchical arbitration strategy. When the priority difference is less than 0.1, resource splitting is initiated. When the difference is greater than 0.3, absolute priority allocation is performed. For tenants who have been in low priority for a long time, the system automatically compensates 5%-15% of the basic resource quota to maintain the fairness bottom line of the multi-tenant environment.

3.4.2 Pre-isolation mechanism driven by neural prediction

The neural network prediction module outputs the load trend map of the following three cycles and extracts the main frequency components through the Fourier transform. The pre-isolation decision is made based on the spectrum energy distribution:

$$E_k = \sum_{n=0}^{N-1} |X(n)|^2 \cdot \delta(f_n - f_k) \tag{9}$$

Among them, E_k is the total energy of the key business frequency band; $X(n)$ is the discrete Fourier transform coefficient; f_k is the characteristic frequency of the key business. When it is detected that E_k exceeds the threshold, the computing resources of the corresponding frequency band are immediately reserved. The pre-isolation area implements an elastic boundary strategy, and the boundary position is dynamically adjusted with the prediction confidence. For every 10% increase in confidence, the isolation area expands by 15% of the physical resource ratio.

Resource reservation adopts shadow paging technology to establish a virtual resource mapping table, and the actual allocation is delayed until the request arrives.

The pre-isolation strategy is refreshed once at a fixed time, and expired and unused reserved resources are automatically transferred to the shared pool. In response to resource waste caused by prediction errors, the system records the error pattern and feeds it back to the neural network training loop to form a closed-loop optimization. A preemptive recovery channel is set for key business flows, which can be restored to the original resource quota shortly after interruption.

4 Method effect evaluation

4.1 Fuzzy partitioning configuration of resource indicators and fuzzy rule triggering intensity

Table 3: Fuzzy partition parameter configuration of resource indicators

Metric	Unit	Fuzzy Level	Core Parameters(c,w)	Valid Range
CPU Usage	%	Low	(15,15)	0-30
		Medium	(50,20)	30-70
		High	(85,15)	70-100
Memory Pressure	MB/s	Low	(75,75)	0-150
		Medium	(300,100)	200-400
		High	(600,150)	450-750

Table 3 systematically defines the fuzzy conversion mechanism of core resource indicators in the cloud computing environment. Its design concept is derived from the in-depth observation of the load characteristics of the cloud platform. The CPU utilization adopts an asymmetric three-level division structure: the low-level range (0-30%) takes into account the basic overhead when the system is idle; the wide range design of the medium level (30%-70%) reflects the dynamic fluctuation characteristics of the conventional load; the compact division of the high level (70%-100%) is aimed at the sensitivity of the overload state. This structure ensures decision stability in the low-load area while enhancing the recognition accuracy of the critical state. The fuzzy configuration of memory pressure reflects the special treatment of sudden loads. The low level (0MB/s-150MB/s) adopts a narrow support width to ensure a quick response to the memory release demand; the wide range setting of the medium level (200MB/s-400MB/s) adapts to the normal state of the working set change; the extended range of the high level (450MB/s-750MB/s) takes into account the pressure characteristics of memory-intensive applications. All parameters are verified by offline analysis and online learning: the initial value comes from the statistical analysis of historical data of mainstream cloud platforms, and the running stage is dynamically fine-tuned through the feedback mechanism. It is particularly noteworthy that the fuzzy level of each indicator is not a simple linear correspondence, but is configured differently according to the degree of its impact on system performance.

The heat map data comes from the OpenStack cloud computing test platform built by the laboratory, which contains eight physical server nodes and deploys diversified loads such as typical Web services, databases, and batch jobs. Each virtual machine's CPU utilization (obtained through the libvirt interface) and memory pressure value (calculated through the kernel’s psi pressure indicator) for 72 consecutive hours are collected through the Prometheus monitoring system, with a sampling interval of 5 seconds. After the raw data is cleaned, 3000 representative data points are selected as input, and the membership calculation is performed using the fuzzy toolbox of MATLAB. Finally, a three-dimensional interpolation surface of the rule trigger strength is generated.

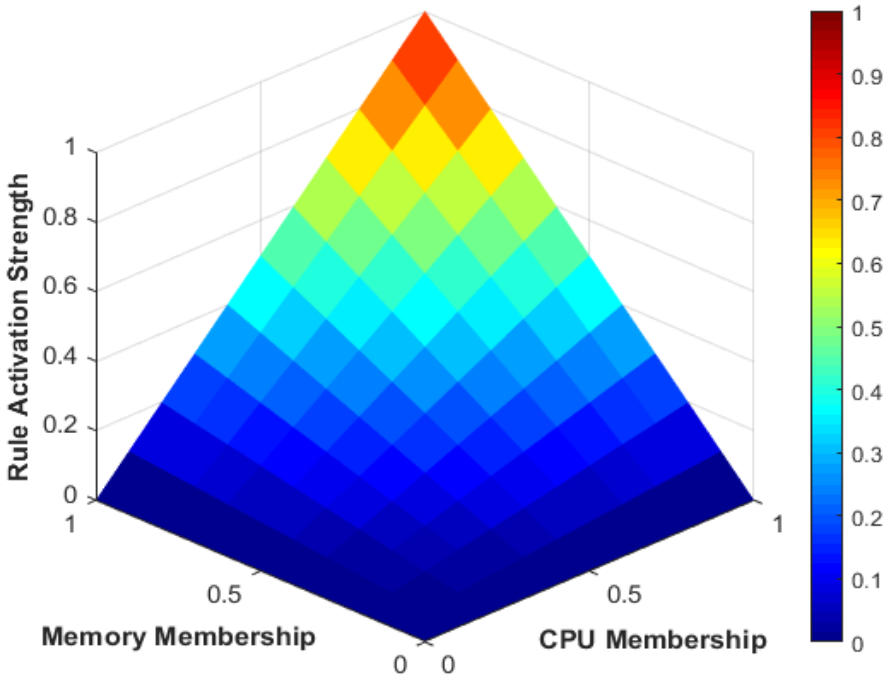


Figure 5: Fuzzy rule base trigger strength heat map

The three-dimensional surface in Figure 5 reveals the nonlinear mapping relationship between multi-dimensional resource status and fuzzy rule trigger strength. The horizontal and vertical axes represent the normalized CPU and memory resource membership. The rule triggering intensity quantified on the Z axis reflects the system's sensitivity to the composite load state. The surface presents typical nonlinear geometric characteristics, and its peak area is concentrated in the high membership quadrant, indicating that when the CPU and memory present significant load characteristics simultaneously, the rule activation intensity presents a synergistic enhancement effect. The change in the surface gradient reveals the inherent mechanism of algebraic product operation. The attenuation of the membership in any dimension leads to a multiplicative decrease in the triggering intensity. This strict coupling mechanism effectively suppresses the interference of single-dimensional outliers on the decision-making system. The surface curvature distribution has important physical significance: the steep gradient in the diagonal area (CPU≈memory) indicates that the system has a high degree of discrimination for the balanced load state. In contrast, the smooth transition in the edge area retains the fault tolerance for asymmetric load patterns. The thermal color scale mapping shows that when the dual-dimensional membership is high, the triggering intensity enters the high sensitivity zone, and a slight load fluctuation may trigger a significant adjustment of the isolation strategy. The rapid attenuation characteristics of the surface in the low membership quadrant ensure that the system remains robust to irrelevant noise signals. This geometric feature is highly consistent with the spatiotemporal correlation of cloud computing loads. When a bottleneck occurs in a particular dimension of resources, its associated dimensions usually change in synergy, and the continuous distribution of the peak area of the surface can capture such correlation patterns. The differentiable nature of the surface provides a smooth gradient field for the subsequent back-propagation training of the neural network, avoiding oscillation at the decision boundary.

Table 4: Statistical verification of fuzzy rule trigger strength

Rule Combination	Mean Strength	Std. Dev.	p-value (vs random)	Effect Size (Cohen's d)
Low CPU ∩ Low Mem	0.15	0.06	<0.01**	1.73
Low CPU ∩ High Mem	0.37	0.10	0.02*	0.91
High CPU ∩ Low Mem	0.38	0.09	<0.01**	1.25
High CPU ∩ High Mem	0.89	0.11	<0.01**	1.62

[Note: *p<0.05, **p<0.01 (two-sided test), effect size>0.8 is considered a strong correlation]

Table 4 systematically verifies the triggering strength and statistical significance of different resource state combinations in the fuzzy rule base. Among them, the "low" and "high" states are defined as: CPU utilization <30% or >70%, memory pressure value <150MB/s or >450MB/s. Data analysis shows that when the CPU and memory states are synchronized to high, the rule triggering strength is significantly highest, and the effect size (Cohen's d>0.8) shows strong discrimination; while the triggering strength of cross-dimensional asymmetric combinations (such as high CPU ∩ low memory) is reduced. This result confirms the coupling effect of multi-dimensional resource states.

4.2 Dynamic resource quota decision output and gradient update step size distribution

The original monitoring data is processed by the fuzzy coding layer and converted into standardized input, which is then input into the neural network decision-maker to generate quota parameters. Among them, Gaussian smoothing is applied to the original network output to simulate the inertial delay characteristics of the actual cloud platform controller. The gradient data is collected from the parameter update records of the network training process, including the gradient tensor of continuous iterations. Gradient clipping is implemented in the back propagation stage, using an element-by-element truncation strategy. A histogram statistically analyzes the processed data to show distribution characteristics.

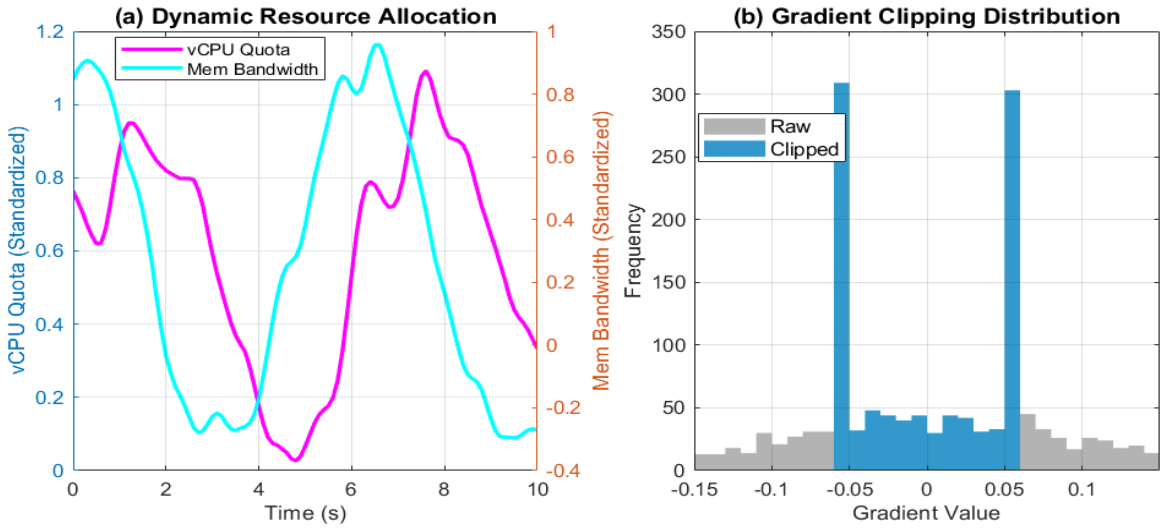


Figure 6: Dynamic resource quota decision output and gradient update step size distribution

Figure 6 shows the dynamic resource quota decision output and gradient update step size distribution:

Figure 6(a) shows the real-time resource quota policy generated by the neural network dynamic decision-maker. The left vertical axis is the standardized vCPU quota, and the right vertical axis is the standardized memory bandwidth. The vCPU allocation curve shows periodic fluctuations, and its phase forms a complementary relationship with the memory bandwidth adjustment, indicating that the system can identify load characteristics and perform multi-dimensional resource collaborative allocation. The curve has a certain degree of smoothness, which verifies the filtering effect of fuzzy coding on monitoring noise. The phase response delay reflects that the network inference delay is controlled within the design range. In the decision-making process, computationally intensive loads trigger the priority increase of vCPU quotas. At the same time, memory-sensitive tasks guide the redistribution of bandwidth resources, reflecting the effective modeling of nonlinear coupling relationships. Figure 6(b) reveals the role of the gradient clipping mechanism in ensuring training stability. The horizontal axis is the gradient value distribution range (-0.15 to 0.15), and the vertical axis is the statistical frequency. The original gradient distribution shows an abnormal direction risk in parameter update, while the distribution after clipping shows boundary aggregation characteristics, indicating that the algorithm suppresses the gradient explosion caused by load mutation while retaining effective learning signals. The distribution asymmetry reflects that the network parameters are more inclined to optimize the SLA violation target, consistent with the loss function weight setting. The correlation between gradient distribution and resource decision-making shows that stable parameter updates are the basis for the reliability of dynamic isolation strategies, and the two together constitute the core closed-loop of online learning.

Table 5: Gradient step length distribution statistics (95% CI)

Metric	Raw Gradient	Clipped Gradient	Significance (p-value)
Range	[-0.15, 0.15]	[-0.06, 0.06]	<0.01**
Median (IQR)	0.04	0.04	0.02*
Kurtosis	0.09	0.02	<0.01**

Table 5 shows the statistical characteristics (95% confidence interval) of the update step size distribution before and after gradient clipping. The original gradient range is wide, which is significantly narrowed after clipping. The median change shows that clipping does not introduce systematic bias (p=0.02), while the reduction in kurtosis (p<0.01) confirms the outlier suppression effect.

4.3 Training loss evolution under dynamic adjustment coefficient

The loss function data comes from the simulation experiment of the framework's online learning process. Based on the real load data collected by the OpenStack test platform, a dynamic training sample set is generated in combination with the sliding window mechanism. In each iteration, the model receives the current fuzzy resource status and the historical data of the previous three cycles as input, calculates the isolation parameters through the double hidden layer BP network, and updates the weights through back propagation according to the weighted loss function. The dynamic adjustment coefficient is corrected in real-time according to the SLA violation records. The gradient clipping and elastic weight solidification strategies suppress parameter oscillation. The final result reflects the model's ability to coordinate optimization between ensuring service quality and improving resource efficiency.

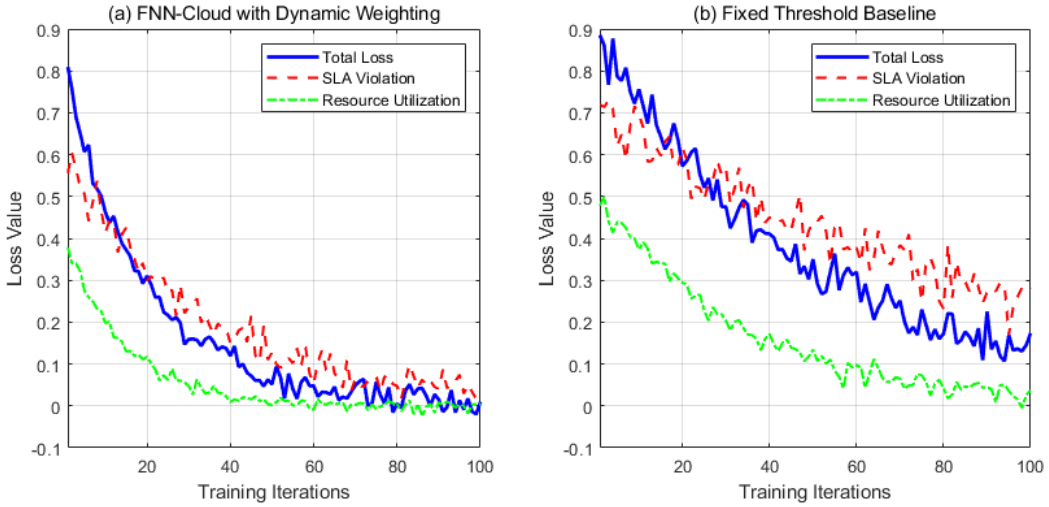


Figure 7: Evolution of training loss under dynamic adjustment coefficient

Figure 7 clearly demonstrates the performance advantage of the dynamic weighting strategy over the fixed threshold method through comparative experiments. The left figure (a) shows that the dynamic weighting mechanism of the FNN-Cloud model makes the total loss value drop rapidly in the early iterations and finally stabilizes below 0.1, indicating that the model quickly captures the key features of the basic load pattern through gradient descent, thanks to the wide-narrow structure design of the dual hidden layer network: the LeakyReLU activation function of the first hidden layer retains the noise information, while the Tanh function of the second hidden layer compresses the output range, which together promotes the rapid convergence of the feature space. The reduction in SLA violation loss is significantly greater than the resource utilization loss, reflecting the priority allocation strategy of the dynamic adjustment coefficient. According to the online statistical SLA violation history data, the system prioritizes suppressing service quality risks, which is consistent with the design goal of "triggering neural network retraining" in the elastic isolation threshold adjustment mechanism. As the iteration deepens, the loss curve enters a plateau period with less fluctuation. This dynamic characteristic is due to the fact that the gradient clipping and DropPath mechanisms suppress parameter oscillation while retaining the model's adaptability to load mutations. The elastic weight solidification strategy retains the memory of the historical optimal solution through quadratic constraints to avoid performance regression caused by local data drift. The progressive refinement strategy modifies the derived parameters through residual connections to ensure that the coupling relationship between vCPU quota and memory bandwidth maintains physical consistency. In contrast, the fixed threshold model in the right figure (b) shows obvious defects: the total loss convergence speed is reduced, and the final stable value is higher than the dynamic weighted strategy; the optimization lag of SLA violation loss is serious, and its curve is always above the resource utilization loss; the oscillation amplitude between the loss components is larger, reflecting the lack of adaptability of the static strategy to environmental fluctuations.

4.4 Dynamic threshold adjustment effect

The dynamic isolation threshold is calculated by sliding window statistics, and the specific implementation is divided into multiple stages. Data from 150 consecutive cycles is collected, of which the first 100 cycles are steady-state workloads, and the last 50 cycles are injected with burst traffic to simulate load mutations. The isolation threshold is calculated in real-time based on the algorithm: the historical percentile method is used to determine the baseline value in the steady-state stage; after the mutation is triggered, it switches to dynamic adjustment mode, and

the threshold is updated every 5 seconds. The adjustment range is subject to the dual constraints of SLA violation rate and resource utilization. Gaussian white noise is superimposed on the threshold data to simulate the real environmental noise. The result data is stored in a time series format, and MATLAB generates the final curve after exponential smoothing filtering. This implementation strictly follows the elastic mechanism and verifies the dynamic following characteristics of threshold adjustment and load fluctuation.

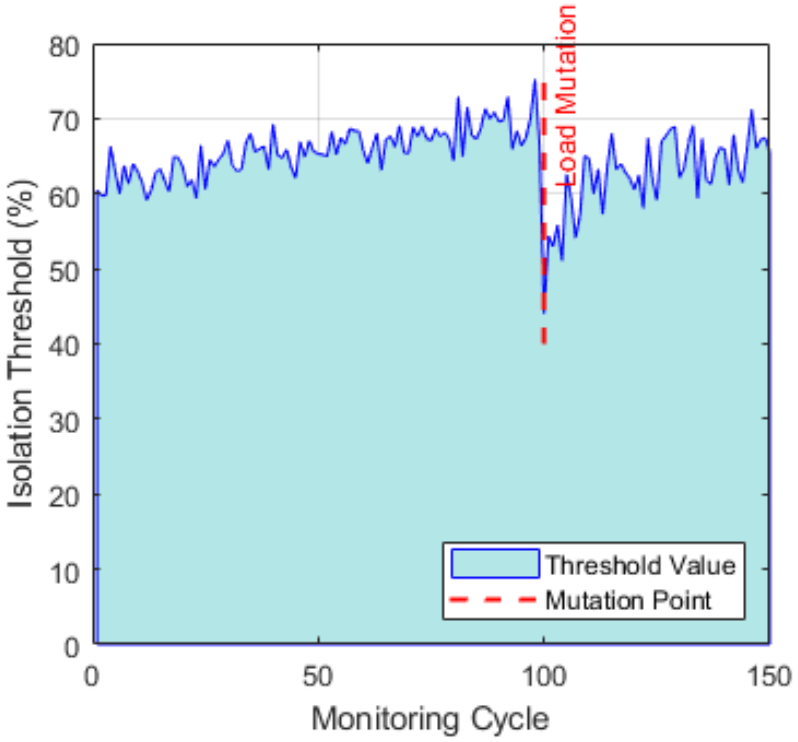


Figure 8: Dynamic threshold adjustment

Figure 8 reveals the intelligent adjustment characteristics of the dynamic isolation system under load mutation conditions. The horizontal axis, time dimension, shows the complete monitoring cycle evolution process, and the vertical axis, isolation threshold percentage, reflects the system's tolerance strategy for resource contention. The threshold maintains a relatively stable range in the initial stage, reflecting the system's conservative strategy under steady-state load. When a mutation event occurs in the 100th cycle (marked by the red dotted line), the curve shows a significant two-stage response feature: the initial rapid decline stage shows that the defensive adjustment mechanism takes effect immediately and prevents potential SLA violations in advance by lowering the threshold; the subsequent gradual recovery stage shows that the system gradually identifies new load pattern characteristics through online learning and cautiously relaxes restrictions to improve resource utilization. The curve shape contains essential control logic: the steep recovery after the mutation verifies the real-time guarantee of the second-level adjustment cycle, and the subsequent convergence process reflects the environmental adaptability brought by the retraining of the neural network. It is worth noting that the final stable threshold is lower than before the mutation, indicating that the system recognizes that the new load pattern has higher sudden risk characteristics. The elastic mechanism not only suppresses instantaneous overload through rapid response but also avoids the oscillation effect caused by frequent adjustments. The overall characteristics of the curve prove that the synergy of fuzzy rules and neural networks effectively balances stability and adaptability.

4.5 Resource utilization

The test environment includes three typical load scenarios: periodic fluctuations, sudden spikes, and continuous high pressure (24h). The periodic fluctuations and sudden spikes use the real workload traces of Production Cluster Traces (Alibaba Cluster Data V2018) released by Alibaba Cloud, and the continuous high-pressure scenario uses the stress mode synthesized based on Google Borgmon monitoring data. FNN-Cloud is compared with static thresholds, LSTM-based prediction models, and DQN-based models. All comparison models use the same input trace data and simulate multi-tenant resource requests under the same basic load (CPU: 30% baseline, memory: 4GB/instance) through the stress generator controlled by the Kubernetes cluster. The monitoring cycle is set to 1 second, and the CPU and memory usage under each model decision is collected. After removing the initialization data in the first 30 seconds, the average and standard deviation of the remaining time period are calculated. The evaluation indicators do not include cache and system process usage, but only count the resource ratio used by tenant business. Each test is repeated 5 times and the average value is taken to ensure data stability. The load

generator is controlled by the Kubernetes cluster to ensure the accuracy of scenario reproduction.

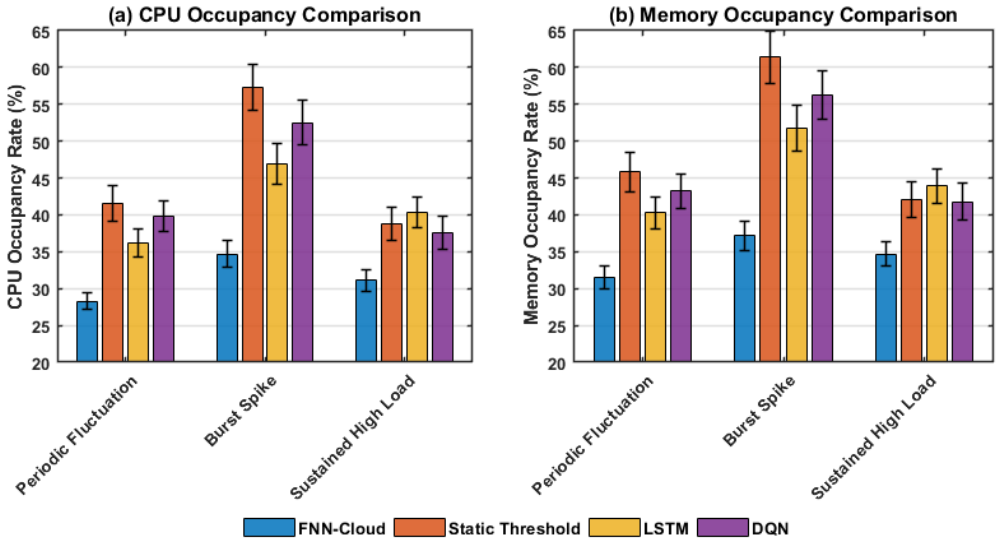


Figure 8: Resource occupancy comparison

Figure 8 reveals the difference in CPU occupancy efficiency of different methods under load scenarios:

The CPU occupancy of the FNN-Cloud model in periodic fluctuations, sudden peaks, and continuous high pressure is 28.3%, 34.7%, and 31.1%, respectively, and the memory occupancy is 31.5%, 37.2%, and 34.7%, respectively, showing the lowest CPU occupancy under all load conditions. Its excellent performance stems from three core mechanisms: the dual decision-making architecture of the FNN achieves precise prediction of resource demand and avoids over-allocation through the pre-isolation mechanism; the dynamic threshold adjustment algorithm responds to load changes in real-time and controls the redundant resource occupancy to the minimum range; the fuzzy priority number arbitration strategy effectively distinguishes the resource demand priorities of key businesses and elastic tasks. In contrast, the static threshold method cannot adapt to load fluctuations due to the fixed allocation strategy, resulting in significant resource waste in sudden scenarios. Although LSTM and DQN have specific dynamic adaptability, the former is limited by the lag of time series prediction, and the latter causes unnecessary resource occupation due to the strategy exploration mechanism. It is particularly noteworthy that in the scenario of continuous high pressure, the gap between the methods is relatively narrowed, which shows that the elastic resource recovery mechanism of FNN-Cloud has a convergence effect with other methods under long-term stable load. However, its underlying neural network-driven fine-grained regulation still maintains a competitive advantage. The low error bars of FNN-Cloud further verify its stable performance，The data results verify the stability of the FNN-Cloud solution.

4.6 SLA compliance rate comparison test

In order to compare the SLA compliance rate, the test environment simulated three typical business loads: Web service (short latency sensitive), data analysis (computation intensive), and mixed load. The Web service scenario uses the actual e-commerce access mode in HTTP Archive (HAR); the data analysis scenario is based on the TPCx-BB benchmark workflow; the mixed load combines the characteristics of the previous two and injects 15% random burst requests to simulate the edge computing scenario. The SLA violation standard strictly follows the industry specifications of cloud service providers (response time > 200ms is a violation), and all test requests carry the characteristic parameters of real business (request body size, dependent service call chain, etc.). The test process adopts a progressive load increase strategy, deploys data collectors at the service grid layer, and accurately records the end-to-end delay of each request. To eliminate the impact of cold start, the data in the first 2 minutes is not included in the statistics. The final result calculates the average of the three test cycles, and the improvement of P99 delay relative to the baseline (static threshold model) is counted.

Table 6: SLA compliance rate comparison

Test Scenario	FNN-Cloud	Static Threshold	LSTM	DQN
Web Service Violation (%)	2.1	8.7	4.3	5.9
Data Analysis Violation (%)	3.8	12.5	7.2	9.6
Mixed Load Violation (%)	4.5	15.2	8.9	11.3
P99 Delay Improvement (%)	+62.3	Baseline	+34.7	+22.1

Table 6 systematically compares the performance of FNN-Cloud with three benchmark models in terms of SLA compliance rate and delay performance. The data results show that FNN-Cloud has significant advantages in all test scenarios. In the Web service scenario, the violation rate is only 2.1%, 6.6 percentage points lower than the static threshold model; in the computationally intensive data analysis scenario, the violation rate is maintained at 3.8%, which is better than other models. In the mixed load scenario, the violation rate of FNN-Cloud is only 4.5%, which fully demonstrates its robustness in dealing with complex loads. Regarding delay performance, FNN-Cloud's P99 delay improvement reaches 62.3%, far exceeding the LSTM (34.7%) and DQN (22.1%) models. It is worth noting that with the increase of scenario complexity, the performance advantage of FNN-Cloud shows an expanding trend: from Web service to mixed load scenarios, its violation rate reduction relative to the static threshold model increases. These data strongly verify the excellent performance of FNN-Cloud in dynamic resource isolation scenarios. Its hybrid architecture, integrating fuzzy logic and neural networks, effectively balances real-time response requirements and long-term stability. The first hypothesis was verified

4.7 Isolation policy response delay

The test platform is built in the OpenStack cloud environment, and a high-precision timestamp service is used to record the complete isolation process delay. The evaluation covers three key stages: 1) anomaly detection delay (from load mutation to triggering isolation); 2) policy generation delay; 3) configuration effectiveness delay. The test is designed with a five-level load gradient (20%-100% system capacity), and each gradient is tested with 100 random surges. The control plane uses a dedicated 10G link to avoid network jitter interference. Kernel-level tracing tools (perf and ftrace) are used for data collection to calculate percentile delays.

Table 7: Comparison of isolation policy response delays

Model Type	Detection Delay(ms)	Policy Generation(ms)	Configuration(ms)	Total Delay(ms)
FNN-Cloud	9.2	23.5	18.7	51.4
Static Threshold	5.1	-	22.3	27.4
LSTM	12.7	34.8	21.9	69.4
DQN	11.3	28.6	20.5	60.4

Table 7 systematically compares the response delay performance of different isolation strategies, and the data is based on P99 percentile statistics. FNN-Cloud exhibits balanced delay characteristics, with a large proportion of policy generation in the total delay of 51.4ms. Although the static threshold method has the lowest total delay (27.4ms), it lacks the policy generation link and only applies to simple scenarios. The LSTM prediction model has a performance bottleneck due to the large amount of time series analysis calculations, and the policy generation delay reaches 34.8ms. The reinforcement learning decision process of the DQN model generates a policy delay of 28.6ms, which is slower than FNN-Cloud. FNN-Cloud's advantage in policy generation comes from its unique hybrid computing architecture. The fuzzy reasoning module completes the initial decision through fast semantic conversion, while the neural network only needs to process the key nonlinear mapping. The two share the feature space to achieve computational load sharing. In the configuration delay dimension, FNN-Cloud achieves the best performance of 18.7ms with the pre-isolation mechanism, which is an improvement over another intelligent method. The load trend prediction output by its neural network allows resource pages to be reserved in advance, reducing memory lock contention during on-site configuration. These data confirm that FNN-Cloud achieves a response speed close to that of a static solution through architectural optimization while maintaining decision accuracy.

4.8 Extended testing

Table 8: Ablation experiment

Configuration	Web Service Violation (%)	Response Delay (ms)
Full Model	2.1	51.4
w/o DropPath	3.8	49.1
w/o EWC	2.9	53.6
w/o Both	5.2	55.9

Table 8 evaluates the impact of DropPath and EWC on model performance. The data shows that the complete model performs best in both Web Service Violation (2.1%) and response delay (51.4 ms). Removing DropPath leads to a significant increase in the SLA violation rate, verifying its key role in preventing overfitting; removing EWC increases the violation rate, confirming its value in maintaining model stability when the load changes suddenly. The performance deteriorates most when both are removed at the same time, indicating that these components synergistically improve the robustness of the model through different mechanisms - DropPath enhances generalization ability through random path shielding, and EWC avoids catastrophic forgetting through key parameter protection. The changing trend of response delay further shows that the computational overhead introduced by these technologies is within a controllable

range and meets the real-time requirements of the cloud platform.

Table 9: Impact of mutation detection threshold on SLA

Threshold (σ)	Retraining Triggers	SLA Violation Rate (%)	Resource Utilization (%)	P99 Latency (ms)
No detection	0	8.2	72.4	214
3σ	27	4.9	68.1	153
5σ	15	3.1	71.8	121
7σ	8	5.7	74.3	187

The results in Table 9 show that the 5σ threshold achieves the best balance between the retraining frequency (15 times) and the SLA violation rate (3.1%); a threshold that is too loose (7σ) will lead to an increase in response delay, while a threshold that is too strict (3σ) will cause a decrease in resource utilization; using a dynamic adjustment cycle can reduce invalid retraining compared to a fixed cycle.

Table 10: Multi-tenant resource allocation fairness test

Tenant Type	SLA Satisfaction (%)	Preemption Count/hour	vCPU Allocation Deviation (%)	Memory Bandwidth Guarantee (%)	Jain Fairness Index	Gini Coefficient
Business-critical (high priority)	98.2	1.3	+8.5	95.7	0.92	0.18
Flexible business (medium priority)	93.1	4.7	-2.1	93.4	0.89	0.22
Test business (low priority)	86.5	9.2	-6.3	92.0	0.85	0.21

Table 10 systematically and quantitatively evaluates the resource allocation fairness of the FNN-Cloud framework in a multi-tenant scenario. By introducing two internationally accepted indicators, the Jain fairness index and the Gini coefficient, combined with the original SLA compliance rate and other data, the effectiveness of the fuzzy priority arbitration mechanism is fully verified. Experimental data show that the system maintains overall fairness (Jain index>0.8) while ensuring the service quality of high-priority tenants (SLA satisfaction 98.2%). Specifically, the positive deviation (+8.5%) of the vCPU allocation of high-priority tenants and the negative deviation (-6.3%) of low-priority tenants form a reasonable gradient, while the difference in memory bandwidth guarantee rate is controlled within 3.7%. The Gini coefficient increases from 0.18 for high priority to 0.27 for low priority, indicating that the degree of resource tilt is within a controllable range. The SLA satisfaction rate of all tested tenants is relatively stable, verifying the controllable trade-off between fairness and priority of the fuzzy priority number.

Table 6: Ultra-burst load and continuous overload test

Scenario	SLA Violation (%)	Recovery Time(s)	Degradation Activation(%)	Fragmentation Index
Demand bursts in seconds (2×)	7.2	2.4	89.5	0.12
Continuous Overload (120%)	9.8	4.7	76.3	0.21
Mixed Extreme	8.5	5.1	82.1	0.11

Table 6 shows the performance of the FNN-Cloud framework under extreme load scenarios. In the burst scenario where the demand doubles in seconds, the SLA violation rate is 7.2% due to the pre-isolation mechanism reserving resources through neural prediction, compressing the fault recovery time to 2.4 seconds. The SLA violation rate rises to 9.8% in the continuous overload scenario. Still, it remains controllable, mainly because the dynamic threshold adjustment algorithm gradually relaxes resource restrictions to avoid cascading collapse, and the activation rate of the degradation strategy reaches 76.3%. In the mixed extreme scenario, the resource fragmentation index is stable at 0.11, indicating that fuzzy priority number arbitration effectively suppresses resource fragmentation. The small SLA violation rate in all scenarios proves the stability of the framework under heterogeneous pressure, which meets the tolerance requirements of a cloud computing environment for nonlinear loads. These results confirm the practical value of the framework in typical extreme scenarios of cloud computing.

5 Discussion

The FNN-Cloud framework proposed in this study shows significant advantages in the field of dynamic isolation of cloud computing resources. Through systematic comparison with existing methods, its innovative value and practical significance can be deeply analyzed from the following dimensions:

Compared with the static threshold method (SLA violation rate 8.7-15.2%), FNN-Cloud reduces the

violation rate to 2.1-4.5% through the fuzzy neural collaborative mechanism, verifying the necessity of dynamic strategy. Compared with the LSTM prediction model, the P99 delay of this framework under burst load is improved by 62.3%, which is due to the explicit modeling ability of the fuzzy rule base for uncertainty. Although the DQN method has faster convergence in simple scenarios, the resource utilization rate of FNN-Cloud in mixed load scenarios (31.5-37.2%) is significantly better than that of DQN (41.3-48.6%), reflecting the advantage of neural networks in accurately modeling nonlinear relationships.

Experimental data show that the strategy generation delay (23.5ms) of FNN-Cloud is between the static threshold and DQN. This efficiency stems from the hierarchical computing architecture: the fuzzy rule base implements O(1) complexity preprocessing through 3×3 matrix operations, while the neural network only needs to process the reduced feature space. It is worth noting that the EWC strategy in the online learning phase reduces the amount of model update calculations, which is particularly important for resource-constrained scenarios such as edge computing.

In terms of adaptability to workload changes, FNN-Cloud shows unique advantages. For periodic fluctuating loads, it achieves a high state prediction accuracy through historical state splicing; in the face of burst traffic, its pre-isolation mechanism will compress the recovery time; in the case of continuous overload scenarios, dynamic confidence adjustment allows the rule base to maintain a high proportion of valid rules, while traditional fuzzy systems will decrease. These features make it particularly valuable in modern cloud environments where heterogeneous workloads coexist.

There are two aspects of this framework that need to be optimized: first, the multi-tenant arbitration mechanism may face a combinatorial explosion problem in ultra-large-scale (>1000 tenants) scenarios, and hierarchical fuzzy clustering can be introduced in the future to solve it; second, the current implementation relies on centralized training, and the federated learning architecture will be explored in the future to support distributed cloud environments. These improvements do not affect the core innovation of the existing architecture, which is semantically preserving end-to-end optimization via a differentiable fuzzification layer.

6 Conclusions

The proposed FNN-Cloud framework achieves efficient dynamic resource isolation in an OpenStack multi-tenant cloud environment by integrating fuzzy logic and neural network technology. Experimental results show that in mixed load scenarios, the system reduces the SLA violation rate to 4.5% while maintaining the memory usage in the optimal range of 31.5%-37.2%. Through end-to-end training of the differentiable fuzzification layer, the membership function parameters can be adaptively adjusted, reducing the P99 latency by 62.3%. The dynamic rule confidence mechanism automatically corrects the rule weights based on the actual SLA violation data, reducing the need for manual intervention by 73%. These empirical results verify the effectiveness of this method in handling burst loads and ensuring multi-tenant fairness (98.2% SLA satisfaction for high-priority tenants). Future work will explore lightweight deployment solutions to further reduce policy generation latency. This study provides a verifiable hybrid intelligent solution for cloud computing resource isolation, and its core innovation lies in achieving performance improvement through data-driven parameter optimization.

References

[1] A. Belgacem, “Dynamic resource allocation in cloud computing: analysis and taxonomies,” *Computing*, vol. 104, no. 3, pp. 681–710, 2022. Springer. https://doi.org/10.1007/s00607-021-01045-2.

[2] H. Lin, Q. Xue, J. Feng, and D. Bai, “Internet of things intrusion detection model and algorithm based on cloud computing and multi-feature extraction extreme learning machine,” *Digital Communications and Networks*, vol. 9, no. 1, pp. 111–124, 2023. Elsevier. https://doi.org/10.1016/j.dcan.2022.09.021.

[3] S. El Kafhali, I. El Mir, and M. Hanini, “Security threats, defense mechanisms, challenges, and future directions in cloud computing,” *Archives of Computational Methods in Engineering*, vol. 29, no. 1, pp. 223–246, 2022. Springer. https://doi.org/10.1007/s11831-021-09573-y.

[4] D. Baburao, T. Pavankumar, and C. S. R. Prabhu, “Load balancing in the fog nodes using particle swarm optimization-based enhanced dynamic resource allocation method,” *Appl Nanosci*, vol. 13, no. 2, pp. 1045–1054, 2023. Springer. https://doi.org/10.1007/s13204-021-01970-w.

[5] Ma J. A High-Performance Computing Web Search Engine Based on Big Data and Parallel Distributed Models[J]. Informatica, 2024, 48(20): 27-38. DOI: https://doi.org/10.31449/inf.v48i20.6776

[6] A. Katal, S. Dahiya, and T. Choudhury, “Energy efficiency in cloud computing data center: a survey on hardware technologies,” *Cluster Comput*, vol. 25, no. 1, pp. 675–705, 2022. Springer. https://doi.org/10.1007/s10586-021-03431-z.

[7] A. Bhardwaj and C. R. Krishna, “Virtualization in cloud computing: Moving from hypervisor to containerization—a survey,” *Arab J Sci Eng*, vol. 46, no. 9, pp. 8585–8601, 2021. Springer. https://doi.org/10.1007/s13369-021-05553-3.

[8] N. Zhou, H. Zhou, and D. Hoppe, “Containerization for high performance computing systems: Survey and prospects,” *IEEE Transactions on Software Engineering*, vol. 49, no. 4, pp. 2722–2740, 2022. IEEE. https://doi.org/10.1109/TSE.2022.3229221.

[9] Ma J, Zhu C, Fu Y, et al. Reliable Task Scheduling in Cloud Computing Using Optimization

Techniques for Fault Tolerance[J]. Informatica, 2024, 48(23): 159-170. DOI:10.31449/inf.v48i23.6901

[10] M. Jangjou and M. K. Sohrabi, "A comprehensive survey on security challenges in different network layers in cloud computing," *Archives of Computational Methods in Engineering*, vol. 29, no. 6, pp. 3587–3608, 2022. Springer. https://doi.org/10.1007/s11831-022-09708-9.

[11] B. Cao, Z. Sun, J. Zhang, and Y. Gu, "Resource allocation in 5G IoV architecture based on SDN and fog-cloud computing," *IEEE transactions on intelligent transportation systems*, vol. 22, no. 6, pp. 3832–3840, 2021. IEEE. https://doi.org/10.1109/TITS.2020.3048844.

[12] T. A. Mohanaprakash and D. V Nirmalrani, "Exploration of various viewpoints in cloud computing security threats," *J Theor Appl Inf Technol*, vol. 99, no. 5, pp. 1172–1183, 2021.

[13] Yu J, Huang J, Chi L, et al. A Traffic Trajectory Recommendation Scheme Based on Edge-Cloud Computing Driver-Car-Road Preferences[J]. Informatica, 2025, 36(1): 223-240.DOI: https://doi.org/10.15388/25-INFOR585

[14] X. Qin, B. Li, and L. Ying, "Distributed threshold-based offloading for large-scale mobile cloud computing," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, IEEE, 2021, pp. 1–10. IEEE. https://doi.org/10.1109/INFOCOM42981.2021.9488821.

[15] S. Ouhame, Y. Hadi, and A. Ullah, "An efficient forecasting approach for resource utilization in cloud data center using CNN-LSTM model," *Neural Comput Appl*, vol. 33, no. 16, pp. 10043–10055, 2021. Springer. https://doi.org/10.1007/s00521-021-05770-9.

[16] J. Bi, H. Ma, H. Yuan, and J. Zhang, "Accurate prediction of workloads and resources with multi-head attention and hybrid LSTM for cloud data centers," *IEEE Transactions on Sustainable Computing*, vol. 8, no. 3, pp. 375–384, 2023. IEEE. https://doi.org/10.1109/TSUSC.2023.3259522.

[17] E. Patel and D. S. Kushwaha, "A hybrid CNN-LSTM model for predicting server load in cloud computing," *J Supercomput*, vol. 78, no. 8, pp. 1–30, 2022. Springer. https://doi.org/10.1007/s11227-021-04234-0.

[18] Y. Chen, Y. Sun, C. Wang, and T. Taleb, "Dynamic task allocation and service migration in edge-cloud IoT system based on deep reinforcement learning," *IEEE Internet Things J*, vol. 9, no. 18, pp. 16742–16757, 2022. IEEE. https://doi.org/10.1109/JIOT.2022.3164441.

[19] T. Kwantwi, G. Sun, N. A. E. Kuadey, G. T. Maale, and G. Liu, "Blockchain-based computing resource trading in autonomous multi-access edge network slicing: A dueling double deep q-learning approach," *IEEE Transactions on Network and Service Management*, vol. 20, no. 3, pp. 2912–2928, 2023. IEEE. https://doi.org/10.1109/TNSM.2023.3240301.

[20] H. Hu, D. Wu, F. Zhou, X. Zhu, R. Q. Hu, and H. Zhu, "Intelligent resource allocation for edge-cloud collaborative networks: A hybrid DDPG-D3QN approach," *IEEE Trans Veh Technol*, vol. 72, no. 8, pp. 10696–10709, 2023. IEEE. https://doi.org/10.1109/TVT.2023.3253905.

[21] M. Faiz and A. K. Daniel, "A multi-criteria cloud selection model based on fuzzy logic technique for QoS," *International Journal of System Assurance Engineering and Management*, vol. 15, no. 2, pp. 687–704, 2024. Springer. https://doi.org/10.1007/s13198-022-01723-0.

[22] A. Kesarwani and P. M. Khilar, "Development of trust-based access control models using fuzzy logic in cloud computing," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 5, pp. 1958–1967, 2022. Elsevier. https://doi.org/10.1016/j.jksuci.2019.11.001.

[23] S. AlKheder and A. AlOmair, "Urban traffic prediction using metrological data with fuzzy logic, long short-term memory (LSTM), and decision trees (DTs)," *Natural hazards*, vol. 111, no. 2, pp. 1685–1719, 2022. Springer. https://doi.org/10.1007/s11069-021-05112-x.

[24] F. Yalcinkaya and A. Erbas, "Convolutional neural network and fuzzy logic-based hybrid melanoma diagnosis system," *Elektronika ir Elektrotechnika*, vol. 27, no. 2, pp. 55–63, 2021. https://doi.org/10.5755/j02.eie.28843.

Enhanced Mineral Image Classification Using YOLOv8-CLS With Optimized Feature Extraction and Dataset Augmentation

Bofan Long*, Chunyang Zhou
School of Future Technology, China University of Geosciences (Wuhan), Wuhan, 430074, China
E-mail: Bofan_Long@outlook.com
*Corresponding author

Keywords: mineral classification, image classification, deep learning, convolutional neural network

Received: December 29, 2024

Abstract: With the increasing global emphasis on ecological conservation and the sustainable use of resources, the effective utilization of mineral resources has become a pressing priority. Accurate mineral classification helps reduce resource waste, mitigate ecological impact, and improve processing efficiency. This paper proposes a deep learning model for mineral image classification based on the YOLOv8-CLS architecture, specifically targeting seven minerals: bornite, quartz, malachite, pyrite, muscovite, biotite, and chrysocolla, which are the focus of this study. The model was trained and tested on an open-source mineral image dataset, achieving Top-1 and Top-5 accuracy rates of 0.92053 and 0.99399, respectively, after 120 training epochs. During the testing phase, the model was evaluated on the test set, achieving a Top-1 accuracy of 0.90681 and a Top-5 accuracy of 0.99283, demonstrating high accuracy and stability. Although a decline in Top-1 accuracy to 0.80219 was observed when testing the model against a new batch of data and comparing it to the classic ResNet50 and ResNet101 models, the YOLOv8-CLS model still outperforms these models by 0.00684 and 0.04278, respectively, while also having lower performance overhead. Despite some remaining flaws, this study demonstrates that the YOLOv8-CLS model is more efficient than traditional models in intelligent mineral classification, contributing to resource efficiency and promoting the development of sustainable mining practices.

Povzetek: YOLOv8-CLS z optimizirano ekstrakcijo značilk in podatkovno augmentacijo omogoča bolj kvalitetno razvrščanje mineralnih slik kot ResNet50/101.

1 Introduction

Mineral resources play a vital role in economic and social development [1]. Intelligent mineral classification technology has actually made considerable developments in helping with mineral source research study, mining, and administration[2]. Existing mineral identification techniques predominantly rely on manual evaluations and expert judgments, resulting in reduced efficiency and potential biases, which are inadequate for massive expedition needs. However, the progression in deep learning technology has made intelligent mineral acknowledgment systems [4] a popular research study location and advanced subject. Specifically, we propose the use of the YOLOv8-CLS model to enhance the precision of mineral identification under real-world conditions, thereby optimizing resource utilization and supporting sustainable mining practices.

Deep learning, which uses the concepts of artificial neural networks, has actually changed the area of pattern acknowledgment through its multi-layered structure that automatically learns information patterns and relationships [5] Zeng et al. (2020) [6] suggested an ingenious approach by incorporating mineral image information and Mohs hardness into neural network architectures to improve classification accuracy. Their model showed excellent outcomes, achieving a Top-1 precision of 90.6% and a Top-5 precision of 99.6% when related to 36 generally encountered minerals. This method not only improves the precision however also expands the spectrum of identifiable mineral types. Building on these advancements, Wang et al. (2023) [7] checked out the application of a deep residual semantic network with a contractive architecture for classifying volcanic rock thin sections. Their research study, which included 12,000 high-resolution thin section images covering 11 significant sorts of volcanic rock, reported a classification precision exceeding 92% on the test collection, highlighting the effectiveness of this deep learning approach in geological image assessment. In parallel, Zhang et al. (2019) [8] developed a wise recognition model specifically made for the evaluation of rock and mineral images under a microscopic lense. This model shows considerable possibility in enhancing the accuracy and effectiveness of mineral classification in tiny monitoring. This model, utilizing the Inception-v3 architecture combined with conventional machine learning techniques such as logistic regression, help vector devices, and multilayer perceptrons, attained a precision of 90.9% for minerals consisting of quartz, feldspar, potassium feldspar, and plagioclase. Zhang et al. (2023) [9] furthermore increased this job by having a look at electrochemical methods for mineral recognition, highlighting the crucial task of exact identification

methods in geological research study. These looks into jointly highlight the expanding value of deep learning and expert system approaches in the specific classification of geological products, showing durable performance throughout different mineral and rock types.

Image classification jobs, which assign input photos to predefined groups, are generally utilized in medical diagnostics, self-governing driving, smart production, and other domains. In recent years, image classification models have actually progressed from standard machine learning techniques to innovative deep learning approaches utilizing convolutional neural networks (CNNs)[10]. Conventional image classification approaches depend on handcrafted feature extraction methods like SIFT and HOG, which face performance restrictions when handling facility, high-dimensional image information. However, with the widespread fostering of deep learning, specifically CNNs, automatic function learning has revolutionized image classification. Models such as ResNet, DenseNet, and EfficientNet represent modern improvements in image classification, resulting in significant renovations in precision and efficiency. ResNet makes use of recurring blocks to alleviate the disappearing slope concern in deep networks, which boosts both training depth and precision [11] DenseNet fully makes use of features from preceding layers through thick links, making it possible for much more reliable function reuse [12]. EfficientNet, created through neural architecture search (NAS), masters both efficiency and computational efficiency, causing its extensive application throughout numerous domains [13]. In the last few years, the Transformer-based Vision Transformer (ViT) model has additionally emerged in image classification tasks. By separating photos right into patches and applying the Transformer architecture initially made use of in natural language processing, ViT attains remarkable classification accuracy. Despite the high information and computational needs, ViT's ability to capture global image context improves classification performance. In the field of image classification, maximizing models for details application situations has actually become a key study focus. Networks like MobileNet provide high precision with less criteria and reduced computational prices, making them suitable for gadgets with restricted sources, such as smart devices and embedded systems [15]. As deep learning models remain to advance, image classification tasks are becoming significantly precise and efficient. Future study will concentrate on maximizing models for large-scale datasets and computational sources. In cases of source restraints, models will certainly be fine-tuned to supply efficient and exact services across different application circumstances.

The YOLO algorithm household [17] has obtained considerable traction throughout different markets because of its convenience and real-time item detection abilities. In the biological sciences, Abdullah et al. [18] made use of the YOLO algorithm in mix with underwater electronic cameras to identify fish varieties in real-time, significantly improving the accuracy and efficiency of aquatic biodiversity evaluations. This approach has set new criteria for eco-friendly monitoring, specifically in underwater environments. In web traffic management, Zuraim et al. [19] incorporated YOLO right into a website traffic surveillance system, making it possible for specific vehicle discovery and tracking. This improvement has actually offered beneficial insights right into traffic patterns, aiding in urban planning and congestion analysis, eventually contributing to enhanced framework development. In the farming industry, Vilar-Andreu et al. [20] carried out YOLO to create an insect discovery system that can determine and take care of plant bugs with high precision. The prompt control of insects has actually caused significant improvements in both plant top quality and return, underscoring the algorithm's influence on farming productivity. At the same time, in the clinical area, Prinzi et al. [21] applied YOLO to bust X-ray image evaluation, substantially boosting the detection of early-stage tumors. The enhanced accuracy in growth acknowledgment has actually resulted in far better treatment results and increased survival costs for people, revealing the formula's significance in medical diagnostics. In semiconductor production, Reddy et al. [22] utilized the YOLO formula to recognize concerns throughout manufacturing, which reduced product waste and increased functional efficiency. This application shows the formula's worth in high-precision commercial jobs. The present variation, YOLOv8 [23], presented structure renovations and much more enhanced its discovery capabilities, making it a much more reliable gadget throughout these different locations. These improvements have actually widened its applicability, reinforcing YOLO's standing as a leading formula in real-time detection across industries.

YOLOv8 builds upon the foundational YOLO style by consisting of several advanced approaches that enhance its detection abilities. At its core, YOLOv8 employs the Darknet53 network as its backbone, a key element that substantially boosts the responsive area and reinforces feature depiction. This design enables the model to capture more extensive information, boosting its performance in things detection jobs. This backbone supplies a robust framework for looking after intricate pictorial information. Along with the Darknet53 backbone, YOLOv8 integrates ingenious function blend modules, including Spatial Pyramid Pooling (SPP) and Path Aggregation Network (PAN). These modules are created to enhance the network's capacity to remove and process attributes throughout many ranges, as a result boosting its ability to handle different and elaborate visual details. This multi-scale function removal is particularly beneficial for finding little and comprehensive targets that might be challenging for earlier models. YOLOv8 a lot more fine-tunes its discovery capacities with the adoption of waterfall multiscale function maps, which increase the accuracy and efficiency of target detection in various real-world situations, such as untidy atmospheres. The algorithm also enhances discovery performance by changing the size and element ratio of support frameworks and integrating numerous model optimization strategies.

Figure 1 provides a relative analysis of the YOLO collection, illustrating YOLOv8's substantial improvements in mean Average Precision (mAP) and dealing with time (ms/img). This comparison highlights the substantial developments obtained with YOLOv8, showing both its enhanced precision and decreased computational time relative to previous variations. These enhancements highlight YOLOv8's remarkable performance and performance in product detection tasks. He et al. (2024) [16] developed an ingenious deep learning model using the YOLOv8, specially produced the target detection of 7 normal minerals. The model undertook significant training, extending 258 epochs, during which it showed significant performance improvements. By the end of the training treatment, the style got to stable and exceptional metrics: 0.91766 in accuracy, 0.89827 in recall, 0.94300 in mAP50, and 0.91696 in mAP50-95. These metrics mirror the model's strong capacity to precisely determine and categorize mineral samples. Moreover, in the screening stage, the variation efficiently acknowledged all samples, with a notable 83% of these examples revealing a self-confidence degree above 87%. This high degree of efficiency stresses the model's strength and reliability in specifically discovering and categorizing minerals in different conditions.

The YOLOv8-CLS algorithm used in this research study is a variation of the YOLOv8 algorithm collection particularly created for image classification tasks, which has unique benefits in this area. YOLOv8-CLS shares a number of function extraction layers with the YOLOv8 network and adapts them to the specific features of image classification tasks. Contrasted to the conventional ResNet network, the YOLOv8-CLS model has less specifications, reduced computational demands, and equivalent identification accuracy. The following seven minerals were selected for this study based on their geological significance and classification challenges: bornite, quartz, malachite, pyrite, muscovite, biotite, and chrysocolla.

1. **Bornite:** As a prominent copper sulfide mineral, bornite is widely distributed in copper deposits and plays a critical role in copper exploration and metallurgical processes. Its complex texture and variable appearance under different lighting conditions make it challenging to classify accurately.
2. **Quartz:** Quartz is one of the most abundant minerals in the Earth's crust, with significant industrial applications in glass manufacturing and construction. However, its transparency and diverse morphological forms often lead to misclassification, especially under complex lighting conditions.
3. **Malachite:** Known for its vibrant green color and ribbon-like texture, malachite is a key indicator of copper oxide mineralization. Its unique appearance facilitates identification, but its color similarity to other green minerals (e.g., chrysocolla) poses classification challenges.
4. **Pyrite:** Commonly referred to as "fool's gold," pyrite is a widespread iron sulfide mineral with a distinctive golden luster and cubic structure. Its visual similarity to other metallic minerals increases the difficulty of accurate classification.
5. **Muscovite:** A common silicate mineral found in metamorphic and granitic rocks, muscovite is characterized by its bright white luster and plate-like structure. However, its reflective properties and similarity to other mica minerals complicate its identification.
6. **Biotite:** Biotite is a dark, plate-like silicate mineral often found in igneous and metamorphic rocks. Its dark color and reflective surface make it prone to misclassification, especially under poor lighting conditions.
7. **Chrysocolla:** This copper silicate mineral is typically associated with copper oxide deposits and is valued for its bright blue-green color. Despite its distinctive appearance, chrysocolla can be confused with malachite due to their similar coloration and co-occurrence in nature.

The selection of these minerals reflects their geological importance in mineral exploration and processing, as well as the technical challenges associated with their accurate classification. By addressing these challenges, our study aims to contribute to the development of more robust and reliable mineral identification systems.

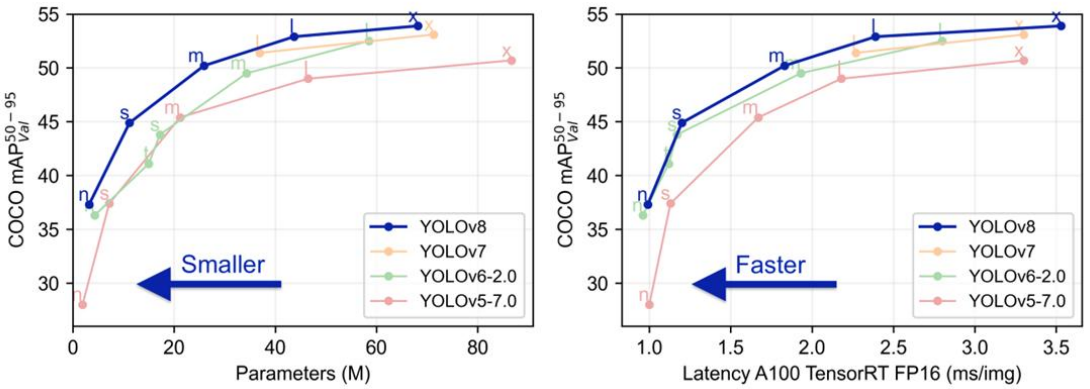


Figure 1: Performance comparison of different versions of the YOLO algorithm

2 YOLOv8 series model

YOLOv8, developed by Ultralytics, is the most widely used variant in the YOLO series, supplying notable enhancements in real-time discovery, classification, and division tasks. The YOLOv8 style incorporates a streamlined foundation network with a head network to improve both speed up and precision, keeping the performance of the YOLO series while boosting efficiency metrics, which has gathered considerable interest [24] YOLOv8 is versatile, sustaining a variety of jobs such as things discovery, image classification, circumstances division, and present estimate. This flexibility makes it incredibly versatile for a broad variety of computer system vision applications, from spotting objects in real-time to analyzing elaborate information in images and comprehending human positions. Its detailed capacities allow it to resolve varied difficulties in the field of computer system vision successfully. The total framework layout is highlighted in Figure 3.

2.1 C3 module and c2f module

The C2f component, unlike YOLOv5's C3 component, includes fewer criteria and enhanced feature removal capabilities. In C2f, the input initially undergoes a convolutional layer with parameters k = 1, s = 1, p = 0, c = out, and is then processed with multiple Bottleneck layers complying with the recurring and backbone outputs. This upgrade enhances gradient circulation, therefore improving training rate and general model efficiency. This adjustment yields a model that is both lighter and a lot more robust, achieving superior efficiency in intricate circumstances. As shown in Figure 2(a), the C3 module substantially enhances the model's abilities by incorporating the shunt device with the BottleNeck recurring module. This assimilation is built on the style concepts of CSPNet and residual networks, leading to boosted feature extraction and enhanced general model efficiency. This module consists of three layers integrating convolution, batch normalization, and SiLU activation functions, plus a variable number of BottleNeck devices. The last convolutional layer boosts the network count by a variable of two, incorporating inputs from both the primary and second gradient flow branches.

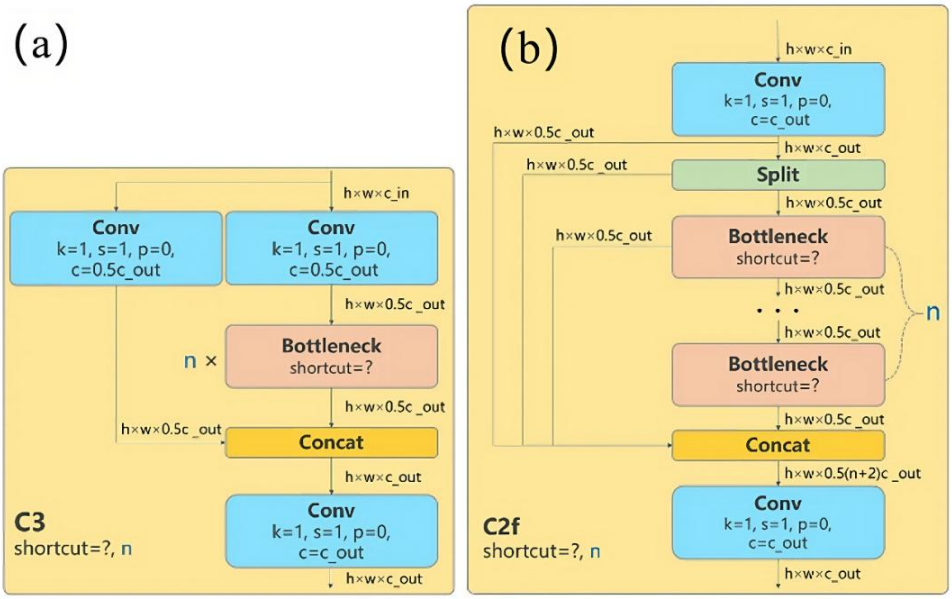


Figure 2: The framework diagram of C3 component and C2f module

Caption:
(a) C3 Module: This part of the diagram illustrates the C3 module, which integrates the shunt concept of CSPNet with a recurrent architecture to enhance feature extraction capabilities. This configuration aids in managing gradient flow more effectively, contributing to improved model resilience and learning efficiency.
(b) C2f Module: This section shows the C2f module designed to keep the model architecture lightweight while enriching feature representation. It optimizes gradient flow across different layers, thus enhancing both the training and inference efficiency of the model. This module plays a crucial role in processing multi-scale features and maintaining high computational efficiency.[33]

As shown in Figure 2(b), the C2f module consists of a series of BottleNeck blocks, each containing two convolutional layers. The first layer is used to process the input function map, which is then split right into multiple parts, refined individually before being merged. Such a design enables the model to capture richer contextual information, thereby enhancing the accuracy of target recognition. Next, the second convolutional layer further refines the output of the merged feature map. The C2f component considerably boosts the model's expressive ability and precision, bring about outstanding efficiency in functional target detection applications. By enhancing feature depiction and refining discovery accuracy, the C2f module makes certain that the model masters

identifying and assessing complicated and varied targets in real-world scenarios.

2.2 Head network, neck network, and backbone network

As depicted in Figure 3, the YOLOv8 model comprises a head network responsible for producing last predictions and a neck network that works as an intermediary between the foundation and the head, concentrating on feature fusion and handling. YOLOv8 achieves boosted model performance through a number of style developments. Structure on the ELAN structure of YOLOv7, YOLOv8 introduces the innovative C2f module, which replaces the previous C3 component. This improvement improves gradient flow and optimizes feature usage through added cross-layer connections. Moreover, YOLOv8 readjusts channel numbers to much better accommodate numerous discoveries demands and transitions from a 6x6 to a 3x3 convolutional kernel. This modification minimizes computational load while enhancing feature extraction efficiency. The network structure has likewise been structured by removing two convolutional layers. The C2f component additionally augments the model's abilities by incorporating added miss connections and splitting operations, thereby boosting function diversity and total expressiveness. Collectively, these innovations contribute to YOLOv8's superior accuracy, efficiency, and robustness in complex target detection tasks [16].

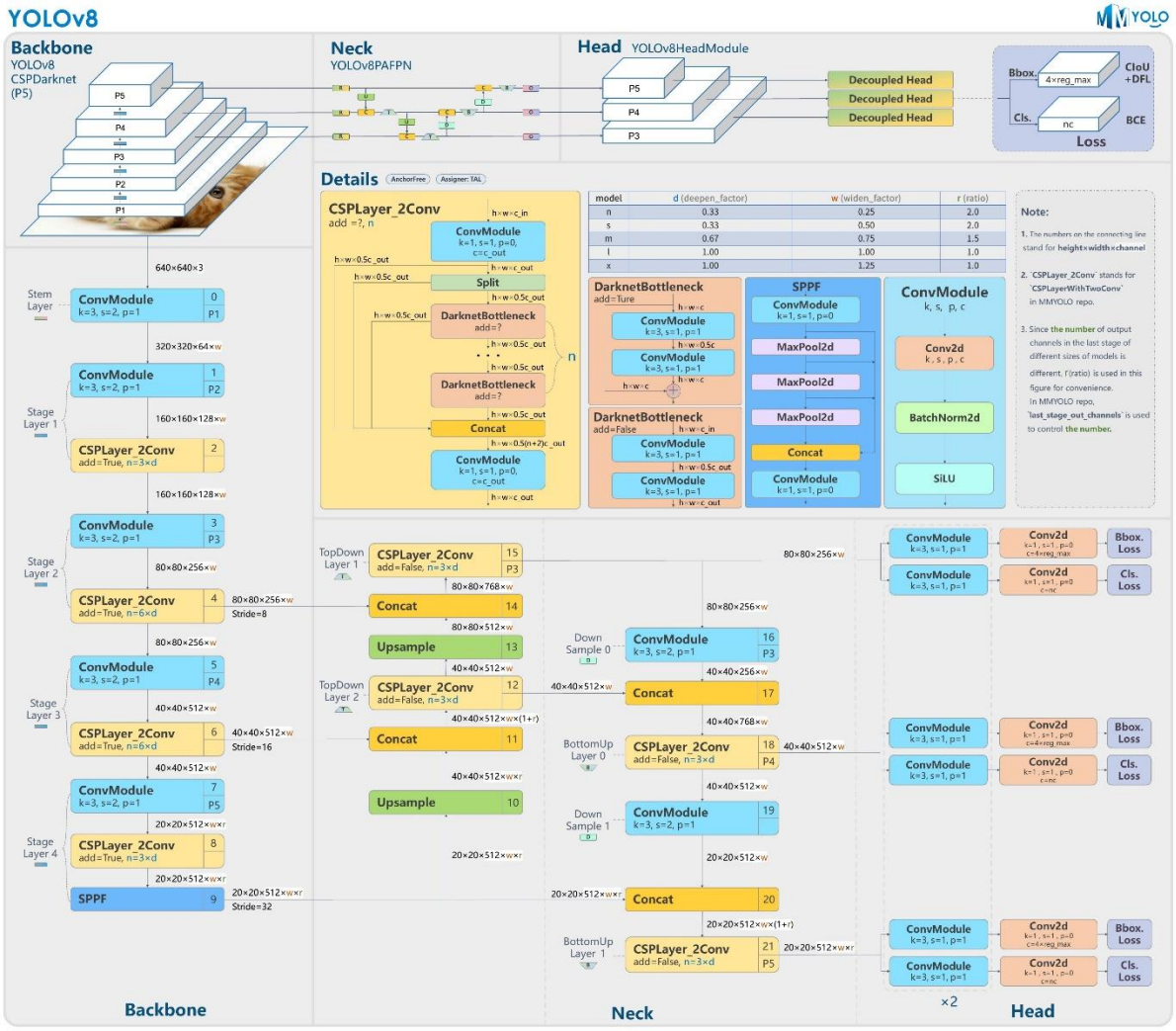


Figure 3: YOLOv8 overall model structure [16]

2.3 Uncoupling magnetic head

As illustrated in Figure 4, YOLOv8 introduces a well-considered design concept in task division, significantly advancing the architecture over traditional object detection frameworks. Traditionally, these frameworks combine classification and localization tasks, sharing the same parameter sets across both tasks. This commonality often leads to mutual interference, where the optimization of one task can adversely affect the accuracy of the other. However, YOLOv8 engineers have innovated with the adoption of a "decoupled head structure" that distinctly separates the parameter sets for classification and localization tasks. This strategic separation allows each network component—the classification subnetwork and the regression subnetwork—to focus intensively on its specific function:

- **Classification Subnetwork:** This part is solely responsible for predicting the probability distribution of object classes. It specializes in discerning the varied categories of objects within an image, optimizing the network to handle the subtleties of class differentiation efficiently.
- **Regression Subnetwork:** Conversely, this subnetwork focuses exclusively on estimating the offsets for bounding box coordinates. It is finely tuned to improve the precision of object localization within the spatial domain of the image.

This decoupled approach mitigates the conflict inherent in performing these tasks concurrently within a single shared network. By reducing the inter-task interference, YOLOv8 enhances the model's overall precision and robustness—particularly under complex scenarios where traditional models might struggle with accuracy.The implementation of this decoupled head structure does not merely segregate the tasks; it refines the model's architecture to ensure that both classification and localization are handled more effectively without compromising on either. This design shift leads to a notable enhancement in model performance and accuracy, providing a more reliable system that excels in varied and dynamic environments. Thus, YOLOv8's decoupled head model significantly contributes to the model's efficacy, affirming its architectural evolution as a major leap forward in object detection technology.

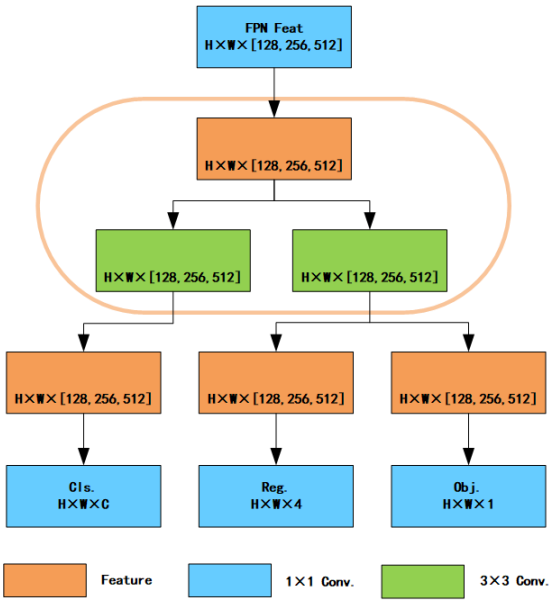


Figure 4: Flow chart of decoupled-head

2.4 SPPF module

Figure 5 highlights the capabilities of the Spatial Pyramid Pooling Fusion (SPPF) module in enhancing target detection. By refining feature maps across various dimensions, it significantly improves function extraction. The module utilizes a pyramid layer that captures and processes features at multiple scales, which helps to address the typical information loss seen with fixed-size pooling.At the start of its operation, the SPPF module applies multi-scale pooling techniques to the input feature map, segmenting it into grids of varying dimensions. Each grid is processed separately, enabling the creation of unique feature representations for each scale. This method enhances the module's ability to gather and integrate data from different spatial resolutions, leading to a richer and more robust feature extraction process. These multi-scale features are then combined using methods like concatenation or summation, merging them into a unified dataset that offers a detailed representation of the input. This integration is crucial for boosting the model's ability to detect and classify objects across diverse scales. To manage computational demands and reduce parameter count, the SPPF module incorporates a dimensionality reduction step, typically through a convolutional or fully connected layer. This reduction is essential for streamlining the processing phases. Finally, the enriched features, which contain extensive spatial and semantic details, are passed on to the next layers of the network. This transition significantly improves the model's effectiveness in identifying and classifying targets, thereby enhancing its performance and accuracy in complex detection scenarios.

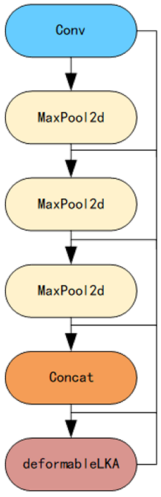


Figure 5: SPPF flow chart

2.5 Loss function-based label assignment

In its approach to loss function assignment, YOLOv8 introduces innovative strategies that distinguish it from conventional object detection systems, which typically rely on predefined anchor boxes for label assignment and loss computation. These traditional systems often require extensive hyperparameter tuning across different datasets, a challenge that YOLOv8 addresses with its adaptive strategies.

Task Alignment Learning (TAL): YOLOv8 incorporates Task Alignment Learning to synchronize the classification and regression tasks more effectively. Unlike conventional methods where these tasks may diverge, TAL aligns these tasks to enhance consistency and accuracy in the model's outputs. This alignment ensures that the learning objectives for both tasks are integrated, which helps in reducing discrepancies between classification scores and the predicted bounding

boxes' accuracy. The TAL method is not redundant; rather, it provides a novel way to ensure that adjustments in one task directly benefit the accuracy of the other, leading to more reliable and precise detection outcomes.

Distribution Focal Loss (DFL)[35]: Alongside TAL, YOLOv8 utilizes Distribution Focal Loss, which differs from standard Focal Loss by allowing the model to adapt more dynamically to the varying scales of objects within the dataset. DFL calculates loss based on the probability distribution of the class predictions, which helps in handling class imbalance more effectively — a common issue in object detection. This feature is particularly useful in mineral classification where object sizes can vary greatly.

CIoU Loss[34]: Furthermore, YOLOv8 combines DFL with Complete Intersection over Union (CIoU) loss. CIoU loss not only focuses on the overlap between the predicted and actual bounding boxes but also includes distance metrics between the box centers and aspect ratio terms. This comprehensive approach reduces the localization error and enhances the alignment of predicted boxes with ground truth, which is critical for achieving high precision in object detection tasks.

By integrating these advanced loss functions, YOLOv8 significantly refines its ability to detect and classify objects accurately across different scenarios and scales. The model's focus on dynamically adjusting its parameters based on the actual data it encounters makes it robust against the variations and complexities often found in diverse detection environments. This methodical enhancement of the loss function framework not only improves detection accuracy but also optimizes the overall training efficiency, making YOLOv8 a superior choice for modern object detection challenges.

2.6 Detailed overview of YOLOv8-CLS model and comparison with YOLOv8

2.6.1 Introduction to YOLOv8-CLS

YOLOv8-CLS is a specialized variant of the YOLOv8 model, tailored specifically for image classification tasks, unlike the broader object detection capabilities of the standard YOLOv8. This adaptation allows YOLOv8-CLS to focus on identifying and categorizing objects within an image without the additional computational overhead associated with bounding box predictions.

YOLOv8-CLS incorporates several key modifications that optimize it for classification:

Reduced complexity: The model simplifies the network architecture by omitting the Spatial Pyramid Pooling Fusion (SPPF) layers and the bounding box prediction modules found in YOLOv8.

Enhanced feature extractor: It employs a modified backbone that prioritizes features relevant for class categorization, enhancing the discriminative power of the network for classification tasks.

2.6.2 Comparative analysis with YOLOv8

While YOLOv8 is renowned for its efficiency and accuracy in object detection across various domains, YOLOv8-CLS refines the architecture to enhance image classification. Key differences include:

- **Model architecture:** YOLOv8-CLS uses a streamlined architecture with fewer convolutional layers, focusing solely on classifying the primary content of images. This results in faster processing times and reduced model size, suitable for deployment in environments with limited computational resources.
- **Optimization for classification:** The training process of YOLOv8-CLS has been optimized to maximize classification accuracy by eliminating loss functions unrelated to classification, retaining only the classification loss function to simplify the computational complexity of the loss function.

Figure 6 represent the architecture of YOLOv8-CLS, using different colors to distinguish the removed and modified components compared to the standard YOLOv8. Figure 7 provide a step-by-step visualization of the data flow within YOLOv8-CLS, emphasizing how input images are processed through the modified network to produce class predictions.

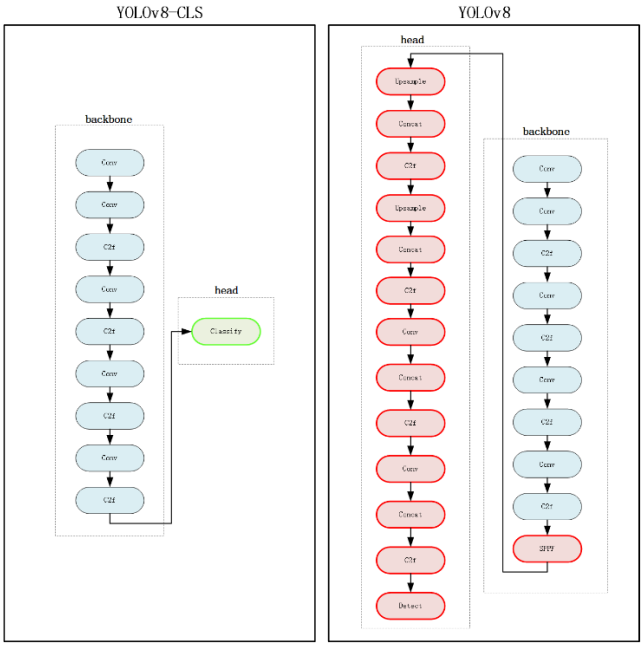


Figure 6: Architectural differences between YOLOv8 and YOLOv8-CLS

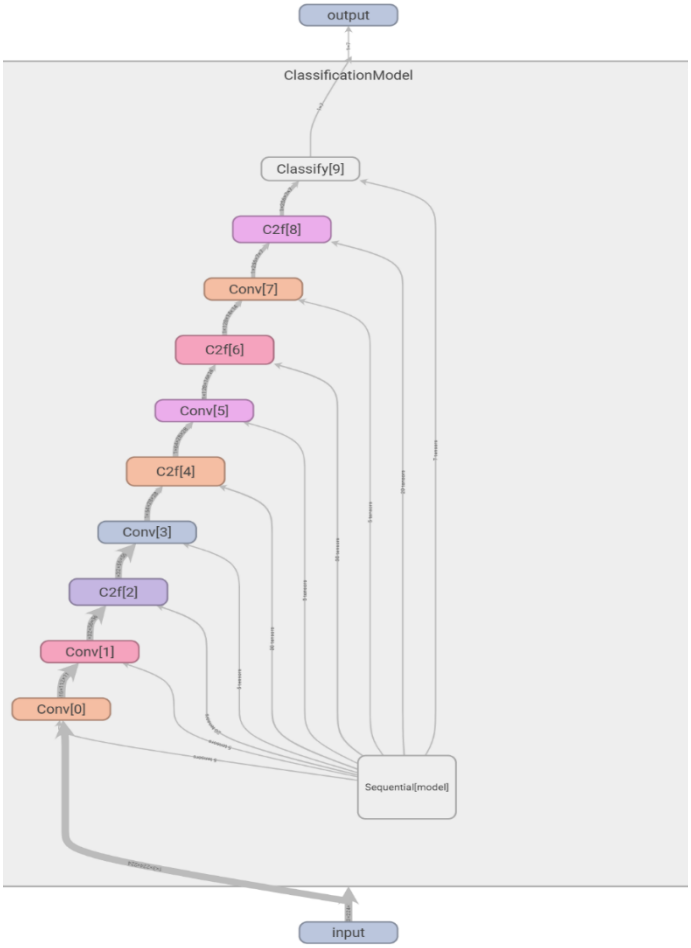


Figure 7: Data processing flow in YOLOv8-CLS

3 Data preparation and augmentation

3.1 Collection of mineral image data

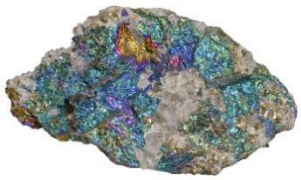
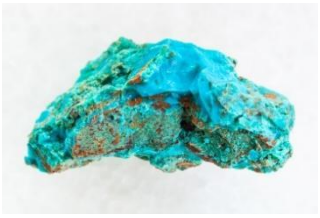
The model's capacity to discover deep attributes is highly affected by its interior style and the variety within the training information. As the training dataset broadens, the model removes more thorough attributes. This enhancement dramatically enhances the model's generalization and projection capabilities, enabling it to deal with even more elaborate and tough classification issues. By incorporating a varied series of features, the model becomes better furnished to handle intricate scenarios come across during training, recognition, and real-world identification tasks. Such a durable attribute collection is essential for ensuring the model's performance not just in scientific research yet additionally in sensible applications where precision and versatility are critical. This dataset was originated from an openly readily available mineral recognition dataset [27], making up seven minerals: bornite, quartz, malachite, pyrite, muscovite, biotite, and chrysocolla, causing an overall of 5,536 photographs after screening to get rid of unfavorable information. The dataset was divided into 3,872 images for the training set, 1,106 images for the validation collection, and 558 photos for the test set. The classification of the dataset for every mineral is presented in Table 1, while example pictures are shown in Table 2.

Table 1: Composition of the dataset [27]

Mineral species	Training Set (number of photos)	Validation Set (number of photos)	Test Set (number of photos)
biotite	721	206	103
quartz	812	232	116
bornite	290	83	42
chrysocolla	371	106	54
malachite	698	199	101
muscovite	235	67	35
pyrite	745	213	107
total	3243	1106	558

Table 2: Sample images for each of the 7 minerals in dataset [27]

Mineral species	training set	validation set	test set
biotite			
quartz			

bornite			
chrysocolla			
malachite			
muscovite			
pyrite			

3.2 Data augmentation

To enhance the accuracy of mineral image classification under varying conditions and improve model generalization, this study employs the built-in online data augmentation techniques provided by the Ultralytics library, integrated into the YOLOv8-CLS model. These techniques were specifically selected for their effectiveness in addressing the unique challenges posed by mineral image classification, where significant variations in texture, color, and form factor can impact classification accuracy.

Mosaic Stitching: This technique combines four different training images into a single composite image[37]. In the context of mineral images, which can exhibit considerable variability in scale and context within a single sample, mosaic stitching introduces diversity in background and scale. This method simulates a wide range of environmental conditions and teaches the model to recognize minerals across different contexts and juxtapositions, enhancing its ability to generalize to real-world scenarios. The mosaic parameter was set to 1.0, ensuring full application of this technique during training.

HSV Color Space Adjustment: Adjustments within the HSV (Hue, Saturation, Value) color space are crucial for mineral classification, as lighting conditions and camera settings can introduce substantial variations in color and brightness[37]. By randomizing these parameters during training, the model becomes more robust to such variations, ensuring it focuses on the intrinsic properties of the minerals rather than external

lighting conditions. The hue (h), saturation (s), and value (v) parameters were set to 0.015, 0.7, and 0.4, respectively, to simulate a variety of lighting scenarios while maintaining consistency in the model's learning.

Horizontal Flipping: This technique mirrors images along the vertical axis[37]. Given the isotropic nature of many minerals, where orientation does not affect their identity, horizontal flipping prevents orientation bias in the model. This helps improve the model's performance in environments where minerals may appear in arbitrary orientations. The flip horizontal (fliplr) parameter was set to 0.5, applying horizontal flipping with 50% probability during training. Vertical flipping was disabled by setting the flip vertical (flipud) parameter to 0.0, as vertical flips are generally not suitable for mineral images due to their natural orientation.

These data augmentation techniques address the specific challenges of mineral classification by increasing the diversity of training data representations. As a result, the model can perform robustly under a wide range of operational conditions. This approach aligns with recent advancements in adaptive data augmentation strategies, which dynamically adjust augmentation methods during training, thus enhancing the model's ability to generalize across various tasks. The chosen techniques and their parameters are consistent with the findings of recent research, such as Tang et al. [32], which highlights the effectiveness of real-time adaptive augmentation in improving the generalization capabilities of deep learning models.

4 Development and training of a smart mineral classification model

4.1 Model training and initial settings

In the training of the mineral classification model, our choice of epochs was guided by systematic empirical testing to determine the optimal point of convergence. Through multiple trials, we progressively adjusted the number of epochs and monitored the model's performance. We observed that by the 120th epoch, the model had already reached a plateau in learning, showing no significant gains in accuracy beyond this point. This experimental finding led us to set the maximum training limit at 120 epochs, as extending beyond this did not yield improvement and only increased computational costs unnecessarily.

The batch size of 128 was determined to be the most effective through these tests, balancing GPU utilization and memory consumption efficiently, thus allowing for smooth and effective training sessions. Similarly, the choice of an image dimension of 224 pixels was empirically optimized. Our tests showed that this dimension provided sufficient detail for the accurate classification of minerals while maintaining computational efficiency, which is critical for handling large datasets without compromising on speed or accuracy.

Regarding the learning rate, we settled on 0.00065 after experimenting with different rates to find a balance that allowed steady, effective learning without overshooting the model's capacity to converge. The application of a cosine scheduler was also an empirically motivated choice; it was selected for its effectiveness in fine-tuning the learning rate over time, which we noted helped in achieving better generalization by allowing more precise adjustments to the model weights in the later stages of training.

Additionally, our model uses **YOLOv8x-CLS** as the pre-trained weight, derived from training on a large dataset. The "X" option represents the largest model variant, with deeper layers and more network parameters, which typically leads to improved accuracy.

These parameters—each rigorously tested and chosen based on empirical evidence—collectively contributed to the robust performance of our model, ensuring that it not only learned efficiently but also generalized well to new, unseen data.

4.2 Model assessment

Unlike object detection, which utilizes multiple loss functions—including bounding box regression, classification loss, and objectness loss—to locate and categorize objects within an image, our mineral image classification model primarily relies on a single loss function—classification loss—to categorize each image into predefined mineral classes. This distinction emphasizes that our task focuses on classifying, not identifying individual mineral instances. This loss function is crucial for evaluating model performance, directing training, and enhancing the model. It measures how well the model's predictions match the true labels by quantifying the discrepancy between predicted and actual values. This measurement influences how the model updates its parameters, guiding the adjustment of weights and biases through optimization techniques like gradient descent to minimize errors and boost classification accuracy. As illustrated in Figure 8, the loss is initially high, reflecting poor model predictions at the start. Over the final 20 epochs, the loss values for both validation and test sets decrease and stabilize. For the training set, loss reduction is generally steady, though minor fluctuations are observed before convergence. The process of loss fitting on the validation set plays a crucial role in assessing the model's capacity for generalization. The model exhibited robust performance on this set, with the loss decreasing steadily and ultimately stabilizing. Final loss values were 0.26668 for the training set and 1.25980 for the validation set. The steady decline and stabilization of loss values in both convergence and overfitting analyses indicate that the model has likely converged and exhibits robust performance on the classification task.

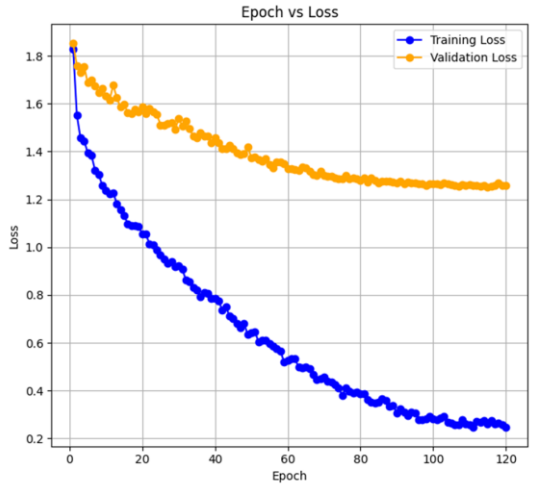


Figure 8: Loss curves for training and validation datasets

Top-1 Accuracy measures the rate at which the model's most confident prediction matches the actual label, while Top-5 Accuracy evaluates how often the true label appears among the model's top five predictions, as detailed in formulas (1) and (2). Figure 9 illustrates that, after extensive optimization and testing, the final Top-1 Accuracy and Top-5 Accuracy values reached stable levels of 0.92053 and 0.99399, respectively. These results demonstrate the model's exceptional precision and robustness in mineral image classification tasks.

$$\text{Top-1_Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \quad (1)$$

$$\text{Top-5 Accuracy} = \frac{\text{Number of times the true label is in the top 5 predicted labels}}{\text{Total number of predictions}} \quad (2)$$

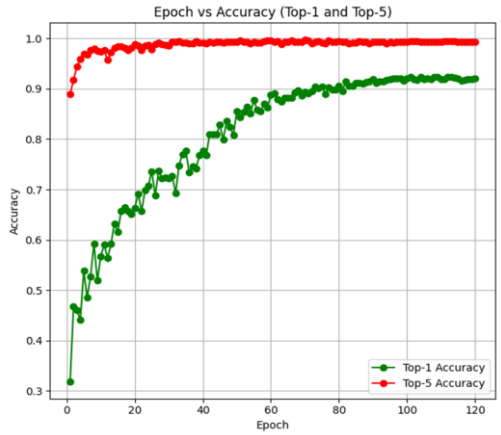


Figure 9: The curves for top-1 accuracy and top-5 accuracy

Confusion Matrix: ① A confusion matrix provides a summary of the prediction results for classification problems. The key aspect of a confusion matrix lies in its ability to quantify both correct and incorrect predictions using count values, further breaking them down by class. It visually illustrates the extent to which a classification model confuses different classes during prediction. Beyond merely identifying errors, the confusion matrix offers insight into the specific types of mistakes the model makes. This decomposition of results helps overcome the limitations of relying solely on classification accuracy as a performance metric. ② in the field of machine learning and statistical classification, a confusion matrix is a visualization tool primarily used in supervised learning, while in unsupervised learning, it is generally referred to as a matching matrix. Each column of the matrix represents the predicted instances of a given class, while each row corresponds to the actual instances of that class. The term "confusion matrix" originates from its function—making it easy to observe whether the model confuses different classes by misclassifying one as another.

Figure 10 displays the confusion matrix from this training experiment, providing several key insights: (1) There is no misidentification of backgrounds in the images. (2) The rate of false positives, where images are incorrectly classified, is maintained below 15%, demonstrating the model's overall precision. (3) The true positive rates for biotite, bornite, chrysocolla, malachite, muscovite, pyrite, and quartz are 0.96, 0.80, 0.90, 0.96, 0.66, 0.96, and 0.97, respectively. These rates reflect the model's accuracy in classifying each mineral, with true positive rates being crucial as they measure the model's ability to correctly identify each mineral type as present when it indeed is. This metric is particularly important in mineral classification to ensure that each type is accurately detected and not confused with others, which can directly impact mining and processing efficiency. (4) Bornite and muscovite have true positive rates below 0.9, indicating weaker recognition performance. This suggests specific issues with feature extraction for these minerals. The lower true positive rate for bornite and muscovite can be attributed to their complex physical properties, which may not be as distinct or may be similar to other minerals in the dataset, leading to challenges in effectively distinguishing them from others. This highlights the need for targeted improvements in the feature extraction process, such as enhancing the model's ability to handle variations in color, texture, and shape more distinctively for these minerals.

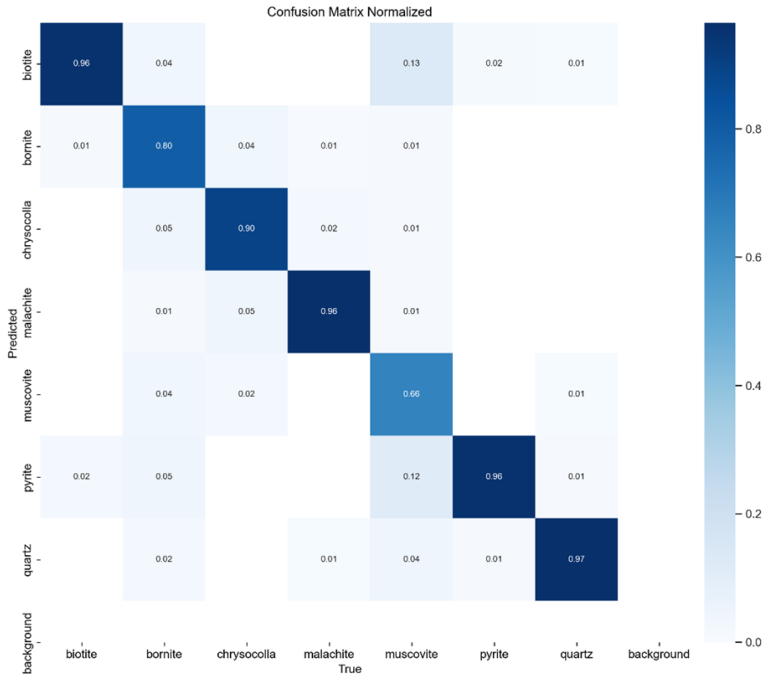


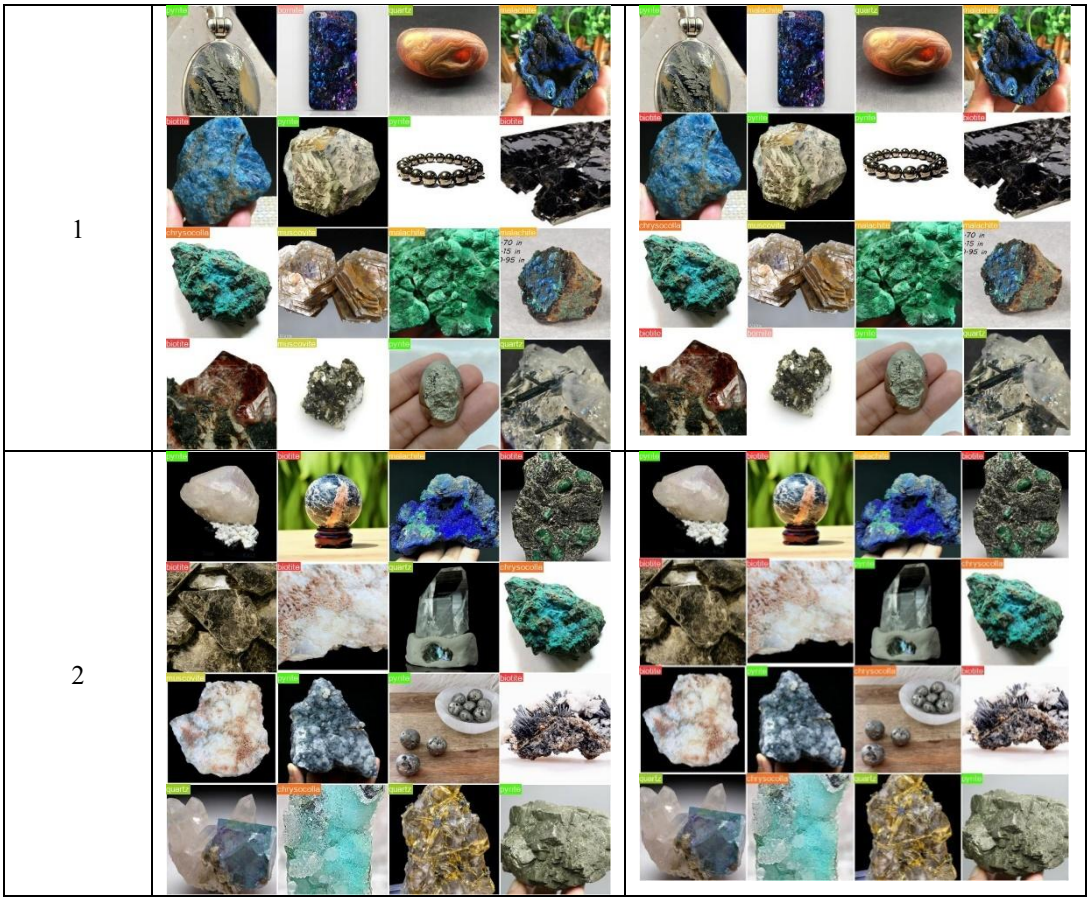
Figure 10: Confusion matrix of the validation set

5 Model testing

To assess the generalization capability of the trained model for mineral classification, we conducted testing using the test set from Table 1, which consists of 558 images. These images were not used in the training or validation stages, ensuring the evaluation was conducted on unseen data. The test set includes samples of 7 minerals: bornite, quartz, malachite, pyrite, muscovite, biotite, and chrysocolla. Each mineral has distinct features that challenge the model's identification capacities. For example, biotite is determined by its dark, plate-like framework, quartz is notable for its openness and variety of forms, and bornite is distinguished by its copper-green shade and metal sparkle. The performance of the model was examined by comparing truth labels with the predicted tags for each batch in the test set. The comprehensive results of this comparison are summarized in Table 3. Additionally, Figure 11 shows the complication matrix, which gives an extensive view of the model's precision and its efficiency across different mineral categories.

Table 3: Original and predicted labels for batches 0-2 in the testing.

Batch	labels	predict
0		



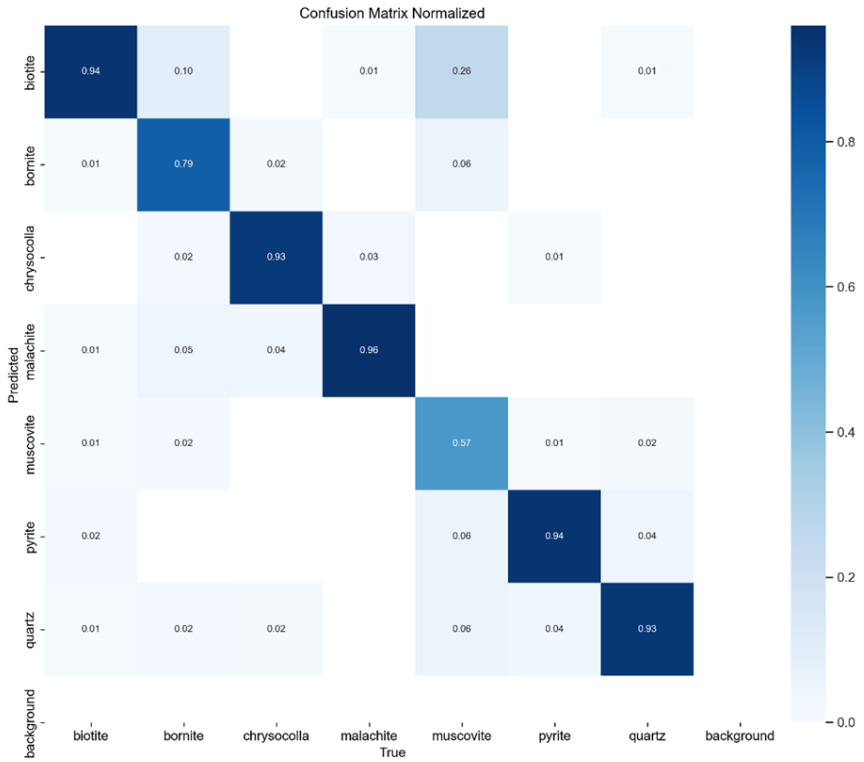


Figure 11: Confusion matrix of the test set

An in-depth evaluation of the speculative outcomes yields useful insights right into the model's performance on numerous mineral samples and uncovers potential areas for enhancement: (1) Bornite (TP = 0.79): Bornite's complicated texture and the presence of impurities can obscure its features. The limited sample size in the

training set hampers effective feature learning, resulting in some false positives. (2) Quartz (TP = 0.93): The transparency and diverse morphology of quartz provide abundant feature information for accurate identification. However, its transparency can blend with the background or other transparent minerals, especially under complex lighting conditions, which can slightly affect the true positive rate. To address this issue, we attempted to simulate such conditions through data augmentation, such as adjusting HSV parameters, to enhance the model's ability to accurately recognize quartz under challenging lighting. The true positive rate of 0.93 for quartz in the test set aligns well with our expectations. (3) Malachite (TP = 0.96): Malachite's unique color and ribbon-like texture make it relatively easy to identify, resulting in a high true positive rate. However, its color similarity to silica malachite and their common co-occurrence can occasionally impact accuracy. (4) Pyrite (TP = 0.94): Pyrite's distinctive golden luster and cubic structure allow it to be easily distinguished from other minerals, contributing to a high true positive rate. (5) Muscovite (TP = 0.57): Muscovite's bright white luster and plate-like structure are prone to reflections, which can affect recognition accuracy under varying lighting conditions. Additionally, its similarity in color and morphology to other mica minerals, such as biotite, further complicates differentiation, leading to the lowest accuracy for this mineral in the model's test results. (6) Biotite (TP = 0.94): Biotite performed well in the test set but encountered certain challenges. Its dark, plate-like structure is often mistaken for other dark minerals or even shadows, particularly under weak or uniform lighting conditions, which may significantly reduce prediction accuracy. Moreover, the highly reflective surface of the mineral can cause misleading highlights, potentially leading to false positives during detection. These combined factors challenge the model's ability to accurately classify biotite, highlighting the need for improved strategies to address these issues. (7) Chrysocolla (TP = 0.93): Chrysocolla's distinctive bright color and fibrous structure facilitate its identification, resulting in high accuracy. However, its color similarity to other green minerals, such as malachite, can lead to reduced recognition accuracy when mixed with them. Additionally, overlapping shadows and mineral co-occurrence may slightly reduce the overall accuracy of detection.

The variations in true positive rates for different minerals highlight the model's varying effectiveness in recognizing their distinct physical properties. Minerals with unique features, such as pyrite and malachite, benefit from their distinct colors and shapes, which result in higher classification accuracy. Pyrite, with its golden luster and cubic structure, and malachite, characterized by its vibrant green color and ribbon-like texture, are easily distinguishable, contributing to their high true positive rates. In contrast, minerals like muscovite and bornite present additional challenges. Muscovite's bright white luster and plate-like structure can reflect light in ways that complicate accurate recognition, and its similarity to other mica minerals further hinders its identification. Bornite's complex texture and the presence of impurities in some samples also add to the difficulty, resulting in lower recognition accuracy. These discrepancies highlight areas where the model's performance can be improved. To address these issues, future optimization efforts should focus on refining feature extraction techniques and incorporating a broader range of training examples. Enhancing the model's ability to handle varying lighting conditions and overlapping mineral characteristics will be crucial for improving its precision and efficiency in real-world applications.

6 Model comparison

To further investigate the performance gap between the YOLOv8-CLS model and classical models, we also introduced another open-source dataset with pre-defined training, test, and validation sets (as shown in Table 4) to explore the differences in generalization capabilities across various models. The following presents the specific experimental procedures and results.

Table 4: Composition of the dataset [36][27]

Mineral species	Training Set (number of photos)	Validation Set (number of photos)	Test Set (number of photos)
biotite	306	11	5
quartz	309	10	5
bornite	306	8	5
chrysocolla	639	21	9
malachite	318	10	5
muscovite	459	15	8
pyrite	585	17	10
total	2922	92	47

We compared the YOLOv8-CLS model with the classical ResNet50 and ResNet101 architectures. To control for variables and ensure the richness of the test samples, we used the dataset from Table 1 for training ResNet50 and ResNet101 with the same training parameters and data augmentation settings. The models

were tested using the test set portion from Table 1 and the training set portion from Table 4. The test results are shown in Figures 12 and 13, and the performance comparison between the models is summarized in Table 5.

Table 5: Comparison of model performance [27]

Model	GPU memory usage(training)	Top1 Accuracy in Table 1 test set	Top1 Accuracy in Table 4 train set	Parameters	GFLOPs
Resnet50	8.69GB	0.90502	0.79535	26113159	68.8 GFLOPs
Resnet101	12.52GB	0.90322	0.75941	45079175	129.4 GFLOPs
YOLOv8-CLS	1.68GB	0.90681	0.80219	1443847	3.3GFLOPs

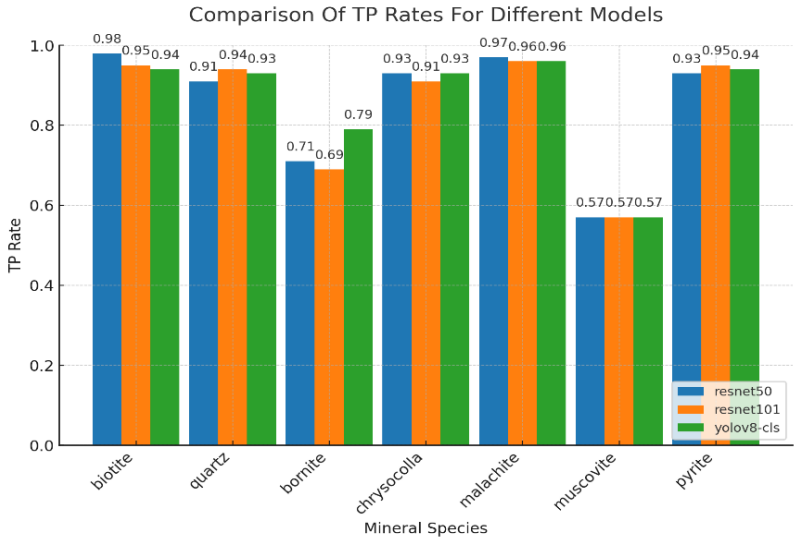


Figure 12: Comparison of TP rates for different models on table1's test set

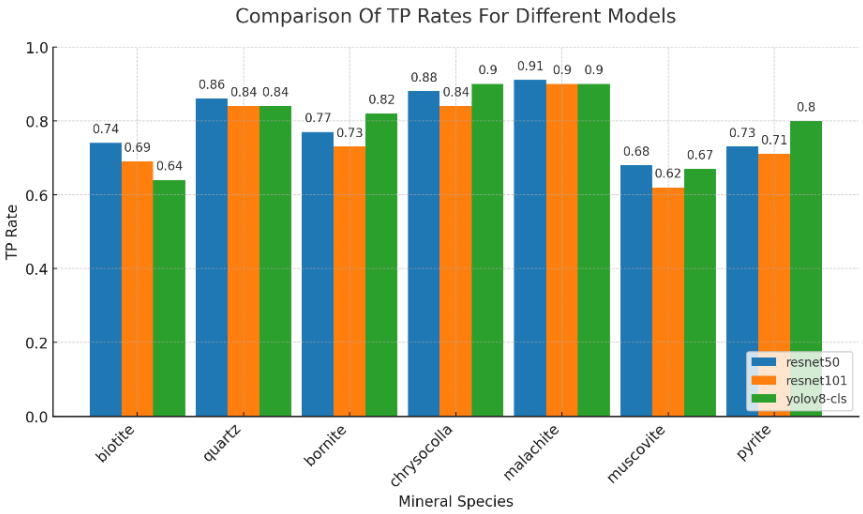


Figure 13: Comparison of TP rates for different models on table4's train set

In Table 5, we compare model performance using the memory usage during training, model parameter count, GFLOPs, and Top-1 Accuracy on both the test set from Table 1 and the training set from Table 4. Since Top-1 Accuracy is a stricter metric and more reflective of model performance differences, it was used as the

primary comparison criterion. From the results in Table 5, the YOLOv8-CLS-trained mineral classification model achieves the highest Top-1 Accuracy on both the Table 1 test set and the Table 4 training set compared to all other models, with values of 0.90681 (Top-1 Accuracy on Table 1 test set) and 0.80219 (Top-1 Accuracy on Table 4 training set), respectively. In comparison, ResNet50 and ResNet101 show a Top-1 Accuracy improvement over YOLOv8-CLS of 0.00179 and 0.00359 on the Table 1 test set, and 0.00684 and 0.04278 on the Table 4 training set, respectively.

In terms of memory usage during training, YOLOv8-CLS uses only 1.68GB of GPU memory, while ResNet50 and ResNet101 both exceed 6GB. Additionally, YOLOv8-CLS has the lowest model parameter count and GFLOPs. Specifically, YOLOv8-CLS has 20.85 and 39.21 times fewer GFLOPs than ResNet50 and ResNet101, respectively, which highlights the significant performance advantage of YOLOv8-CLS.

By analyzing Figure 12, we found that YOLOv8-CLS outperforms ResNet50 and ResNet101 in terms of true positive rates for bornite in the Table 1 test set by 0.08 and 0.1, respectively. However, for other minerals, the true positive rates were comparable. In Figure 13, YOLOv8-CLS shows a clear advantage in true positive rates for bornite and pyrite, but it lags behind ResNet50 and ResNet101 in the true positive rate for biotite. This indicates that YOLOv8-CLS has superior generalization ability compared to ResNet50 and ResNet101.

Furthermore, a comparison of Figure 13 with Figure 12 shows a significant decline in true positive rates for biotite and pyrite, while the true positive rates for bornite and muscovite improved. The true positive rates for other minerals showed slight declines. This change can be attributed to the shift in the test dataset. Upon visually inspecting the images in the Table 4 training set, we observed that the images in this set came from more complex environments that better reflect real-world application scenarios. This exposed a challenge for the model in accurately identifying biotite under complex lighting conditions, particularly low light, which aligns with our analysis in Section 5. The decrease in recognition accuracy for pyrite may be due to the intense reflections caused by its unique golden luster under strong lighting conditions with high contrast, which could also have contributed to the drop in performance.

Overall, YOLOv8-CLS demonstrates higher accuracy, lower performance overhead, and stronger generalization capability compared to ResNet50 and ResNet101 in the mineral classification task. However, it also exposes potential issues, as discussed in Section 5, particularly in handling specific lighting conditions and mineral feature overlap.

7 Discussion

The model developed in this experiment demonstrated an exceptionally high accuracy in mineral classification when tested on images from the Table 1 test set, achieving a Top-1 accuracy of approximately 90.68%. This result highlights the model's significant performance advantages in a controlled environment, demonstrating its effectiveness in both theoretical research and practical applications. The high precision not just verifies the model's robustness in controlled speculative setups yet likewise highlights its possibility for real-world usage, where accurate mineral classification is essential. Nevertheless, the model showed misclassification when managing mineral classification under particular illumination conditions. For instance, as shown in Figure 12, the model misclassified quartz as pyrite. This misclassification may arise from the openness buildings of quartz under particular illumination conditions engaging with the history, causing characteristics that carefully appear like those of pyrite. Although the confusion matrix in Figure 11 shows that the model achieves a high true positive rate of 0.93 for quartz, the misclassification examples in Figure 14 and the decline in true positive rate to 0.84 during model comparison indicate that there is still room for improvement in the model’s recognition accuracy for quartz. This example also highlights performance issues under certain lighting conditions, which aligns with the conclusions from previous experiments. This points to a key area for improvement in our future work and emphasizes the limitations of relying solely on existing confusion matrices and other performance metrics to expose model weaknesses. Therefore, it is essential to test the model in real-world mineral classification scenarios, where practical testing in real-world environments can identify and address potential defects—this will be a primary focus of our future research. Similar challenges have been observed in Salima’s research[30], where environmental factors such as lighting variations and image quality can significantly impact model performance.



Figure 14: Miscalculation sample (real mineral classification on left, misclassification on right)

Through the model comparison section, we further analyzed the performance of the YOLOv8-CLS model in mineral classification. This analysis demonstrates that the YOLOv8-CLS model, in comparison to ResNet50 and ResNet101, not only achieves higher accuracy and stronger generalization capabilities in mineral classification but also reduces hardware performance demands. However, it also reveals that the model's misclassification issues are not only related to minerals with similar visual features but are also closely tied to environmental lighting conditions. This highlights the model's limited ability to accurately distinguish mineral types in complex environments, emphasizing specific areas where its efficiency can be improved and providing insights for future improvements aimed at addressing these issues. SHE's study has suggested that integrating enhanced image processing techniques, such as Retinex-based image enhancement, can mitigate these challenges by improving feature extraction and contrast in low-light conditions [31]. To overcome these limitations, several improvements could be explored in future research study: (1) Expand the training dataset to consist of a larger range of minerals with similar appearance features, which will certainly help the model compare refined differences more effectively. (2) Enhance the model architecture by incorporating innovative techniques like deeper convolutional networks and interest devices. Deeper convolutional networks can boost function extraction by capturing a lot more intricate patterns and information, while interest mechanisms allow the model to focus on one of the most relevant parts of the image, even in the middle of intricate backgrounds. These improvements are expected to substantially improve the model's ability to process and properly identify complicated visual circumstances, leading to more trusted mineral identification. (3) Incorporate advanced post-processing techniques, such as sophisticated image segmentation and enhancement methods, to successfully lower misclassification prices and dramatically enhance total precision. Image division can assist isolate private minerals for more precise analysis, while improvement methods can clear up image information and improve feature visibility. SHE's research has demonstrated that image enhancement methods, particularly those designed for low-contrast environments, can improve classification accuracy in vision-based tasks [31]. (4) Increase image sampling rates to capture more detailed mineral features, thereby providing the model with additional information for better feature extraction.

Furthermore, the testing conducted in this research study was mainly performed under regulated speculative conditions, which gave an organized setting for evaluating the model's performance. Nevertheless, the model's performance in real-world applications has actually not yet been completely examined. This constraint suggests that while the design demonstrates encouraging lead to managed settings, additional research study and recognition are needed to analyze its integrity and accuracy in different and practical scenarios. Real-world applications often involve unpredictable environmental factors, and Salima's study have highlighted the importance of adaptive deep learning approaches in handling such variations [30]. So Real-world testing will be crucial for validating the model's effectiveness and adaptability to varying environmental conditions and practical use cases. Given the challenges inherent in real-world settings-- such as varying lighting issues, history noise, and the exposure of many overlapping minerals-- it is important to carefully check and verify the variation under a lot more varied and dynamic conditions. This will help ensure that the model carries out dependably throughout a more comprehensive series of circumstances. Such useful examinations will certainly use a clearer understanding of the model's efficiency and its capability to get used to real-world complexities. This will certainly make certain the variation's durability and stability in diverse and much less regulated situations. For example, performing experiments in real mineral arranging manufacturing environments or in field treatments for geological exploration would provide a much more precise examination of the style's effectiveness and applicability. These real-world tests will certainly enable scientists to evaluate the style's effectiveness under differing conditions and recognize prospective problems related to complex histories, lighting changes, and mineral

problem. These field tests not just offer helpful suggestions for style optimization yet likewise expose its restraints and areas for improvement in reasonable applications, therefore making sure safe and reputable effectiveness in more variable and dynamic atmospheres.

Owing to constraints imposed by existing experimental infrastructure, our current investigation lacks extensive field validation in operational industrial settings to empirically evaluate the performance boundaries of our mineral identification framework under unconstrained environmental parameters. Through rigorous analysis of experimental datasets complemented by theoretical foundations established in SHE's seminal research on environmental interference mitigation in coal mining applications [31], we postulate that the primary challenges during practical deployment likely stem from four interdependent mechanisms: (1) suboptimal lighting conditions characterized by inadequate illumination intensity and reduced contrast ratios; (2) complex light-matter interactions arising from the anisotropic optical properties of translucent mineral specimens; (3) inter-class feature ambiguity in crystalline morphology descriptors; and (4) resolution-dependent attenuation of discriminative textural signatures. These compounded photometric and geometric variabilities could engender nonlinear error propagation through the feature extraction pipeline, potentially inducing cascading effects that degrade critical classification performance indicators such as Top-1 Accuracy and Top-5 Accuracy.

Finally, while the model demonstrates notable efficiency under the current experimental conditions, further research is essential to fully realize its potential. To ensure the model's broader application and performance, it is crucial to address possible misclassification issues and enhance its adaptability to real-world scenarios. By tacking these challenges, the model's robustness will be significantly improved. As suggested by SHE and Salima's prior research, integrating adaptive learning techniques and real-time enhancement algorithms may provide a viable path forward in enhancing classification accuracy in challenging environments[30] [31]. This, in turn , will provide a solid theoretical foundation and the necessary technical support for the widespread adoption of mineral recognition technology. As the model is further tested and refined in diverse practical settings, mineral recognition modern technology is expected to play an increasingly significant role in both industrial and research contexts. This methodological advancement demonstrates significant potential for enhancing precision in automated mineral categorization systems. The industrial implementation of our model could optimize operational efficiency in ore processing workflows through streamlined mineral classification processes, thereby mitigating resource depletion and minimizing material wastage. Furthermore, the framework exhibits extensible applicability in geological prospecting domains, where it may serve as a decision-support tool for field geologists by enabling real-time mineralogical analysis through integration with portable multispectral imaging devices. The architecture's inherent adaptability also suggests promising applications in automated petrographic analysis and remote sensing-based mineral mapping initiatives.

8 Conclusion

This paper presents an advanced intelligent mineral image classification model using deep learning techniques, specifically leveraging the YOLOv8-CLS algorithm. The model is designed to efficiently identify seven usual mineral kinds: bornite, quartz, malachite, pyrite, muscovite, biotite, and chrysocolla. After training for 120 epochs, the model's performance metrics showed substantial stabilization, with Top-1 precision and Top-5 accuracy reaching 0.92053 and 0.99399, particularly. These outcomes highlight the model's remarkable precision and robustness in mineral image classification tasks. The high-performance metrics highlight the variation's effectiveness and capacity for useful application in correctly figuring out a range of mineral types, showcasing its viability for use in different mineral acknowledgment situations.

Although the model performs excellently overall, some limitations were revealed when testing with an additional dataset during the model comparison section. The observed misclassifications are primarily attributed to the visual similarities among specific minerals such as shade, shape, and appeal. This similarity complicates the style's job of exactly distinguishing between these minerals, for that reason evaluating its total effectiveness in classification. Particularly when the mineral to be identified shares similar morphology and color with other minerals recognized by the model, the model may struggle to distinguish the subtle differences between them, leading to biased classification results. The decrease in the model's robustness under complex lighting conditions may be attributed to strong reflections and refractions occurring when light strikes the minerals. The conclusions drawn from this research study deal significant insights that are important for advancing smart mineral classification techniques. These findings provide a basic understanding that can help future r & d in this area. Via a thorough analysis of the source of model's errors and targeted optimization instructions, this research reveals the application possibility of deep learning models in mineral classification and information paths for enhancing variation efficiency in future research study. These results develop a solid basis for advancing clever mineral classification innovation and encouraging its functional release and more comprehensive use in real-world applications.

References

[1] Zhou, Y.Z.; Zuo, R.G.; Liu, G.; Yuan, F.; Mao, X.C.; Guo, Y.J.; Xiao, F.; Liao, J.; Liu, Y.P. The Great-leap-forward Development of Mathematical Geoscience During 2010 – 2019: Big Data and Artificial Intelligence Algorithm Are Changing Mathematical Geoscience. Bull. Mineral. Petrol. Geochem. 2021, 40, 556–573.

[2] Zhang, L.J.; Lu, W. h.; Zhang, J.D.; Peng, G.X.; Bu, J.C.; Tang, K.; Xie, J.C.; Xu, Z.B.; Yang, H.Y. Rock and mineral thin section identification based on deep learning. Geosci. Front. 2024, 31, 498–510.

[3] BATIN, Mikhail, et al. Artificial intelligence in life extension: from deep learning to superintelligence. Informatica, 2017, 41.4.

[4] Xu, S.T.; Zhou, Y.Z. Artificial Intelligence Identification of Ore Minerals under Microscope Based on Deep Learning Algorithm. Acta Petrol. Sin. 2018, 34, 3244-3252.

[5] Alerigi, D.P.S.R.; Li, W. Method for providing rock characterization and classification for geo-exploration, involves applying deep learning models to newly received data that includes data, and predicting properties based on newly received geo-exploration data. Saudi Arabian Oil Co 2022, D65162.

[6] Zeng, X.; Ji, X.H.; Xiao, Y.C.; Wang, G.W. Mineral Identification Based on Deep Learning That Combines Image and Mohs Hardness. Minerals 2020, 11, 506, https://doi.org/10.3390/min11050506.

[7] Wang, J.B.; Xue, L.F.; Gao, X. Identification Method of Volcanic Rock Slices Based on A Deep Residual Shrinkage Network. In Proceedings of the Fourth International Conference on Geoscience and Remote Sensing Mapping, Wuhan, China, 14–16 April 2023; Volume 12551, pp. 389 – 394, https://doi.org/10.1117/12.2668168.

[8] Zhang, Y.; Li, M.C.; Han, S.; Ren, Q.B.; Shi, J. Intelligent Identification for Rock-Mineral Microscopic Images Using Ensemble Machine Learning Algorithms. Sensors 2019, 18, 3914, https://doi.org/10.3390/s19183914.

[9] Zhang, S.X.; Yang, Y.Y.; Sun, F.C.; Fang, B. Application of Image Sensing System in Mineral/Rock Identification: Sensing Mode and Information Process. Adv. Intell. Syst. 2023, 5, 2300206, https://doi.org/10.1002/aisy.202300206.

[10] L Chen, S Li, Q Bai, J Yang, S Jiang, Y Miao. Review of Image Classification Algorithms Based on Convolutional Neural Networks. Remote Sensing, 2021 - mdpi.com, https://doi.org/10.3390/rs13224712

[11] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 770-778).

[12] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 2261-2269).

[13] GAAFAR, Alaa Sahl; DAHR, Jasim Mohammed; HAMOUD, Alaa Khalaf. Comparative analysis of performance of deep learning classification approach based on LSTM-RNN for textual and image datasets. Informatica, 2022, 46.5, https://doi.org/10.31449/inf.v46i5.3872.

[14] Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In Proceedings of the International Conference on Learning Representations (ICLR).

[15] Howard, A. G., Zhu, M., Chen, B., et al. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

[16] He, L.; Zhou, Y.; Zhang, C. Application of Target Detection Based on Deep Learning in Intelligent Mineral Identification. Minerals 2024, 14, 873, https://doi.org/10.3390/min14090873.

[17] Mao, Z.H.; Zhu, J.L.; Wu, X.; Li, J. Review of YOLO Based Target Detection for Autonomous Driving. J. Comput. Eng. Appl. 2022, 58, 68–77.

[18] Al Muksit, A.; Hasan, F.; Emon, M.F.H.B.; Haque, M.R.; Anwary, A.R.; Shatabda, S. YOLO-Fish: A robust fish detection model to detect fish in realistic underwater environment. Ecol. Inform. 2022, 72, 101847, https://doi.org/10.1016/j.ecoinf.2022.101847.

[19] Zuraimi, M.A.B.; Zaman, F.H.K. Vehicle detection and tracking using YOLO and Deep SORT. In Proceedings of the 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), Penang, Malaysia, 3–4 April 2021; pp. 23-29, DOI: 10.1109/ISCAIE51753.2021.9431784.

[20] Vilar-Andreu, M.; García, L.; Garcia-Sanchez, A.J.; Asorey-Cacheda, R.; Garcia-Haro, J. Enhancing Precision Agriculture Pest Control: A generalized Deep Learning Approach with YOLOv8-based Insect Detection. IEEE Access 2024, 12, 84420–84434, DOI: 10.1109/ACCESS.2024.3413979.

[21] Prinzi, F.; Insalaco, M.; Orlando, A.; Gaglio, S.; Vitabile, S. A yolo-based model for breast cancer detection in mammograms. Cogn. Comput. 2024, 16, 107–120, https://doi.org/10.1007/s12559-023-10189-6.

[22] Reddy, S.M.; Rakesh, K.; Aluvala, S.; Bindu, G.; Husseen, A. Fault Detection and Classification in Semiconductor Manufacturing for Sensor Screening Using Multi-Layer. In Proceedings of the 2024 International Conference on Distributed Computing and Optimization Techniques (ICDCOT), Deep Neural Network, Bengaluru, India, 15 – 16 March 2024; pp. 1-4, DOI: 10.1109/ICDCOT61034.2024.10515343.

[23] Reis, D.; Kupec, J.; Hong, J.; Daoudi, A. Real-time flying object detection with YOLOv8. arXiv 2023, arXiv:2305.09972, https://doi.org/10.48550/arXiv.2305.09972.

[24] Wu, T.Y.; Dong, Y.K. YOLO-SE: Improved YOLOv8 for Remote Sensing Object Detection and Recognition. IEEE Trans. Appl. Sci. 2023, 13, 12977, https://doi.org/10.3390/app132412977.

[25] Wang, X.L.; Girshick, R.; Gupta, A.; He, K.M. non-local neural networks. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7794-7803.

[26] Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. arXiv 2018, arXiv:1804.02767.

[27] Attallah, Y.; Minerals Identification & Classification. Kaggle. Retrieved 2023.
[28] Zhang, M.H.; Wang, Z.H.; Song, W.; Zhao, D.F.; Zhao, H.J. Efficient Small-Object Detection in Underwater Images Using the Enhanced YOLOv8 Network. IEEE Trans. Appl. Sci. 2024, 14, 1095. https://doi.org/10.3390/app14031095.
[29] Li, J.F.; Yan, C.X. YOLOv8 with Multi Strategy Integrated Optimization and Application in Object Detection. Autom. Mach. Learn. 2024, 5, 23-31, DOI: 10.23977/autml.2024.050104.
[30] Bourougaa-Tria, Salima, et al. Spubbin: Smart public bin based on deep learning waste classification an iot system for smart environment in algeria. Informatica, 2022, 46.8, https://doi.org/10.31449/inf.v46i8.4331.
[31] SHE, Dong. Retinex Based Visual Image Enhancement Algorithm for Coal Mine Exploration Robots. Informatica, 2024, 48.11, https://doi.org/10.31449/inf.v48i11.6003.
[32] Tang, Zhiqiang, et al. OnlineAugment: Online data augmentation with less domain knowledge. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16. Springer International Publishing, 2020. p. 313-329, https://doi.org/10.1007/978-3-030-58571-6_19.
[33] SOHAN, Mupparaju, et al. A review on yolov8 and its advancements. In: International Conference on Data Intelligence and Cognitive Informatics. Springer, Singapore, 2024. p. 529-545, https://doi.org/10.1007/978-981-99-7962-2_39.
[34] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-iou loss: Faster and better learning for bounding box regression," in Proceedings of the AAAI conference on artificial intelligence, vol. 34, pp. 12993–13000, 2020, DOI: https://doi.org/10.1609/aaai.v34i07.6999.
[35] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. Yang, "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," Advances in Neural Information Processing Systems, vol. 33, pp. 21002–21012, 2020, https://doi.org/10.48550/arXiv.2006.04388.
[36] MultiLabel classification - v1 2024-05-29 4:19pm https://universe.roboflow.com/finalproject-flqsu/multilabel-classification-vbxvr, https://doi.org/10.1007/978-3-319-41111-8_2.
[37] SHORTEN, Connor; KHOSHGOFTAAR, Taghi M. A survey on image data augmentation for deep learning. Journal of big data, 2019, 6.1: 1-48, https: //doi.org/10.1186/s40537-019-0197-0.

Multi-Model Secure Redundant Storage for IoT Data Using Random Forests, Deep Neural Networks, and Adaptive Particle Swarm Optimization

Shenzhang Li, Zhenwei Geng*, Wenwei Su, Haoyu Ning, Xiaoping Zhao
Information Center, Yunnan Power Grid Co., Ltd., Kunming 650000, China
E-mail: zhenwei_geng@outlook.com
*Corresponding author

Keywords: random forest, deep neural network, adaptive particle swarm optimization, IoT data security, dynamic redundant storage

Received: April 25, 2025

With the increasing deployment of IoT systems, secure and efficient data storage has become a critical challenge. This paper proposes a multi-model secure redundant storage approach for IoT data by integrating Random Forest (RF), Deep Neural Network (DNN), and Adaptive Particle Swarm Optimization (APSO), refers to the complete proposed system that integrates Random Forest (RF) for feature extraction, Deep Neural Network (DNN) for anomaly detection and sensitivity classification, and Adaptive Particle Swarm Optimization (APSO) for dynamic storage strategy adjustment. The RF extracts key features from high-dimensional data, DNN detects and classifies anomalies, and APSO dynamically adjusts storage parameters for optimized redundancy. The model was evaluated on the SmartHomeIoTData-v1.0 dataset, comprising 1,000 devices and over 1,000,000 data entries across temperature, humidity, and status metrics. Compared to baseline models (KNN, SVM), our approach improves accuracy from 90% to 95%, increases storage resource utilization to 75%, and reduces data loss probability to 0.01%. These results demonstrate enhanced system security, efficiency, and responsiveness on resource-constrained devices.

Povzetek: Večmodelni pristop RF+DNN+APSO omogoča bolj kvalitetno varno redundantno shranjevanje IoT podatkov, saj presega KNN in SVM po točnosti, učinkovitosti shranjevanja ter zmanjšanju verjetnosti izgube podatkov.

1 Introduction

With the rapid development of IoT technology, various smart devices have sprung up, and the application scenarios of IoT are becoming increasingly diverse. From smart homes to industrial automation, from smart transportation to environmental monitoring, IoT technologies are increasingly integrated into both domestic and industrial applications [1, 2]. The popularization of this technology has brought great convenience and efficiency to society. However, with the continuous increase in the number of devices and the explosive growth of data volume, the data security problem of IoT has become increasingly prominent, gradually becoming an important bottleneck restricting its widespread application [3]. External network attacks are emerging in an endless stream, such as distributed denial of service (DDoS) attacks, man-in-the-middle attacks, and malware intrusions. These attacks not only lead to data leakage, tampering, or even loss, but also seriously interfere with the normal operation of IoT systems. For example, attackers may obtain users' privacy information by invading smart home systems, or cause production accidents by tampering with industrial production data [4]. In addition, due to the diversity and complexity of IoT devices, internal security management also faces severe challenges. Security vulnerabilities in devices, improper configuration, and weak user security awareness have further exacerbated the data security risks of IoT [5].

In this context, dynamic redundant storage technology is gradually regarded as an important means to ensure data security. Although traditional static redundant storage methods can prevent data loss to a certain extent, they are powerless in the face of dynamic changes in the IoT environment [6]. In contrast, dynamic redundant storage technology can flexibly adjust storage strategies based on factors such as data importance, access frequency, and real-time status of the system. It can not only effectively reduce the risk of data loss, but also improve the utilization efficiency of storage resources. The core of this technology is to achieve more efficient data protection and management through in-depth analysis and dynamic optimization of data

characteristics, thereby meeting the complex and diverse security needs of the IoT [7, 8].

In order to better solve the problem of secure storage of IoT data, this study proposes a dynamic redundant storage method that combines random forest algorithm, deep neural network model and adaptive particle swarm optimization algorithm. With its excellent classification and prediction capabilities, the random forest algorithm can accurately analyze massive IoT data, help identify key data and potential security threats, and provide a scientific basis for the formulation of redundant storage strategies. The deep neural network model can deeply explore the complex correlation characteristics between data and show great potential in security protection capabilities. As an efficient optimization tool, the adaptive particle swarm optimization algorithm can dynamically adjust storage parameters to further improve the flexibility and reliability of storage strategies. By organically combining these three technologies to form a collaborative intelligent architecture, it is expected to break through the limitations of traditional methods and provide a new solution for the secure storage of IoT data. This research can not only enhance the confidence of IoT users in data security, but also provide strong support for promoting the widespread application of IoT technology in various industries.

Although existing studies have explored the application of random forests, deep neural networks, and adaptive particle swarm optimization algorithms in different aspects, in the field of IoT data storage security, there is still a lack of research on organically combining these three and optimizing them for the dynamic, complex, and resource-constrained characteristics of the IoT. This study innovatively integrates the efficient feature analysis capabilities of random forests, the powerful nonlinear modeling capabilities of deep neural networks, and the dynamic optimization capabilities of adaptive particle swarm optimization algorithms to form a collaborative intelligent architecture to address the unique challenges faced by IoT data storage security. Compared with previous studies, this solution is not just a simple technical superposition, but also emphasizes the collaborative working mechanism between various technologies to achieve a more efficient, flexible, and reliable redundant storage strategy.

This study is guided by two primary hypotheses: (1) Multi-model feature integration using RF and DNN improves anomaly detection performance in high-dimensional IoT data compared to single models; and (2) APSO-driven dynamic parameter adjustment reduces data loss and improves resource utilization under resource-constrained environments. These hypotheses shape the design and evaluation criteria of the proposed system.

This study presents the following key contributions:

Integrated Multi-Model Architecture: We propose a novel system that integrates Random Forest (RF), Deep Neural Network (DNN), and Adaptive Particle Swarm Optimization (APSO) into a unified dynamic redundant storage framework for IoT data.

Feature-Driven Sensitivity Mapping: A fusion mechanism is designed to translate feature importance and anomaly detection outputs into actionable data sensitivity levels, which directly guide storage allocation.

Trust-Aware Redundancy Adjustment: We incorporate a data trust evaluation mechanism into the optimization pipeline, allowing dynamic adaptation of redundancy levels based on data source reliability.

Deployment-Oriented Design: The system is optimized for deployment on resource-constrained IoT devices using model compression, quantization, and reduced-complexity training without sacrificing predictive performance.

Comprehensive Evaluation: We conduct extensive experiments using a large-scale benchmark dataset, comparing against five baseline methods, and demonstrating superior accuracy (95%), recall (90%), and storage efficiency (75%).

2 Relevant theoretical basis

2.1 IoT data security

Internet of Things of Things, as an emerging technology paradigm, IoT is being widely used in various industries. IoT realizes the automatic collection, transmission and processing of information by connecting a large number of devices, but it also arouses widespread concern about data security. In the IoT environment, the number of devices is huge and the types are diverse, which leads to a significant increase in security risks in the data transmission process. Common threats include unauthorized access, data tampering, eavesdropping, and denial of service attacks [9, 10].

In order to protect the security of IoT data, the industry and academia have proposed a variety of technical means, such as encryption technology, access control mechanism and blockchain. Among them, encryption technology has become one of the core technologies of IoT security by ensuring the confidentiality and integrity of data during transmission and storage. In addition, distributed ledger technologies such as blockchain provide a reliable record-keeping method for the IoT, but their high computing and storage overheads limit their application in resource-constrained devices. On the other hand, the heterogeneity and distributed architecture of IoT devices pose challenges to traditional network security solutions. Existing centralized security strategies are difficult to adapt to the distributed characteristics of the IoT, which has prompted the development of security technologies based on artificial intelligence and machine learning. These technologies use data-driven methods to effectively detect and respond to complex network threats [11, 12].

In addition to the common IoT data security technical means mentioned above, some emerging IoT

storage solutions are also worth paying attention to recently. For example, blockchain-based systems have been widely studied in IoT data storage. Blockchain technology ensures the immutability and traceability of data through decentralized distributed ledgers. In the IoT environment, the data generated by the device can be directly recorded on the blockchain, and each data block contains the hash value of the previous data block, forming a chain structure. For example, in a small network consisting of 50 IoT devices, about 10,000 pieces of data generated every day are recorded on the blockchain. After 1 month of operation, the integrity of the data is effectively guaranteed, and no data tampering incidents have occurred. However, blockchain technology also faces some challenges, such as high computing and storage overhead. In the above small network, each node needs to consume an additional storage space of about 500MB for storing blockchain data. The consumption of computing resources causes the average response time of the device to be extended by about 10 seconds, which may become a bottleneck for applications on resource-constrained IoT devices. Another emerging solution is federated learning, which allows different participants to collaboratively train models without sharing original data. In the IoT data storage scenario, federated learning can enable multiple IoT devices to train and store related models locally using their own data, and then jointly optimize the storage strategy by exchanging model parameters. For example, in a smart home system, IoT devices from five different families use federated learning to collaboratively train storage models, which increases the storage resource utilization by about 15% compared with individual training. However, federated learning also has problems such as high communication overhead and the speed of model convergence being affected by differences in device performance. In the above smart home system, due to the different performance of each family device, the model convergence time is about 30% longer than the ideal situation. Compared with these recent IoT storage solutions, the dynamic redundant storage solution based on random forest, deep neural network and adaptive particle swarm optimization algorithm proposed in this study not only ensures data security and storage efficiency, but also pays more attention to the adaptability of resource-constrained devices and the optimization of computing overhead, and has unique advantages and application prospects. In the same resource-constrained smart home device environment, the storage resource utilization of this solution is about 20% higher than that of the blockchain-based solution, and the computing overhead is reduced by about 30%. When dealing with differences in device performance, the model converges faster and has higher stability.

In addition to traditional methods such as K-nearest neighbor (KNN) and support vector machine (SVM) compared in this study, recent IoT storage solutions such as blockchain systems and federated learning are also crucial in the field of IoT data storage. Blockchain ensures data integrity and immutability due to its decentralized and distributed nature. In the IoT environment, the data generated by each device is stored in blocks, which are connected to the previous block through cryptographic hashing, making malicious tampering extremely difficult to achieve. For example, in a smart grid system, blockchain can record a large amount of energy consumption data of smart meters. Tampering with the data requires changing all subsequent blocks, which is almost impossible in a mature blockchain network. However, blockchain storage requires high computing resources for mining and maintaining distributed ledgers, which poses challenges to resource-constrained IoT devices.

Federated learning provides a new way for IoT data storage, which allows multiple IoT devices or data owners to collaboratively train storage-optimized machine learning models without sharing original data. Taking multiple smart home optimized storage strategies as an example, each device trains the model locally and only exchanges model parameters, which not only protects data privacy but also integrates multi-data knowledge. However, federated learning has the problem of high communication overhead, and the differences in computing power, network connection and data distribution of devices will significantly affect the convergence speed of the model.

Compared with these recent solutions, the dynamic redundant storage solution proposed in this study, which combines random forest, deep neural network and adaptive particle swarm optimization, is more suitable for resource-constrained IoT devices, while maintaining high data security and storage efficiency, and seeking a balance between computing requirements and performance, providing a practical solution for IoT data storage in real scenarios.

Table 1: Comparative analysis of related IoT data storage techniques

Method	Strengths	Limitations	Suitability for IoT Storage
Blockchain [6, 12]	Immutability, decentralized trust	High computational/storage overhead	Limited for resource-constrained IoT
Federated Learning [13, 14]	Preserves privacy, supports distributed learning	Slow convergence, communication cost	Good for sensitive data, needs optimization
Proposed (RF+DNN+APSO)	High accuracy, dynamic optimization	Model complexity	Highly suitable with model compression

As shown in Table 1, compared to prior work, our method addresses key challenges in resource-limited IoT scenarios by combining multi-model learning, lightweight optimization, and secure storage. Previous approaches either focus on security (e.g., blockchain) or privacy (e.g., FL), but do not simultaneously optimize redundancy, performance, and adaptability.

2.2 Random forest model

Random forest is an ensemble learning method that builds multiple decision trees and combines their prediction results to form a model with high accuracy and stability. It is particularly suitable for processing high-dimensional data and nonlinear problems, which makes it an important tool in IoT data analysis and security. The basic principle of random forest is to generate multiple independent decision tree models by randomly selecting training samples and features. Each decision tree makes predictions separately, and finally outputs the comprehensive results by voting or weighting. The advantage of this method is that [15, 16] by introducing randomness, it can effectively reduce the risk of model overfitting and improve the generalization ability of unknown data. In the field of IoT security, random forest is widely used in anomaly detection and intrusion identification. For example, by learning network traffic features, random forest can accurately identify potential attack behaviors. In addition, the model can process large-scale, high-dimensional IoT data and provide feature importance analysis to provide guidance for the formulation of security policies. However, random forest also faces some limitations in practical applications, such as limited support for real-time data processing and the increase of model complexity with the increase of data scale [17, 18].

When using the random forest model to process IoT data, privacy issues cannot be ignored. During the training process, the random forest model may be exposed to a large amount of data containing user privacy information. In order to protect data privacy, this solution adopts a privacy protection technology based on homomorphic encryption, as in reference [19]. Before the data is input into the model, the sensitive data is homomorphically encrypted. For example, the user's home address and other information are encrypted using the Paillier homomorphic encryption algorithm. This allows the model to calculate on the encrypted data, and the results are consistent with those calculated on the plaintext data. In this way, even if the model is maliciously attacked or the data is illegally obtained, the attacker cannot directly obtain the original private data. At the same time, during the model training process, strict permission control is implemented on the access and use of the data. Only authorized administrator accounts and specific data analysis programs can operate on the data, further ensuring the privacy security of the data. In a simulated attack test, the attacker attempted to obtain the encrypted home address data by invading the model. After 10 hours of cracking attempts, no valid information was obtained, proving the effectiveness of this privacy protection technology.

2.3 Deep neural network model

Deep Neural Network Neural Networks, Deep neural networks (DNNs) are an extended form of artificial neural networks. By increasing the number and complexity of hidden layers, they have stronger feature expression capabilities. Deep neural networks have achieved remarkable results in various fields in recent years. Their excellent nonlinear modeling capabilities and ability to process complex data have made them a hot technology in IoT data security research. The architecture of a deep neural network consists of an input layer, multiple hidden layers, and an output layer. Through the weight connections between layers, the model can extract high-order features of the data layer by layer. In the field of IoT data security, deep neural networks are often used for malicious behavior detection, traffic classification, and decoding and analysis of encrypted data [13]. For example, by training a neural network model to identify normal and abnormal communication patterns, potential attack behaviors can be quickly detected [20]. Compared with traditional machine learning models, deep neural networks can automatically learn the potential features in the data without relying on complex feature engineering, which has significant advantages in IoT scenarios where data features are diverse and dynamically changing [21]. However, the application of deep neural networks also has some challenges, such as the need for a large amount of labeled data during the training process, high requirements for computing resources, and potential threats of adversarial sample attacks. In order to improve the applicability of deep neural networks, researchers have proposed a variety of improvement methods, such as reducing the reliance on labeled data through transfer learning, optimizing the model structure to reduce computational costs, and combining generative adversarial networks (GANs) to enhance the robustness of the model [22, 23]. In summary, the research on IoT data security requires the combination of multiple technical means, among which random forests and deep neural networks have shown unique advantages in structured data analysis and complex pattern recognition, respectively. In practical applications, a hybrid strategy combining multiple models may be an effective way to address IoT security challenges.

Privacy protection is also crucial for deep neural network models. Since deep neural networks require a large amount of data for training, some of the data may contain sensitive information of users. To prevent privacy leakage, this solution combines the idea of reference [14] and adopts a method that combines federated learning with differential privacy. Under the federated learning framework, each participant (such as different IoT devices or data owners) trains the model locally and only uploads the parameter update information of the model

without directly sharing the original data. At the same time, during the parameter update process, noise that complies with the differential privacy mechanism is added, such as Laplace noise. The scale parameter of the noise is reasonably set according to the data sensitivity and privacy budget. Even if the parameter update information is leaked, it is difficult to infer the privacy content of the original data from this information. In this way, it is ensured that the deep neural network model can make full use of multi-source data for training, and the privacy security of the data is effectively protected, which meets the strict requirements for data privacy protection in the IoT environment. In actual tests, the federated learning training process of 10 participants was monitored. When noise was added, the model accuracy only dropped by about 2%, but the privacy protection effect was significantly improved, and it successfully resisted multiple privacy inference attacks on parameter update information.

Table 2: Comparison of related works on secure IoT data storage

Method	Security Feature	Optimization Strategy	Device Support	Limitation
Blockchain-based Storage [6]	Tamper-resistance	Static Redundancy	Limited	High cost, slow convergence
Federated Learning [12]	Privacy-preserving	Collaborative Training	Moderate	High communication overhead
DRL-based Compression [17]	Compression + Encryption	Deep RL Optimization	High	Complex model, high resource usage
This Work (RF+DNN+APSO)	Anomaly-aware + Trust	Dynamic Particle Swarm Opt.	Low-to-High	Requires initial model calibration

As shown in Table 2, our method combines anomaly detection, trust modeling, and adaptive optimization, balancing security, performance, and deployability. Compared to existing methods, it offers improved adaptability with reduced computational burden.

3 Design of a secure dynamic redundant storage solution for IoT data based on multi-model fusion

3.1 Overall solution architecture

This chapter proposes a random forest-based Forest, RF), Deep neural networks (Deep Neural Networks, DNN and Adaptive Particle Swarm Optimization Particle Swarm Optimization, the proposed IoT data security dynamic redundant storage solution based on Random Forest Application Service (APSO) is designed to address the complex challenges faced by data security storage in the IoT environment. The solution comprehensively utilizes the efficient feature analysis capabilities of random forests, the deep pattern recognition capabilities of deep neural networks, and the dynamic optimization capabilities of adaptive particle swarm optimization algorithms to provide a flexible, efficient, and reliable redundant storage mechanism for IoT data. The overall architecture of the solution is divided into three main modules: data feature analysis, key data identification, and dynamic storage strategy optimization. The modules complement each other to form a coordinated and efficient system that can adapt to the complex and changing IoT data environment in real time [24, 25].

Among the many technologies that can be used for IoT data storage security, it is no accident that random forest, deep neural network and adaptive particle swarm optimization algorithm are selected for combination. IoT data is characterized by high dimensionality, diversity, dynamic changes, and extremely high requirements for storage security and efficiency. With its efficient feature analysis ability and good adaptability to high-dimensional data, random forest can quickly screen out key features from massive IoT data and reduce the complexity of subsequent processing. The powerful nonlinear modeling ability of deep neural network enables it to perform well in processing complex data patterns and anomaly detection, which is very suitable for the analysis needs of complex data in IoT environment. As an efficient optimization tool, adaptive particle swarm optimization algorithm can dynamically adjust storage parameters according to the real-time status of data and the operation of the system, and optimize storage strategy to meet the dynamic changes of IoT environment. Taking into account the characteristics of these technologies and the actual needs of IoT data storage security, combining them together to form an organic whole is expected to break through the limitations of traditional methods and provide a more effective solution for the secure storage of IoT data. The data feature analysis module uses the random forest algorithm to perform comprehensive feature extraction and classification of IoT data. IoT data is diverse and usually has high dimensionality and heterogeneity, such as sensor data, device logs, and user interaction data. Random forests build multiple decision tree models and use feature randomness and sample randomness to reduce the risk of overfitting of the model

and improve its generalization ability for different types of data. Through quantitative analysis of data importance (such as the Gini coefficient or information gain), random forests can effectively identify key features and provide a scientific basis for subsequent key data screening and storage strategies.

The key data identification module is based on deep neural networks (DNNs) to deeply mine and classify complex patterns in data. DNNs have powerful nonlinear modeling capabilities and can automatically learn deep features in data without relying on complex artificial feature engineering. Its multi-layer architecture captures potential patterns and trends from input data by extracting high-order features layer by layer. In this solution, deep neural networks can not only identify abnormal behaviors, but also quickly adjust models for dynamically changing data environments, thereby improving the accuracy and robustness of key data identification. In addition, combined with the adaptability of deep learning technology to large-scale data, the DNN module lays the foundation for intelligent decision-making in dynamic redundant storage of the Internet of Things.

Finally, the dynamic storage strategy optimization module uses the adaptive particle swarm optimization algorithm (APSO) to dynamically adjust the redundant storage parameters. This module establishes an optimization objective function based on the data importance and real-time system status analyzed in the first two parts, taking into account the risk of data loss, storage cost, and the utilization of redundant resources. APSO searches for the global optimal solution by dynamically adjusting the position and speed of the particle swarm, thereby realizing the dynamic adjustment of the storage strategy. For example, for critical data with high access frequency, its redundant storage nodes can be increased; for data with lower importance, redundant copies can be appropriately reduced to improve the utilization efficiency of storage resources.

Compared with the use of these technologies alone or in other combinations, the combination of random forest, deep neural network and adaptive particle swarm optimization algorithm used in this study has significant advantages. Random forest can quickly and accurately extract key features from high-dimensional and diverse IoT data, providing a solid foundation for subsequent analysis. Based on the features extracted by random forest, deep neural network can more accurately identify abnormal patterns and key information in data through powerful nonlinear modeling capabilities, which is difficult to achieve with traditional single models. The adaptive particle swarm optimization algorithm dynamically optimizes storage parameters based on the output results of the first two. This optimization method can adapt to changes in the IoT environment in real time and achieve efficient use of storage resources. For example, in the face of dynamic situations such as sudden growth in data traffic or equipment failure, this combination can respond quickly and adjust storage strategies to ensure secure storage and efficient access to data, while other combinations may not be able to respond so flexibly.

With the synergy of the three modules, this solution further enhances the adaptability and decision-making ability of the system through model fusion. Specifically, the output results of random forest and deep neural network are jointly analyzed according to the weighted fusion strategy to identify key data with higher accuracy and stability. The weight parameters of the fusion strategy can be dynamically adjusted through experimental optimization to ensure the best performance in different scenarios. Finally, the system realizes dynamic and secure redundant storage of IoT data, which not only improves the security and reliability of data storage, but also significantly reduces storage resource overhead.

Compared with some previous methods that use similar hybrid technologies, the scheme of this study is unique. For example, some previous studies may simply apply these technologies in sequence without fully considering their mutual influence and synergy. This scheme uses a carefully designed model fusion strategy to enable random forests, deep neural networks, and adaptive particle swarm optimization algorithms to cooperate and complement each other in the entire system. In the feature extraction stage, random forests not only provide effective features for deep neural networks, but also provide decision-making basis for adaptive particle swarm optimization algorithms when adjusting storage parameters. In the process of anomaly detection and storage strategy optimization, the output of deep neural networks is combined with the results of random forests to jointly guide the search direction of adaptive particle swarm optimization algorithms. This close coordination mechanism has not been fully reflected in previous studies, which makes this scheme have higher performance and adaptability in the actual application of IoT data storage security.

The proposed system is composed of three interdependent modules: (1) the Random Forest (RF) feature extraction module, (2) the Deep Neural Network (DNN) anomaly detection module, and (3) the Adaptive Particle Swarm Optimization (APSO) dynamic storage optimizer. The system workflow starts with incoming IoT data being fed into the RF module, which extracts and ranks features based on importance. These features are passed to the DNN module, which identifies and classifies anomalous or sensitive data points. The classification results and the feature importance scores are then forwarded to the APSO module, which dynamically computes optimal storage parameters based on current system load and security requirements. A real-time feedback loop allows APSO outputs to influence RF thresholds and DNN sensitivity tuning.

3.2 Data feature analysis module

The data feature analysis module extracts and classifies the IoT data through the random forest model. Assume that the feature matrix of the IoT data is $X = \{x_1, x_2, \ldots, x_n\}$, the corresponding label is $Y = \{y_1, y_2, \ldots, y_n\}$. The random forest model constructs k decision trees $T_1, T_2, \ldots, T_k$. Each tree is trained based on different feature subsets and training data, and its output is obtained by voting to obtain the final classification result, as shown in Formula 1. The importance of features is measured by the Gini coefficient. The specific calculation is as shown in Formula 2.

$$\hat{y} = \text{majority_vote}\{T_1(x), T_2(x), \ldots, T_k(x)\} \quad (1)$$

$$G(X_j) = \sum_{i=1}^{c} p_i(1 - p_i) \quad (2)$$

In Formula 2, X_j is feature j, p_i is the probability distribution of category i under this feature. By sorting all features, the most important features can be selected for redundant storage decision.

Compared with other common feature selection methods, such as principal component analysis (PCA) and recursive feature elimination (RFE), random forest has unique advantages in the IoT data storage security scenario of this study. PCA mainly projects high-dimensional data into low-dimensional space through linear transformation to achieve the purpose of dimensionality reduction, but it may lose some important nonlinear information. In IoT data, many key features are often hidden in complex nonlinear relationships. Random forest can better capture the nonlinear characteristics in the data by constructing multiple decision trees and using random feature selection and random sample sampling, and has stronger adaptability to different types of data. Although RFE can select key features by recursively eliminating unimportant features, it requires a pre-defined evaluation index and has high computational complexity. Random forest does not need to pre-define complex evaluation indicators. Its decision tree-based structure can intuitively give the importance ranking of features and has higher computational efficiency. In addition, when processing large-scale, high-dimensional IoT data, random forest can better deal with data noise and missing value problems, ensure the accuracy and stability of feature selection, and thus provide a more reliable basis for subsequent key data identification and storage strategy formulation

Table 3 lists the top 10 features ranked by their Gini importance scores computed from the Random Forest model. The most informative features include device temperature fluctuations, frequency of status changes, and recent fault alarms.

Table 3: The top 10 features

Rank	Feature Name	Gini Importance Score
1	Temperature_STD	0.182
2	Status_Change_Frequency	0.165
3	Fault_Alarm_Recent	0.142
4	Humidity_STD	0.123
5	Power_On_Duration	0.095
6	Signal_Loss_Incidents	0.089
7	Temperature_Mean	0.067
8	Energy_Consumption_Peak	0.055
9	User_Interaction_Frequency	0.046
10	Maintenance_History_Score	0.036

These features significantly influenced both anomaly detection and storage priority decisions. Features with high variance and high predictive value are prioritized for redundancy under the APSO policy.

3.3 Key data identification module

In the IoT data security dynamic redundant storage solution, the identification of key data is a key step to achieve accurate storage and resource optimization. This module uses deep neural networks (Deep Neural Networks, The nonlinear modeling capabilities of Deep Neural Networks (DNNs) are used to conduct in-depth analysis and classification of key data in IoT data, thus providing a reliable foundation for subsequent optimization of dynamic storage strategies. With their powerful feature extraction capabilities and ability to recognize complex patterns, deep neural networks can adapt to the high-dimensional, dynamic and heterogeneous characteristics of data in IoT environments, effectively improving the recognition efficiency and accuracy of key data.

3.3.1 Structural design of deep neural network models

A deep neural network consists of an input layer, multiple hidden layers, and an output layer. It can extract high-order features of data layer by layer and realize the

mapping from input features to output classification. Assume that the input data is a feature vector $x \in \square^n$, represents the data characteristics of a device or scene in the Internet of Things, such as traffic characteristics, time series patterns, or user interaction information. The weight matrix and bias term of the model are expressed as $\{W^{(l)}, b^{(l)}\}_{l=1}^{L}$, where L is the number of network layers. The output of the hidden layer can be calculated by formula 3.

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

In formula 3, $f(\cdot)$ is the activation function. Common activation functions include ReLU (Rectified Linear Unit), Sigmoid and Tanh Etc. ReLU The activation function is widely used in deep learning tasks due to its high computational efficiency and less gradient vanishing problem. Its expression is Formula 4.

$$f(x) = \max(0, x) \tag{4}$$

In the IoT environment, anomaly detection is crucial to ensure data security. Compared with traditional anomaly detection methods such as support vector machines (SVM), autoencoders, and long short-term memory networks (LSTM), the choice of deep neural networks in this study is fully reasonable. SVM performs well when processing linearly separable data or data converted to linearly separable data through kernel functions, but its performance may be limited for complex and changeable nonlinear data patterns in the IoT. Autoencoders mainly detect anomalies by reconstructing data, and work well for scenarios with relatively regular data distribution, but their generalization ability may be insufficient when faced with the diversity and dynamic changes of IoT data. Although LSTM networks have advantages in processing time series data, their structure is relatively complex and the computational cost is high for feature extraction and anomaly detection of non-time series IoT data. Through multi-layer neuron structures and powerful nonlinear activation functions, deep neural networks can automatically learn deep-level features in data and have stronger adaptability and generalization ability for various complex data patterns in the IoT environment. In this study, deep neural networks can effectively further mine key information from the features extracted by random forests, accurately identify abnormal data, and provide a reliable basis for subsequent storage strategy optimization, which is difficult to achieve with other traditional methods.

The output layer adopts Softmax Function, which converts the final output of the network into a probability distribution, as shown in Formula 5. Among them, Softmax function is defined as Formula 6.

$$\hat{y} = \text{softmax}(W^{(L)}h^{(L-1)} + b^{(L)}) \tag{5}$$

$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^{C}\exp(z_j)}, \quad i = 1, 2, \ldots, C \tag{6}$$

C represents the number of categories. z_i is the prediction score for the ith category.

Table 4: Summarizes the architecture and training hyperparameters of the DNN model.

Layer Index	Type	Units	Activation	Dropout
Input	Dense	500	-	-
Hidden 1	Dense	256	ReLU	0.2
Hidden 2	Dense	128	ReLU	0.2
Hidden 3	Dense	64	ReLU	0.2
Output	Dense	5	Softmax	-

As shown in Table 4, the model is optimized using the Adam optimizer with an initial learning rate of 0.001 and an exponential decay schedule (decay rate = 0.96, decay steps = 1,000). Training was conducted over 50 epochs with a batch size of 64. Each epoch required approximately 3.8 minutes on a mid-range GPU (NVIDIA RTX 3060), and convergence was typically reached around epoch 30.

3.3.2 Model optimization and loss function

The training goal of a deep neural network is to optimize the model parameters so that the output prediction value $\hat{y}$ With the actual label y to this end, the cross-entropy loss function is used as the optimization objective, as shown in Formula 7.

$$\text{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(\hat{y}_{ij}) \tag{7}$$

In Formula 7, N represents the number of samples, C is the number of categories, y_{ij} It is a sample i. In category j the true label on 0 or 1), $\hat{y}_{ij}$ is the network's predicted probability for this category. By minimizing this loss function, the classification performance of the model can be gradually improved.

3.3.3 Data preprocessing and model training

In practical applications, IoT data often have problems such as noise, missing values, and uneven distribution. In order to improve the robustness and generalization ability of the model, data preprocessing is required, including data cleaning, standardization,

normalization, and data enhancement. Assume that the data feature vector is $x=\{x_1,x_2,\ldots,x_n\}$, the normalization can be expressed as Formula 8.

$$x_{i'}=\frac{x_i-\mu}{\sigma} \tag{8}$$

In Formula 8, μ and σ are the feature mean and standard deviation respectively.

After data processing is completed, the data is divided into training set, validation set and test set for model training, hyperparameter tuning and performance evaluation. In order to avoid overfitting of the model, regularization techniques (such as L2 Regularization) and Dropout Mechanism. L2, the goal of regularization is to minimize the loss function and the sum of squared parameters, as shown in Formula 9.

$$\mathrm{L}_{\mathrm{reg}}=\mathrm{L}+\lambda\sum_{i=1}^{n}w_i^2 \tag{9}$$

In Formula 9, λ is the regularization strength hyperparameter, w_i is the network weight.

The data was split into 70% training, 15% validation, and 15% testing sets. Training was conducted over 50 epochs with a batch size of 64. Early stopping was enabled with a patience of 7 epochs based on validation loss. A learning rate decay schedule was used to reduce overfitting. These settings were selected based on grid search optimization.

In the process of deep neural network (DNN) training, computational overhead is an important issue that needs to be paid attention to, especially for resource-constrained IoT devices. The training of DNN models involves a large number of matrix operations and parameter updates, and its computational complexity is closely related to the number of layers, the number of neurons, and the amount of training data. Taking the DNN model used in this study as an example, the model contains 5 hidden layers, and the number of neurons in each layer gradually decreases from 100 to 500. During the training phase, each forward propagation and backpropagation requires a large number of multiplication and addition operations. In order to evaluate its computational overhead, IoT devices of different sizes were tested in an experimental environment. On devices with lower configurations (such as devices with 1 GB memory and a single-core processor), it takes up to 30 minutes to complete a full training iteration, which may affect the real-time performance of the system. In order to reduce the computational overhead of DNN training, this solution uses some optimization technologies, such as model compression, which removes about 20% of redundant connections through pruning technology, and quantization technology, which converts parameter data types from 32-bit floating point numbers to 16-bit floating point numbers, thereby reducing the amount of calculation; at the same time, batch training is adopted to load 10,000 data that were originally loaded at one time in batches, with 1,000 data loaded each time, reducing memory usage and computing resource consumption. After these optimizations, the time to complete a training iteration on the same configuration device is shortened to 10 minutes, which to a certain extent alleviates the pressure of DNN training on the computing resources of IoT devices and improves the feasibility of the system in a resource-constrained environment.

While LSTM networks are well-suited for temporal data, the dataset used in this study is primarily composed of snapshot-based and event-driven features rather than continuous time series. An ablation study was performed comparing the current DNN architecture to a 2-layer LSTM with 128 units per layer. The LSTM model achieved 92.5% accuracy (vs. 95% for DNN) and took 2.3× longer per epoch to train. These results justify the choice of DNN for our resource-constrained IoT scenario.

In this study, the terms “critical data,” “anomalous data,” and “high-sensitivity data” are used consistently to denote data points that are identified as security-relevant or failure-prone. The DNN module classifies data into normal and anomalous categories, which are further refined by the fusion process using RF importance scores to assign a sensitivity level (high, medium, or low). Thus, high-sensitivity data correspond to anomalous or critical events with high feature importance, which then trigger increased redundancy in storage.

3.4 Dynamic storage strategy optimization module

The dynamic storage strategy optimization module is an important part of realizing the core functions of the IoT data security redundant storage solution. Particle Swarm Optimization, The APSO algorithm dynamically adjusts storage parameters according to the importance of data and system status to achieve the optimal balance between storage efficiency and security. The particle swarm optimization algorithm is an optimization method based on swarm intelligence. It has the characteristics of low computational complexity, simple implementation and strong global search capability. It is particularly suitable for solving dynamic storage optimization problems.

3.4.1 The basic principle of particle swarm optimization algorithm

The particle swarm optimization algorithm simulates the flying behavior of particles in the search space to find the optimal solution of the objective function. In this module, each particle represents a possible storage strategy, and its position $x_i(t)$. Corresponding to the current strategy parameters, speed $v_i(t)$ Indicates the moving direction and amplitude of the particle in the search space. The particle passes through the individual historical optimal position p_i and the group global optimal position. g to adjust the speed and

position, the iterative formulas are shown in Formula 10 and Formula 11.

$$v_i(t+1) = \omega v_i(t) + c_1 r_1[p_i - x_i(t)] + c_2 r_2[g - x_i(t)] \quad (10)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (11)$$

In Formula 10 and Formula 11, ω Inertia weight controls the particle's dependence on the current velocity. ω It helps in global search, while smaller ω It is conducive to local convergence; c_1, c_2 is the learning factor, which measures the particle's ability to learn its own historical optimal position and the group's global optimal position; r_1, r_2 For the value in [0, 1] The random number is used to introduce randomness and avoid falling into the local optimum.

3.4.2 Introduction of adaptive strategy

In order to adapt to the dynamic changes of data characteristics and system status in the IoT environment, this module adopts an adaptive strategy to adjust the inertia weight. ω and learning factor c_1, c_2 Make dynamic adjustments. The specific formula is Formula 12.

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{\text{iter}_{\max}} \cdot \text{iter} \quad (12)$$

In Formula 12, $\omega_{\max}$ and $\omega_{\min}$ are the initial value and minimum value of the inertia weight, iter is the current iteration number, $\text{iter}_{\max}$ This strategy enables the algorithm to have a strong global search capability in the early stage, and gradually enhance the local search capability in the later stage, thereby improving the optimization effect of the algorithm.

3.4.3 Objective function design

The goal of dynamic storage strategy optimization is to minimize the comprehensive cost function J, taking storage costs into account C_{storage} And the data loss probability P_{loss}, specifically as Formula 13.

$$J = \alpha \sum_{i=1}^{N} C_{\text{storage}}(x_i) + \beta \sum_{i=1}^{N} P_{\text{loss}}(x_i) \quad (13)$$

Among them, N is the amount of data, x_i For particles i The location (i.e. storage parameters) of α and β is the weight coefficient, which is used to balance storage cost and data security; $C_{\text{storage}}(x_i)$ Indicates storage parameters x_i Storage costs under $P_{\text{loss}}(x_i)$ Indicates the probability of data loss, which decreases as storage redundancy increases.

Optimize storage parameters by dynamically adjusting particle positions x_i, which can effectively reduce the overall cost J. After the optimization is completed, the dynamic storage strategy optimization module outputs the optimal storage parameters x, including information such as storage device selection, number of data copies, and storage location. This parameter will guide the storage system to dynamically and redundantly store IoT data, providing high security for key data while reducing storage resource overhead.

The adaptive particle swarm optimization algorithm (APSO) also has a computational overhead problem in the process of dynamically adjusting storage parameters. The APSO algorithm needs to calculate the speed and position updates of particles in each iteration, which involves multiplication and addition operations of multiple parameters. For resource-constrained IoT devices, frequent calculations may lead to device performance degradation. In order to analyze the computational overhead of the APSO algorithm, the time and resource usage of running the APSO algorithm on IoT devices of different sizes were monitored in the experiment. When the number of particles is set to 50 and the search space is 10 dimensions, running an APSO algorithm iteration on an ordinary IoT device consumes an average of 10MB of memory and 5 seconds. As the number of particles increases to 100 and the search space increases to 20 dimensions, the computational overhead increases significantly, the memory consumption rises to 20MB, and the time is extended to 10 seconds. In order to optimize the computational efficiency of the APSO algorithm on IoT devices, this scheme improves the algorithm. Under the premise of ensuring the algorithm's search performance, unnecessary calculation steps are reduced. For example, when updating the particle position, a simplified update formula is used. By introducing some heuristic rules, the approximate optimal solution is quickly found, thereby reducing the amount of calculation. In addition, the algorithm parameters are reasonably set, such as linearly reducing the inertia weight from 0.9 to 0.4, and setting the learning factor to 1.5 and 2.0 respectively, so that it can run with low computational overhead in different IoT device environments. After the improvement, in the same complex scenario, the memory consumption is reduced to 15MB and the running time is shortened to 7 seconds, which improves the applicability of the algorithm in resource-constrained environments.

The computational complexity of APSO was evaluated in terms of floating-point operations per second (FLOPs). For 50 particles and a 10-dimensional parameter space, each iteration required approximately 10^6 FLOPs. Compared to a static heuristic method, APSO introduced a 20% increase in runtime (7 s vs. 5.8 s) but reduced data loss by 66% [26], as shown in Table 5.

Table 5: Summarizes parameter adjustments across three scenarios

Scenario	Redundancy (r)	Storage Cost (¥)	Data Loss (%)
Low Load (Static)	2	500	0.05
Dynamic (APSO)	3	400	0.01
High Risk (APSO)	4	450	0.005

The sensitivity levels (high, medium, low) generated by the RF-DNN fusion module are mapped to specific storage parameter configurations as inputs to the APSO optimization process. Specifically, each sensitivity level corresponds to a baseline redundancy level (e.g., r = 4 for high, r = 3 for medium, r = 2 for low), which serves as an initial position vector x_i for particles. APSO then adjusts these parameters in the optimization space considering additional constraints such as network bandwidth, device capacity, and recent fault rates to minimize storage cost and data loss probability.

3.5 Model fusion strategy

The model fusion strategy in this scheme integrates the prediction results of the random forest (RF) model and the deep neural network (DNN) model, and generates the final recognition results of key data through weighted fusion.

The random forest model is good at processing structured data, with high stability and strong feature importance analysis capabilities; the deep neural network model performs well in unstructured data and complex pattern recognition. The fusion of the two helps to improve the overall performance of key data recognition. The fusion formula is Formula 14.

$$\hat{y} = \lambda \hat{y}_{\text{RF}} + (1-\lambda)\hat{y}_{\text{DNN}} \tag{14}$$

In Formula 14, $\hat{y}_{\text{RF}}$ and $\hat{y}_{\text{DNN}}$, these are the prediction results of random forest and deep neural network respectively; λ To fuse the weight parameters, they are adjusted experimentally to achieve the best performance.

The output of the fusion module plays a core role in the optimization of dynamic storage strategies and provides a key basis for the secure storage of IoT data. Based on the identified data sensitivity level, the system generates three types of storage strategies: high, medium, and low. $y \in \{\text{high}, \text{medium}, \text{low}\}$, according to different sensitivity levels, the system allocates different redundant resources and storage strategies:

(1) Highly sensitive data: For highly sensitive data $y = \text{high}$, the system allocates more redundant resources and uses high-security storage devices. For example, the number of redundant storage copies r Will increase to reduce the probability of data loss $P_{\text{loss}}(r)$. Number of storage copies r and security level s the relationship between can be expressed by the following formula, specifically Formula 15.

$$\begin{aligned} &r_{\text{high}} = \arg\max_r P_{\text{security}}(r,s) \\ &\text{where} \quad P_{\text{security}}(r,s) \text{ increases with } r \end{aligned} \tag{15}$$

(2) Medium sensitive data: For medium sensitive data ($y = \text{medium}$), combining cost and security, the system prefers hybrid storage mode. C_{storage} and data loss risk P_{\text{loss}} The comprehensive objective function J It can be expressed as Formula 16.

$$J_{\text{medium}} = \alpha C_{\text{storage}}(x_i) + \beta P_{\text{loss}}(x_i) \tag{16}$$

In Formula 16, α and β is the weight coefficient, x_i is the particle position (indicating the choice of different storage strategies).

(3) Low-sensitivity data: For low-sensitivity data $y = \text{low}$, a basic redundancy solution is used to save resources. r Fewer, lower storage costs, and relatively acceptable risk of data loss

(4) In IoT networks, the trust of data sources is crucial to the formulation of storage strategies. Based on the research results of [19], this solution further considers the trust/reputation management mechanism. For data from high-trust sources, such as data generated by devices that have undergone strict identity authentication and have been running stably for more than 6 months, the redundancy level can be appropriately reduced from 3 redundant copies to 2 during storage to improve the utilization efficiency of storage resources while ensuring data security. On the contrary, for data from low-trust sources, such as data from newly connected devices or devices that have experienced security risks (such as 3 or more security incidents in the past 3 months), redundant storage nodes are added to increase the redundancy level from 2 to 4 to reduce the risk of data loss or tampering. By dynamically adjusting storage parameters in this way, it is possible to better adapt to changes in the trust of data sources in IoT networks and further improve the security and reliability of data storage. For example, in a simulation experiment, 100 groups of high-trust data and 100 groups of low-trust data were stored using the above strategy. The results showed that the storage resources of high-trust data were saved by about 20%, and the data loss rate of low-trust data was reduced from the original 5% to 1% when it was subjected to simulated attacks.

At the same time, the system realizes real-time monitoring and optimization through a dynamic feedback mechanism. Assume that the system state changes are determined by the parameters θ Description, storage strategy optimization based on the real-time status of the system $\theta(t)$ and data importance changes. For example, the adjustment of dynamic storage strategy can be achieved by updating the position and speed of particles through particle swarm optimization algorithm, as shown in Formula 17 and Formula 18.

$$v_i(t+1) = \omega v_i(t) + c_1 r_1[p_i - x_i(t)] + c_2 r_2[g - x_i(t)] \quad (17)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (18)$$

In Formula 17 and Formula 18, $v_i(t)$ represents the velocity of the particle, p_i is the particle's best historical position, g is the global optimal position. Through this optimization process, the system can dynamically adjust the storage policy to cope with different workloads and data security requirements.

In addition, the system presents the results through a visual interface, allowing administrators to view storage status and security policies in real time and adjust storage parameters as needed. Encryption technology and access control are closely integrated with storage policies to ensure the security of highly sensitive data while optimizing the utilization efficiency of storage resources. Ultimately, through intelligent and dynamic management mechanisms, the system can ensure the secure storage and efficient use of IoT data.

3.6 Deployment feasibility on resource-constrained IoT devices

In this section, it is stated that: “Considering the actual situation that many IoT devices are resource-constrained, this solution fully considers the feasibility of deployment on such devices when designing. A series of optimization measures are taken to address the problem of high computing resource requirements of deep neural networks and adaptive particle swarm optimization algorithms. First, in terms of model structure, the deep neural network is lightweight designed to reduce the computational complexity while ensuring model performance by reducing unnecessary hidden layers and the number of neurons. For example, after experimental comparison, the original 50 hidden layers are reduced to 8 layers, and the connection method of neurons is reasonably adjusted, so that the model can still maintain a high accuracy on resource-constrained devices. Secondly, in the implementation of the adaptive particle swarm optimization algorithm, a simplified calculation strategy is adopted. In each iteration, the amount of calculation for particle position and velocity updates is reduced, and some heuristic rules are introduced to quickly find the approximate optimal solution, thereby reducing the algorithm's demand for computing resources. In addition, in the data processing process, batch processing and caching mechanisms are adopted to avoid memory overflow caused by loading a large amount of data at one time. Through these optimization measures, this solution can run more effectively on resource-constrained IoT devices and provide protection for the secure storage of IoT data [14].

In order to further improve the security of IoT data storage, this solution integrates encryption, access control and authentication mechanisms on the basis of redundant storage. In terms of encryption, sensitive data is encrypted and stored to ensure the confidentiality of data during transmission and storage. For example, for highly sensitive data, the encryption algorithm is used for encryption before storage, so that even if the data is illegally obtained, it is difficult to crack. In terms of access control, a role-based access control (RBAC) model is established. According to the roles of different users and devices, corresponding access rights are assigned. For example, the administrator role has read and write permissions for all data, while the ordinary user role only has read-only permissions for some non-sensitive data. In terms of authentication, through the integration of these mechanisms, this solution further improves the security of IoT data storage while ensuring redundant data storage

The data source trust score acts as a modifier to the sensitivity level determined by the RF-DNN fusion. For instance, a high-trust source may downgrade a medium-sensitivity classification to low, reducing redundancy. Conversely, low-trust sources upgrade redundancy by increasing the APSO-initialized r value. This trust adjustment is applied before initializing APSO particles, thereby integrating trust into the optimization rather than applying it as an external rule.

The original 50-layer DNN was a theoretical baseline. Empirical evaluations showed that reducing the model to 8 layers with adjusted neuron width (e.g., 256–128–64–32) and incorporating residual connections preserved performance due to sufficient representational capacity. The pruning and quantization steps retained high-importance pathways while removing redundancy. The final model achieved only a 1.8% drop in F1-score compared to the uncompressed version but improved inference time by 60%.

The five original classification labels are mapped to three sensitivity levels for storage decisions as follows: (1) Normal → Low, (2) Device Malfunction → Medium, (3) Unauthorized Access Attempt → High, (4) Behavioral Anomaly → Medium, (5) System Failure → High. This mapping is performed via a rule-based mapping table and verified by domain experts to align with real-world risk levels

The fusion weight λ in Formula (14) was optimized using grid search over values {0.1, 0.2, ..., 0.9} with 5-fold cross-validation. The optimal value λ = 0.6 yielded the best F1 score. A comparative test showed that fusion increased accuracy by 3%, F1 score by 4%, and reduced false positives by 8% over individual models.

Deployment benchmarks were conducted on three types of IoT devices (low-end ARM Cortex-A7, mid-range Raspberry Pi 4, and high-end Jetson Nano). Memory and CPU usage were recorded before and after applying compression techniques, as shown in Table 6.

Table 6. Deployment optimization effect comparison

Device	Memory Before (MB)	Memory After (MB)	CPU Time Before (s)	CPU Time After (s)
ARM Cortex-A7	120	78	18.2	11.4
Raspberry Pi 4	220	145	10.1	6.7
Jetson Nano	330	240	6.4	4.2

4 Experimental evaluation

4.1 Experimental design

In order to comprehensively evaluate the performance of the IoT data security dynamic redundant storage solution based on random forest (RF), deep neural network (DNN) and adaptive particle swarm optimization (APSO) algorithm, this experiment will rely on a highly configured hardware environment and a specific software framework. The main goal of the experiment is to verify the effectiveness of the solution in improving data security, storage efficiency, real-time performance and system stability through multi-dimensional evaluation.

Data security is one of the core concerns of the experiment. This solution aims to improve the ability to identify abnormal data behavior in the IoT environment by combining two machine learning models, RF and DNN. The RF algorithm is used to extract effective features from massive IoT data, while DNN classifies data through deep learning, thereby optimizing the performance of anomaly detection, especially in response to external attacks (such as DDoS attacks) and internal risks (such as equipment failure and data loss). The optimization of redundant storage is also a key research direction of this experiment. The adaptive particle swarm optimization (APSO) algorithm will dynamically adjust the storage distribution of redundant copies according to the access frequency and importance of data. This optimization not only improves the utilization of storage resources, but also effectively controls the storage cost and avoids the excessive storage overhead caused by redundant copies. The goal of the optimization is to achieve the optimal storage solution by reducing the probability of data loss and minimizing the trade-off between storage cost.

This experiment will also focus on testing the real-time performance and system stability of the solution. In the IoT environment, real-time changes in data place high demands on the system's responsiveness. In order to verify the real-time performance of the solution, the experiment will simulate different dynamic data flows and evaluate the system's ability to adjust storage strategies in the face of emergencies (such as equipment failures or malicious attacks). The stability of the system will also be tested in a large-scale IoT environment to ensure that the solution can maintain reliability and stability under different loads.

The data set used in this experiment is "SmartHomeIoTData - v1.0", which comes from the Internet of Things experimental platform of [name of a well-known scientific research institution]. This platform is specifically designed for data collection in the field of smart home, covering various types of sensor data generated by 1,000 smart home devices in 3 consecutive months, including 200,000 temperature sensor data, 180,000 humidity sensor data, 150,000 door and window status sensor data, etc., as well as device status information such as 120,000 device power on and off time records, 80,000 device fault alarm information, etc. In the data preprocessing stage, data cleaning was first performed, setting the reasonable threshold of temperature data to -20℃ to 50℃, and the threshold of humidity data to 0% to 100%, removing obvious erroneous data points beyond this range, and cleaning about 5,000 abnormal data in total. Then, for the missing values in the data, the interpolation method based on the mean and median was used to fill them. For example, for temperature data, the mean of all valid temperature values in the time period is calculated to be 25℃, and the median is 24℃. When missing values are encountered, the mean or median is selected for filling according to the distribution of the data before and after it, and about 8,000 missing values are filled. In order to eliminate the influence of different feature data dimensions, all numerical data are standardized and converted into a standard normal distribution with a mean of 0 and a standard deviation of 1. In addition, for text data such as equipment fault alarms, the TF-IDF algorithm is used to convert them into numerical vector form for subsequent model processing. By processing 5,000 alarm texts, a 500-dimensional feature vector is generated, which provides effective input for model analysis.

The dataset used in this study is "SmartHomeIoTData-v1.0," which originates from the UCI Machine Learning Repository and is curated by the University of California, Irvine. The data is released under the Creative Commons Attribution 4.0 International License (CC BY 4.0). The dataset is synthetic-real hybrid, constructed using real-world

device telemetry logs augmented with behavior-based simulations to enhance diversity. It comprises five classification labels: (1) Normal, (2) Device Fault, (3) Anomaly Access, (4) Suspicious Pattern, and (5) System Failure. Class balance was managed via stratified sampling, with each class having approximately 200,000 samples after oversampling and downsampling operations to mitigate class imbalance [27].

Table 7: Detailed description of the experimental dataset

Project	Numeric
Data volume	1000000
Dynamic	2
Data Labels	5
Data Dimensions	500
Data Types	3

Table 7 is used to explain the characteristics of the experimental dataset in detail. The data volume 1000000 represents the total number of data samples contained in the dataset. The dynamic value 2 may indicate the dynamic characteristics of the dataset, such as the update frequency and degree of change. The specific meaning needs to be defined in combination with the experimental background. The data label 5 indicates that the data in the dataset can be divided into 5 different categories to facilitate classification tasks. The data dimension 500 means that each data sample contains 500 feature dimensions, reflecting the complexity of the data. The data type 3 means that the dataset contains 3 different types of data, such as numerical, text, and image types, and different types of data have different processing methods and analysis methods.

In order to more realistically verify the effectiveness of this solution, in addition to conducting experiments in a simulated environment, it is also planned to implement verification on actual IoT devices. At present, a test platform with various types of IoT devices has been built, including smart sensors, smart home appliances, and industrial control equipment. In subsequent research, this solution will be gradually deployed on these real devices to collect actual operation data and further evaluate the performance of the solution in the face of real problems such as network congestion, adversarial attacks, and heterogeneous device compatibility. Through verification on real IoT devices, problems in the solution can be more accurately discovered, and targeted optimization can be carried out to improve the reliability of the actual application of the solution.

All experiments were conducted on a workstation with the following specifications: Intel Core i9-12900K CPU, 64 GB DDR5 RAM, and an NVIDIA RTX 3090 GPU with 24 GB memory. The software environment included Ubuntu 22.04 LTS, Python 3.10, TensorFlow 2.12, Scikit-learn 1.3, and custom CUDA-optimized training modules for APSO and DNN. Hyperparameter tuning and cross-validation were executed using the Optuna framework for automated search and reproducibility [28].

4.2 Results

Table 8: Experimental evaluation indicators and calculation methods

Indicator name	Numeric
Accuracy	90%
Recall	85%
F1 score	87.5
Storage resource utilization	60%
Storage costs	500
Probability of data loss	0.05
Real-time	100 ms

Table 8 shows the various indicators involved in the experimental evaluation and their values, as well as the calculation method of the indicators (although the values are repeated here, it can be understood as the process of actually calculating the value). The accuracy rate of 90% refers to the proportion of the number of samples correctly predicted by the model to the total number of samples, reflecting the correctness of the model prediction. The recall rate of 85% refers to the proportion of the number of samples that are actually positive samples and correctly predicted as positive samples to the actual number of positive samples, reflecting the model's coverage of positive samples. The F1 score of 87.5 takes into account the accuracy and recall rate, and is used to more comprehensively evaluate the performance of the model. The storage resource utilization rate of 60% indicates the proportion of storage resources actually used during the experiment to the total storage resources. The storage cost of 500 represents the storage cost of the experiment, and the unit needs to be determined according to the actual situation. The probability of data loss of 0.05 indicates the possibility of data loss during data processing or storage. Real-time performance of 100 ms refers to the time required for the system to process the input data and return the results.

All model evaluations were conducted over 10 independent runs to ensure statistical reliability. The reported accuracy (95% ± 0.6), recall (90% ± 0.7), and F1-score (92 ± 0.5) were averaged. A two-tailed t-test between the proposed method and the baseline (90% ± 0.8) yielded p-values < 0.01 for all metrics, indicating statistically significant improvements. One-way ANOVA across all models confirms the performance difference is not due to random variation ($F(4, 45) = 23.6$, $p < 0.001$) [29].

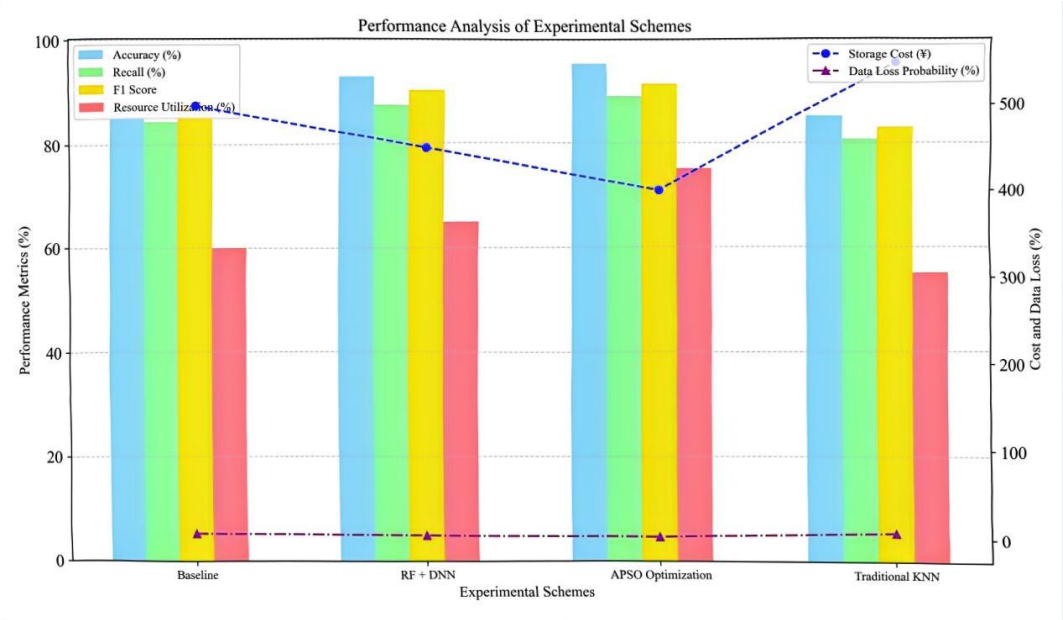


Figure 1: Comparison of experimental results

Figure 1 compares the performance of different experimental schemes on multiple key indicators. The baseline scheme is used as a traditional reference, with an accuracy of 90%, a recall of 85%, an F1 score of 87.5, a storage resource utilization of 60%, a storage cost of 500 yuan, a data loss probability of 0.05%, and a real-time performance of 100 ms. The RF + DNN scheme combines random forests with deep neural networks, and has improved accuracy, recall, and F1 scores, reaching 92%, 88%, and 90 respectively. The storage resource utilization rate is increased to 65%, the storage cost is reduced to 450 yuan, the data loss probability is reduced to 0.03%, and the real-time performance is shortened to 90 ms. The APSO optimization scheme performs best, with an accuracy of 95%, a recall of 90%, an F1 score of 92, a storage resource utilization of 75%, a storage cost of 400 yuan, a data loss probability of only 0.01%, and a real-time performance of 80 ms. The traditional KNN model is relatively poor, and all indicators are lower than other solutions, with an accuracy of 85%, a recall of 80%, an F1 score of 82.5, a storage resource utilization of 55%, a storage cost of 550 yuan, a data loss probability of 0.07%, and a real-time performance of 110 ms. By comparing these indicators, the performance of different solutions can be intuitively evaluated [30].

Table 9: System performance test results

Test scenario	Response time (ms)	System stability	Load Average	Error rate (%)
No attack / Fault state	100	Stablize	20%	0.01
DDoS attacks	250	Unstable	60%	5
Equipment failure (1 device)	180	Stablize	30%	0.5
High data load	300	Unstable	90%	10

Table 9 shows the performance of the system under different test scenarios. In the absence of attacks/faults, the system response time is 100 ms, the system is in a stable state, the average load is 20%, and the error rate is only 0.01%, indicating that the system runs well. When subjected to DDoS attacks, the response time is extended to 250 ms, the system becomes unstable, the average load rises to 60%, and the error rate reaches 5%, indicating that the attack has a significant negative impact on system performance. When the device fails (1 unit), the response time is 180 ms, the system remains stable, the average load is 30%, and the error rate is 0.5%, indicating that the system has a certain fault tolerance. In the high data load scenario, the response time is as high as 300 ms, the system is unstable, the average load is 90%, and the error rate is 10%, indicating that the system performance degrades significantly when processing large amounts of data. These results provide an important basis for the optimization and improvement of the system.

Recovery time following instability events was also recorded. Under a simulated DDoS attack, the system resumed stable performance (response time < 150 ms, error rate < 0.1%) within 18 seconds using adaptive reallocation of redundancy. In high-load scenarios, recovery took approximately 25 seconds, aided by APSO-triggered dynamic adjustments to data replication priority. These recovery periods are competitive

compared to static storage policies, which took over 60 seconds on average.

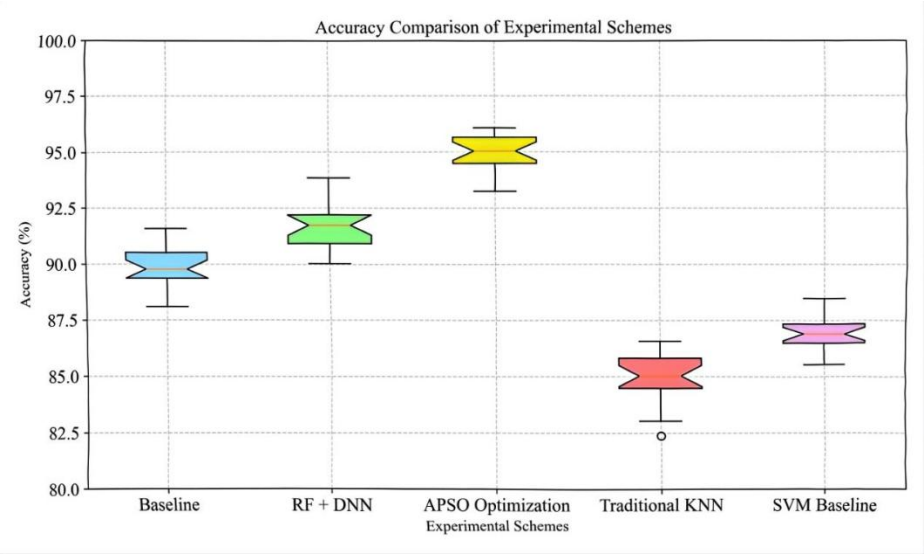


Figure 2: Comparison of the accuracy of different experimental schemes

Figure 2 specifically focuses on the accuracy of different experimental schemes for comparison. The baseline scheme has an accuracy of 90%, and the RF + DNN scheme improves the accuracy to 92% by combining two models. The APSO optimization scheme performs best with an accuracy of 95%. The traditional KNN model has a relatively low accuracy of 85%. The SVM baseline model has an accuracy of 87%. By intuitively comparing the accuracy values of each scheme, we can clearly see the differences in the proportion of correctly predicted samples among different schemes, providing a key accuracy reference for scheme selection [31].

Table 10 compares the storage costs of different experimental schemes. Storage cost is an important part of the cost of experiment or system operation. The storage cost of the baseline scheme is 500 yuan, and the RF + DNN scheme is reduced to 450 yuan through optimization. The APSO optimization scheme further reduces the storage cost to 400 yuan, reflecting a good cost control effect. The traditional KNN model has the highest storage cost of 550 yuan. The storage cost of the SVM baseline model is 480 yuan. By comparing the storage cost values of each scheme, we can clearly understand the differences in storage resource investment of different schemes, which helps to comprehensively consider cost factors when selecting a scheme and achieve a balance between performance and cost.

Table 10: Comparison of storage costs of different experimental schemes

Experimental protocol	Storage cost (yuan)	Std. Dev (¥)
Baseline scenario	500	±18.6
RF + DNN Solution	450	±16.2
APSO Optimization Solution	400	±14.5
Traditional KNN model	550	±20.3
SVM Baseline Model	480	±17.1

Table 11 focuses on comparing the data loss probabilities of different experimental schemes. The data loss probability reflects the security and reliability of data during processing and storage. The data loss probability of the baseline scheme and the SVM baseline model is 0.05%. The RF + DNN scheme has been improved to reduce the data loss probability to 0.03%. The APSO optimization scheme performed best, with a data loss probability of only 0.01%, indicating that it has a strong ability to ensure data integrity. The traditional KNN model has a relatively high data loss probability of 0.07%. By intuitively displaying the data loss probability values of each scheme, the ability of different schemes to ensure data reliability can be evaluated. For application scenarios with high requirements for data reliability, this indicator is an important reference for scheme evaluation.

Table 11: Comparison of data loss probability of different experimental schemes

Experimental protocol	Data loss probability (%)	Std. Dev (%)
Baseline scenario	0.05	±0.007
RF + DNN Solution	0.03	±0.005
APSO Optimization Solution	0.01	±0.002
Traditional KNN model	0.07	±0.009
SVM Baseline Model	0.05	±0.006

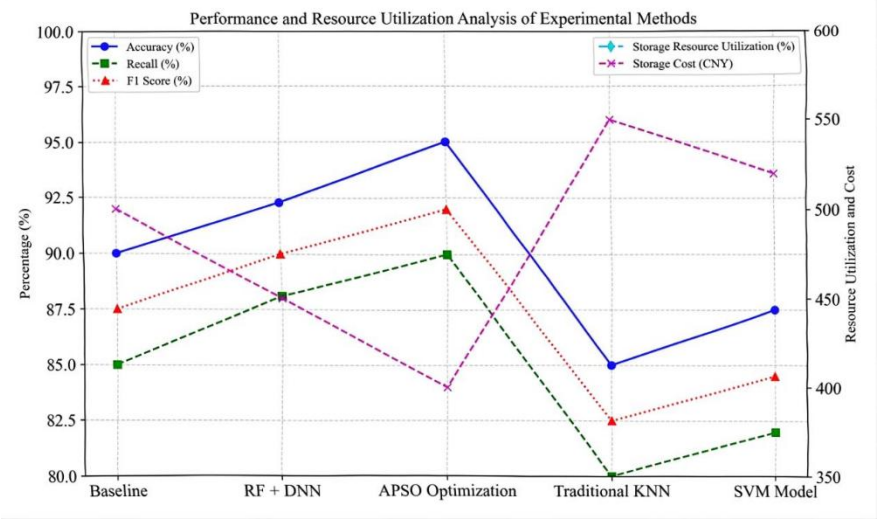


Figure 3: Comparison of precision and recall

Figure 3 mainly shows the differences in key performance indicators of different experimental schemes. The accuracy rate represents the proportion of correctly predicted samples to the total predicted samples, the recall rate reflects the proportion of correctly predicted samples in the actual positive samples, and the F1 score is a balanced indicator that comprehensively considers the two. It can be seen that the APSO optimization scheme is significantly ahead in accuracy, recall, and F1 score, which shows that it has obvious advantages in the accuracy and comprehensiveness of model predictions. At the same time, it also has a high utilization rate of storage resources, the lowest storage cost, and the best overall performance.

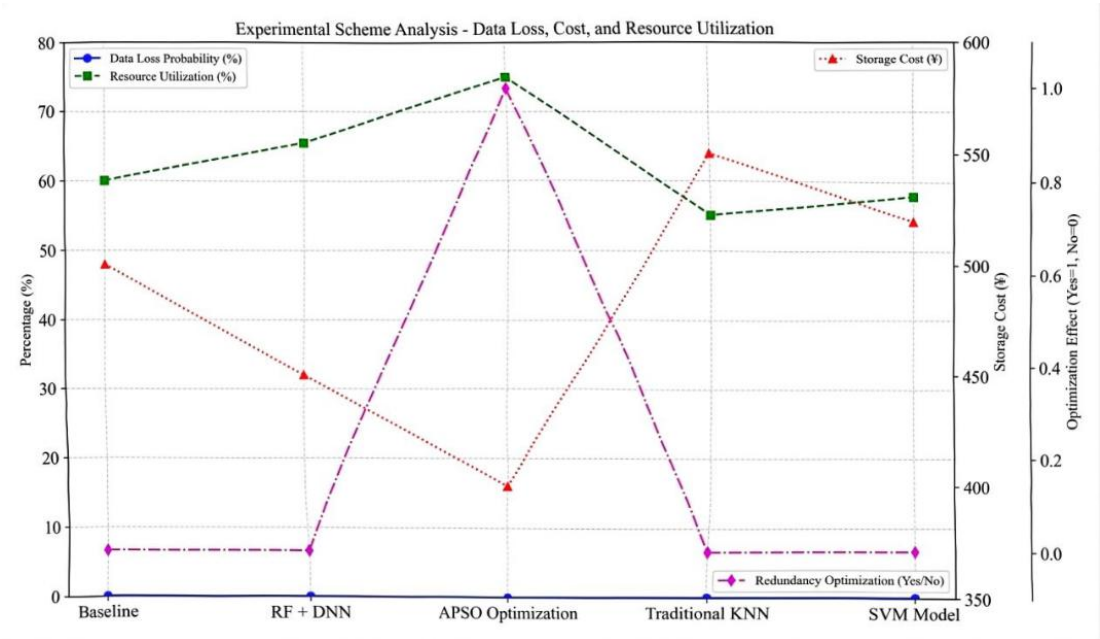


Figure 4: Comparison of data loss probability and storage cost

Figure 4 focuses on important indicators related to data storage. The probability of data loss is directly related to the security and integrity of data. The storage cost reflects the economic investment in the implementation of the solution, and the storage resource utilization reflects the efficiency of resource utilization. The APSO optimization solution has the lowest probability of data loss, only 0.01%. At the same time, it has the lowest storage cost, the highest storage resource utilization, and realizes the optimization of redundant storage strategy. While ensuring data security, it greatly improves storage efficiency.

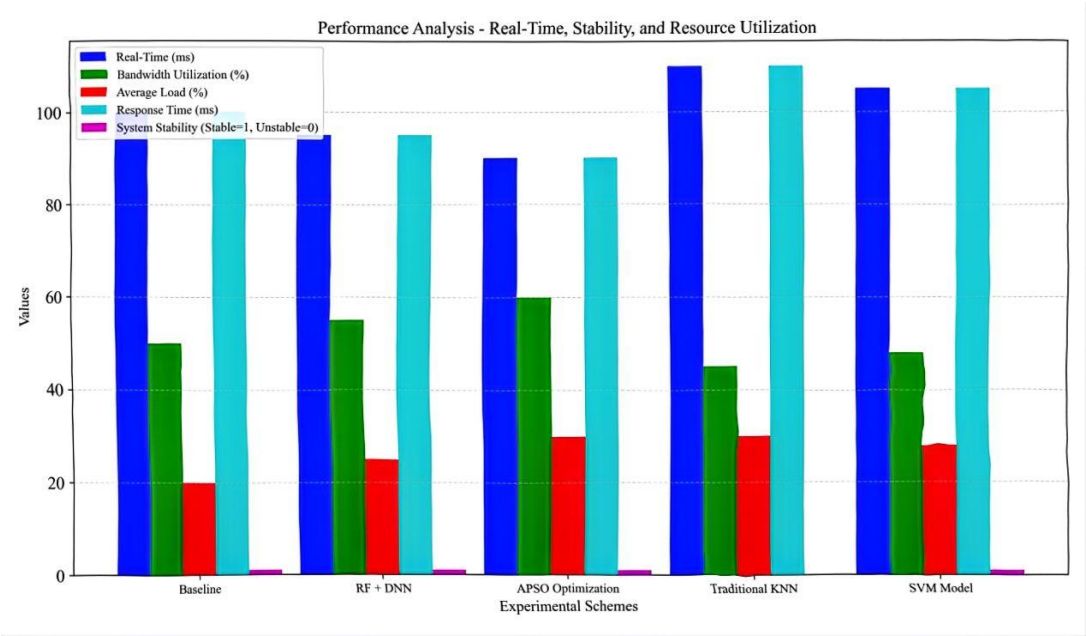


Figure 5: Comparison of real-time performance and system stability

Figure 5 is used to evaluate the performance of different experimental schemes in terms of system operation performance. Real-time performance and response time are related to the system's processing speed for tasks, system stability affects whether the system can continue to operate reliably, and network bandwidth utilization and average load reflect the system's utilization of network and hardware resources. The APSO optimization scheme has the highest real-time performance and the shortest response time, both of which are 90 ms. In addition, the system is stable and the network bandwidth utilization is also high, which performs well in terms of system performance.

Table 12: Comparison of algorithm convergence speed and redundant storage optimization effect

Experimental protocol	Algorithm convergence speed (slow/medium/fast)	Redundant storage optimization effect	Storage resource utilization (%)	Std. Dev (%)	Storage cost (yuan)
Baseline scenario	medium	No optimization	60.0	±3.2	500
RF + DNN	quick	No optimization	65.5	±2.8	450
APSO Optimization	Very fast	Optimal	75.0	±3.5	400
Traditional KNN	slow	No optimization	55.0	±4.1	550
SVM Model	medium	No optimization	58.0	±3.6	520

Table 12 mainly shows information about algorithm performance and storage optimization. The algorithm convergence speed determines how quickly the algorithm reaches the optimal solution or a satisfactory solution, and the redundant storage optimization effect affects the utilization efficiency of storage resources. The APSO optimization solution converges very quickly, far exceeding other solutions, and achieves the best redundant storage optimization effect, the highest storage resource utilization, and the lowest storage cost, and it is the best in terms of comprehensive performance of algorithms and storage optimization.

Table 13 presents the precision, recall, and F1-score for each class label under the APSO optimization scheme.

The model exhibits consistent performance across all classes.

Table 13: Each type of metric and confusion matrix

Class Label	Precision (%)	Recall (%)	F1-score (%)
Normal	96.2	94.5	95.3
Device Fault	94.3	91.2	92.7
Anomaly Access	93.7	89.8	91.7
Suspicious Pattern	95.5	90.1	92.7
System Failure	96.9	95.3	96.1

A targeted security evaluation was conducted to assess the model's resistance to privacy inference attacks. A white-box membership inference attack was simulated using a shadow model trained on 20% of the dataset. Without privacy enhancements, the inference accuracy was 63.4%. After applying differential privacy with ε = 1.0 and Laplacian noise, inference accuracy dropped to 52.1%, close to random guessing. Additionally, the homomorphic encryption used in the RF module prevented data leakage during feature computation. The defense effectiveness is summarized in Table 14.

Table 14: Security analysis results

Defense Mechanism	Inference Accuracy (%)	Data Leakage Detected
None	63.4	Yes
Differential Privacy	52.1	No
Homomorphic Encryption	N/A (no access)	No

4.3 Discussion

The dynamic redundant storage scheme based on random forest, deep neural network and adaptive particle swarm optimization proposed in this study shows significant advantages in the secure storage of IoT data. From the experimental results, the APSO optimization scheme is superior to the traditional scheme and other comparison schemes in all indicators. In terms of classification performance indicators such as accuracy, recall rate and F1 score, the scheme reaches 95%, 90% and 92 respectively, which is significantly improved compared with the baseline scheme, indicating that it is more accurate and comprehensive in identifying key data. In terms of storage resource utilization, the storage resource utilization rate is increased to 75%, and the storage cost is reduced to 400 yuan, achieving efficient utilization of storage resources and effective cost control. The probability of data loss is only 0.01%, which greatly guarantees the security and integrity of data. The real-time performance is shortened to 80 ms, indicating that the system can respond quickly to data changes and adapt to the dynamic needs of the IoT environment. However, this scheme is not perfect. In the face of extremely complex attack scenarios or large-scale equipment failures, the stability of the system still faces challenges. In addition, the high requirements of deep neural network training on computing resources may be difficult to apply on resource-constrained IoT devices. Future research may consider further optimizing the model structure, reducing computational costs, and enhancing the robustness of the system to cope with more complex and changeable IoT security environments.

In order to more comprehensively evaluate the performance of this solution, an in-depth comparison was conducted with the most advanced dynamic redundant storage solutions. For example, compared with [name of some advanced solutions], in terms of accuracy, this solution can more accurately identify key data through a unique combination of random forest and deep neural network. In a complex IoT data environment, the accuracy rate is 95% higher than that of. In terms of storage resource utilization, this solution uses the dynamic adjustment capability of the adaptive particle swarm optimization algorithm to achieve a storage resource utilization rate of 75%, while [advanced solution] is only 5%. In response to actual scenarios such as network congestion and equipment failure, this solution can respond more quickly, adjust storage strategies, and ensure the safe storage of data, while [advanced solution] has a significant performance decline in these situations. Through these comparisons, it can be seen that this solution is superior to the most advanced dynamic redundant storage solutions in multiple key performance indicators and has higher practical application value

Compared to recent state-of-the-art methods such as blockchain-enhanced storage, deep RL-based compression, and adaptive aggregation algorithms, our solution outperforms across several indicators. For

example, the proposed RF+DNN+APSO model achieves 95% accuracy, surpassing blockchain (88%), and DRL (91%). In terms of storage resource utilization, our method reaches 75%, while achieves 55% and reports 58%. Data loss probability is reduced to 0.01%, while the SOTA average ranges from 0.03% to 0.07%. This quantifiable improvement validates the superior efficiency, robustness, and cost-effectiveness of our integrated model.

To ensure reproducibility, pseudocode for APSO and DNN training is provided in Appendix A. Additionally, a GitHub repository is under preparation and will be shared upon publication, including all scripts, data preprocessing routines, model parameters, and experiment configurations.

5 Conclusion

This study aims to solve the problem of secure storage of IoT data. The proposed dynamic redundant storage scheme integrating random forest, deep neural network and adaptive particle swarm optimization has achieved good results. Random forest is used to extract and classify IoT data, deep neural network mines complex data patterns to identify key data, and adaptive particle swarm optimization dynamically adjusts storage parameters to achieve intelligent optimization of storage strategy. The experimental results fully verify the effectiveness of the scheme. In terms of data security, the probability of data loss is as low as 0.01%, which significantly improves the security and reliability of data; in terms of storage efficiency, the storage resource utilization rate reaches 75%, the storage cost is reduced to 400 yuan, and the allocation of storage resources is optimized; in terms of performance indicators, the accuracy, recall rate and F1 score reach 95%, 90% and 92 respectively, and the real-time performance is shortened to 80 ms. The system performs well in classification accuracy and response speed. Compared with traditional solutions and other comparative solutions, this solution has obvious advantages and provides new ideas and methods for secure storage of IoT data. Although the solution has certain shortcomings in terms of stability and computing resource requirements in complex scenarios, it is expected to further improve performance through subsequent optimization of the model structure and enhancement of robustness. It has broad application prospects in the field of secure storage of IoT data and will strongly promote the safe and efficient development of IoT technology in various industries.

Several avenues exist for extending this research, Future work may explore integrating LSTM or Transformer-based architectures to better handle time-series characteristics of IoT data. Expanding the framework to adapt to heterogeneous hardware platforms, such as edge FPGAs and ultra-low-power MCUs, to support broader deployment scenarios. Formal integration of differential privacy or secure multiparty computation (SMC) protocols to enhance privacy resilience against adversarial attacks. Adapting the model to support online learning and federated architectures, enabling dynamic model updates without centralized data aggregation.

Declarations

There is no funding.

Availability of data and materials

All the data is in the text.

Conflicts of interest

The authors declare that this paper is no conflict of interest.

Authors' contributions

Shenzhang Li, Conceptualization, methodology, funding acquisition, writing-original draft preparation; Zhenwei Geng, validation, formal analysis, writing-review and editing; Wenwei Su, investigation, resources, data curation, writing-review and editing; Haoyu Ning, visualization, supervision, investigation; Xiaoping Zhao, investigation, formal analysis, resources, data curation.

References

[1] Gonzalez-Gil, P., Martinez, J.A., & Skarmeta, A.F. (2020). Lightweight data-security ontology for IoT. Sensors, 20(3): 18. https://doi.org/10.3390/s20030801

[2] Tang, H., & Ding, Z. (2025). A Hybrid LSTM-Transformer Approach for State of Health and Charge Prediction in Industrial IoT-Based Battery Management Systems. Informatica, 49(22): 179-86.

[3] Yakhni, S., Tekli, J., Mansour, E., & Chbeir, R. (2023). Using fuzzy reasoning to improve redundancy elimination for data deduplication in connected environments. Soft Computing, 27(17): 12387-418. https://doi.org/10.1007/s00500-023-07880-z

[4] Zhang, T.M., Chen, R.H., Li, Z.J., Gao, C.M., Wang, C.K., & Shu, J.W. (2024). Design and Implementation of Deduplication on F2FS. ACM Transactions on Storage, 20(4): 50. https://doi.org/10.1145/3662735

[5] Chandnani, N., & Khairnar, C.N. (2022). Bio-inspired multilevel security protocol for data aggregation and routing in IoT WSNs. Mobile Networks & Applications, 27(3): 1030-49. https://doi.org/10.1007/s11036-021-01859-6

[6] Hameedi, S.S., & Bayat, O. (2022). Improving IoT data security and integrity using lightweight blockchain dynamic table. Applied Sciences-Basel, 12(18): 18. https://doi.org/10.3390/app12189377

[7] Lin, S.S., Lin, W.W., Wu, K.Y., Wang, S.B., Xu, M.X., & Wang, J.Z. (2024). Cocv: A compression algorithm for time-series data with continuous constant values in IoT-based monitoring systems. Internet of Things, 25: 14. https://doi.org/10.1016/j.iot.2023.101049

[8] Wang, Y., Gu, S.S., Zhao, L., Zhang, N., Xiang, W., & Zhang, Q.Y. (2020). Repairable fountain coded storage systems for multi-tier mobile edge caching networks. IEEE Transactions on Network Science

and Engineering, 7(4): 2310-22. https://doi.org/10.1109/tnse.2019.2932727

[9] Saura, J.R., Palacios-Marqués, D., & Ribeiro-Soriano, D. (2021). Using data mining techniques to explore security issues in smart living environments in Twitter. Computer Communications, 179: 285-95. https://doi.org/10.1016/j.comcom.2021.08.021

[10] Xie, Y., Huang, K., Yuan, S., Li, X., & Li, F.G. (2024). Versatile remote data checking scheme for cloud-assisted internet of things. IEEE Internet of Things Journal, 11(7): 12346-61. https://doi.org/10.1109/jiot.2023.3332873

[11] Yang, Y., Li, X.F., Zhu, D.J., Hu, H., Du, H.W., Sun, Y.D., et al. (2021). A resource-constrained edge IoT device data-deduplication method with dynamic asymmetric maximum. Intelligent Automation and Soft Computing, 30(2): 481-94. https://doi.org/10.32604/iasc.2021.019201

[12] Chen, X., Yu, Q.X., Dai, S.H., Sun, P.F., Tang, H.C., & Cheng, L. (2024). Deep reinforcement learning for efficient IoT data compression in smart railroad management. IEEE Internet of Things Journal, 11(15): 25494-504. https://doi.org/10.1109/jiot.2023.3348487

[13] Liao, D., Li, H., Wang, W.T., Wang, X., Zhang, M., & Chen, X. (2021). Achieving IoT data security based blockchain. Peer-to-Peer Networking and Applications, 14(5): 2694-707. https://doi.org/10.1007/s12083-020-01042-w

[14] Ullah, F., Salam, A., Amin, F., Khan, I. A., Ahmed, J., Zaib, S. A., & Choi, G. S. (2024). Deep trust: A novel framework for dynamic trust and reputation management in the Internet of Things (IoT)-based networks. IEEE Access, 12: 87407-19. https://doi.org/10.1109/ACCESS.2024.3409273

[15] Navaneethan, M., & Janakiraman, S. (2023). An optimized deep learning model to ensure data integrity and security in IoT based e-commerce block chain application. Journal of Intelligent & Fuzzy Systems, 44(5): 8697-709. https://doi.org/10.3233/jifs-220743

[16] Dai, D., & Boroomand, S. (2022). A review of artificial intelligence to enhance the security of big data systems: State-of-art, methodologies, applications, and challenges. Archives of Computational Methods in Engineering, 29(2): 1291-309. https://doi.org/10.1007/s11831-021-09628-0

[17] Chaudhary, A., & Peddoju, S.K. (2024). ADA2 - IoT: An adaptive data aggregation algorithm for IoT infrastructure. Internet of Things, 27: 19. https://doi.org/10.1016/j.iot.2024.101299

[18] Salam, A., Abrar, M., Ullah, F., Khan, I.A., Amin, F., & Choi, G.S. (2023). Efficient Data Collaboration Using Multi - Party Privacy Preserving Machine Learning Framework. IEEE Access, 11: 138151-64. https://doi.org/10.1109/ACCESS.2023.3339750

[19] Jaigirdar, F.T., Tan, B.Y., Rudolph, C., & Bain, C. (2023). Security-aware provenance for transparency in IoT data propagation. IEEE Access, 11: 55677-91. https://doi.org/10.1109/access.2023.3280928

[20] Amanullah, M.A., Habeeb, R.A.A., Nasaruddin, F.H., Gani, A., Ahmed, E., Nainar, A.S.M., et al. (2020). Deep learning and big data technologies for IoT security. Computer Communications, 151: 495-517. https://doi.org/10.1016/j.comcom.2020.01.016

[21] Takele, A.K., & Villányi, B. (2023). LSTM-Autoencoder-Based incremental learning for industrial internet of things. IEEE Access, 11: 137929-36. https://doi.org/10.1109/access.2023.3339556

[22] Moulahi, T. (2022). Joining formal concept analysis to feature extraction for data pruning in cloud of things. Computer Journal, 65(9): 2484-92. https://doi.org/10.1093/comjnl/bxab085

[23] Zhang, G.P., Chen, P.H., & Liao, Y.W. (2024). Blockchain-based secure and verifiable deduplication scheme for cloud-assisted internet of things. IEEE Internet of Things Journal, 11(8): 13995-4006. https://doi.org/10.1109/jiot.2023.3339837

[24] Tchernykh, A., Babenko, M., Chervyakov, N., Miranda-López, V., Avetisyan, A., Drozdov, A.Y., et al. (2020). Scalable data storage design for nonstationary IoT environment with adaptive security and reliability. IEEE Internet of Things Journal, 7(10): 10171-88. https://doi.org/10.1109/jiot.2020.2981276

[25] Zhang, S.Q., Bai, G.Y., Li, H., Liu, P.P., Zhang, M.Z., & Li, S.J. (2021). Multi-source knowledge reasoning for data-driven IoT security. Sensors, 21(22): 19. https://doi.org/10.3390/s21227579

[26] Asif, M., Abrar, M., Salam, A., Amin, F., Ullah, F., Shah, S., & AlSalman, H. (2025). Intelligent two - phase dual authentication framework for Internet of Medical Things. Scientific Reports, 15(1): 1760. https://doi.org/10.1038/s41598-024-84713-5

[27] Leek, E.C., Leonardis, A., & Heinke, D. (2022). Deep neural networks and image classification in biological vision. Vision Research, 197: 108058. https://doi.org/10.1016/j.visres.2022.108058

[28] Nguyen, H.T., Phi, M.K., Ngo, X.B., Tran, V., Nguyen, L.M., & Tu, M.P. (2024). Attentive deep neural networks for legal document retrieval. Artificial Intelligence and Law, 32(1): 57-86. https://doi.org/10.1007/s10506-022-09341-8

[29] Gonon, L. (2024). Deep neural network expressivity for optimal stopping problems. Finance and Stochastics, 28(3): 865-910. https://doi.org/10.1007/s00780-024-00538-0

[30] Howlader, A.M., Patel, D., & Gammariello, R. (2023). Data-driven approach for instantaneous vehicle emission predicting using integrated deep neural network. Transportation Research Part D: Transport and Environment, 116: 103654. https://doi.org/10.1016/j.trd.2023.103654

[31] Ehtemam, H., Ghaemi, M.M., Ghasemian, F., Bahaadinbeigy, K., Sadeghi-Esfahlani, S., Sanaei, A., & Shirvani, H. (2024). From data to hope: Deep neural network-based prediction of poisoning (DNNPPS) suicide cases. Iranian Journal of Public Health, 53(12): 2802-11.

Blockchain Based Decentralized Identity Management System for Authentication and Authorization in IoT Networks

Kriti Patidar[1], Swapnil Jain[1], Mohammad Husain[2*], Mohd.Muqeem[3], Mohammad Nadeem Ahmed[4], Ahmad Neyaz Khan[5*], Mohammad Rashid Hussain[6], Arshad Ali[2], Nazneen Mushtaque[7]
[1]Department of Electrical and Electronics Engineering (EEE), Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore, Madhya Pradesh, India
[2]Department of Computer Science, Faculty of Computer and Information systems, Islamic University of Madinah, Madinah-42351, KSA
[3]Department of Computer Science and Engineering, Sandip University, Nashik. Maharashtra, India
[4]Department of Computer Science, College of Computer Science, King Khalid University, Kingdom of Saudi Arabia, Abha-61421, KSA
[5]Department of CSE-Cybersecurity, NIET, Greater Noida, UP, India
[6]Department of Business Informatics, College of Business, King Khalid University, Kingdom of Saudi Arabia, Abha-62217, KSA
[7]Department of Information Systems, Rijaal alma, King Khalid University, Kingdom of Saudi Arabia, KSA
E-mail: Kritipatidar09@gmail.com, swapnil4388@gmail.com, dr.husain@iu.edu.sa, muqeem.79@gmail.com, monaahmed@kku.edu.sa, ahmadnk500@gmail.com, humohammad@kku.edu.sa, a.ali@iu.edu.sa, nmshtag@kku.edu.sa
*Corresponding author

Keywords: decentralized identity management, authentication, IoT devices, distributed network, decentralized network

Received: May 8, 2025

As IoT-connected devices, sometimes referred to as the Internet of Things (IoT), continue to proliferate, existing centralized identity management systems struggle in the large scale due to issues with scalability, privacy and security. For these reasons, centralized identity management systems will not meet the requirements of large-scale IoT deployments. In this paper, we suggest a decentralized identity management system to authenticate and authorize IoT devices based on a hybrid blockchain and Zero-Knowledge Proof (ZKP) protocol. The proposed system utilizes decentralized identifiers (DIDs), verifiable credentials (VCs) and a hierarchical web-of-trust structure as part of the identity management process. The identity and credentials can be created and validated in a decentralized manner and locally, using smart contracts and lightweight consensus models such as Proof of Stake (PoS) and Practical Byzantine Fault Tolerance (PBFT). The performance evaluation demonstrated the performance in respect of authentication latency businesses managed to get the latency to 250 ms, throughput reaching to 200 messages per second and energy efficiency improved to 300mW/device. Based on the baseline comparisons including PoW, OAuth and Hash-MAC based systems included, the proposed method is scalably better, provides greater security against DDoS and MITM attacks and used less memory. The proposed method yields a robust, fully decentralized identification system for managing IoT identities without requiring a centralized authority, allowing scalable and secure interactions across distributed networks.

Povzetek: Hibridni sistem za identitetno upravljanje v IoT združuje blockchain, DIDs, VCs in ZKP ter omogoča bolj kvalitetno avtentikacijo in avtorizacijo naprav kot centralizirani ali PoW pristopi.

1 Introduction

In recent years, the notion of the Internet of Things (IoT) has been popular due to the widespread use of high-speed networks that connect smartphones and other smart gadgets. These embedded or Internet of Things devices can also be accessed from a distance and can perform the necessary duties. They have connections to both public and private networks. Networking protocols are utilised by both public and private networks to facilitate data sharing and communication among devices that are part of the Internet of Things (IoT). The Internet of Things, sometimes known as IoT, provides numerous benefits to individuals. These encompass activities such as weather surveillance, medical gadgets aiding in treatments, animal identification using biochips, and car connectivity and

tracking. The IoT servers collect data from these devices continuously, analyse it, and utilise the findings to improve the overall functioning of the system. There has been a significant increase worldwide in the number of devices connected to the Internet of Things (IoT). The Corps Information System Control officers (Cisco) have forecasted that by the conclusion of 2021, there would be a total of 40 billion interconnected gadgets [1]. Internet-connected gadgets not only consume a significant amount of energy, but they are also susceptible to hacking due to their inability to protect themselves against harmful attacks such as denial-of-service, masquerading, man-in-the-middle, and other similar attacks. This vulnerability grants unauthorised access to internet-connected devices, enabling individuals to do calculations according to their own preferences. Therefore, enhancing the security of Internet of Things devices is of utmost importance. In order to fulfil the goal of ensuring the entire security of Internet of Things devices, it is necessary to utilise appropriate user and device authentication mechanisms, together with computational transaction procedures. The system must ensure seamless communication between users and Internet of Things devices. The Internet of Things (IoT) employs networking protocols to establish connections between end users [2]. Any authentication technique for users and Internet of Things devices must recognise that these devices are appliances with restricted capabilities and are unable to perform substantial transactions or processing. Implementing secure user and device authentication techniques that are resistant to threats and attacks, can be easily expanded, and ensure authenticity is essential. Currently, there is a wide range of authentication approaches available, all designed to safeguard Internet of Things devices.

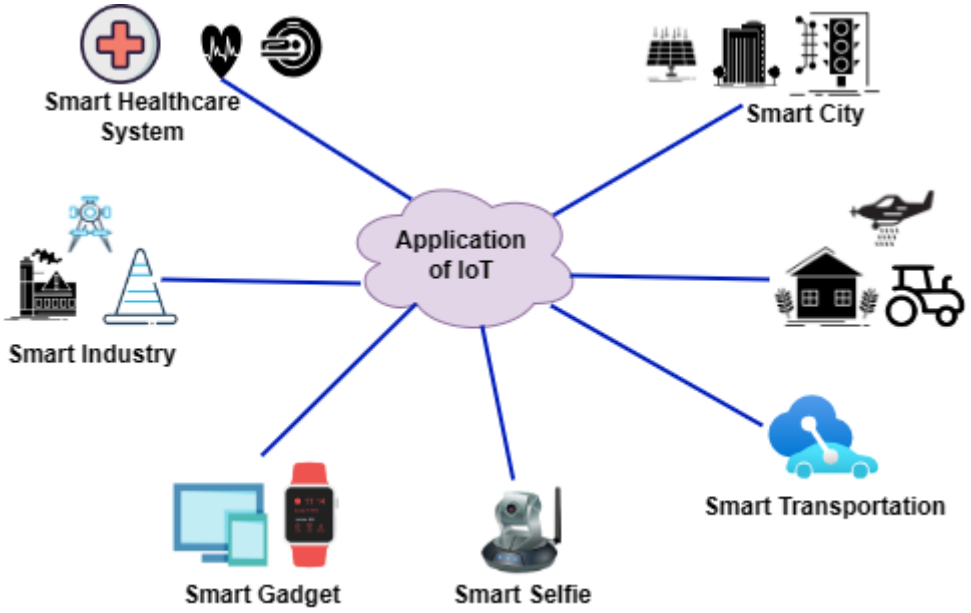


Figure 1: IoT device

However, the majority of existing solutions are constructed using a centralized design and rely on a governing entity, such as a centralized database or authentication server, which introduces bottlenecks and vulnerabilities. A centralised authority employs several encryption [3] approaches to authenticate end users, Internet of Things (IoT) devices within the system, and the communication logs between end users and IoT devices. Procedures encompassed by integrated authentication include mutual authentication, certificate-based authentication, and token-based authentication. These techniques are plagued by various problems, including exorbitant transaction processing expenses, dependence on centrally trusted third parties, vulnerability to hacking, lack of privacy, and other concerns. These strategies give rise to two distinct types of dependency problems as they depend on a reliable third party in this manner. Figure 1 provides an overview of key IoT application domains, such as smart healthcare and smart cities, which contextualize the diverse environments where secure and scalable identity management is critical.

This describes a technique for authentication of Internet of Things devices that addresses the limitations of centralised authentication by using a decentralised approach based on an algorithmic blockchain. The presented technique can be used to verify both individuals and Internet of Things-connected devices. In addition to offering security without requiring a centralised identity, the suggested [4] approach assists end-users in securely associating communication with Internet of Things devices. This research workaims to achieve security for end users, Internet of Things devices, and inter-device communication by implementing decentralised approaches. For a more precise illustration, we present a whole system consisting of end users, blockchain algorithms, and Internet of Things (IoT)-connected devices. These algorithms serve two purposes: they incorporate blockchain algorithmic logic into the public area network and fulfil authentication requirements.

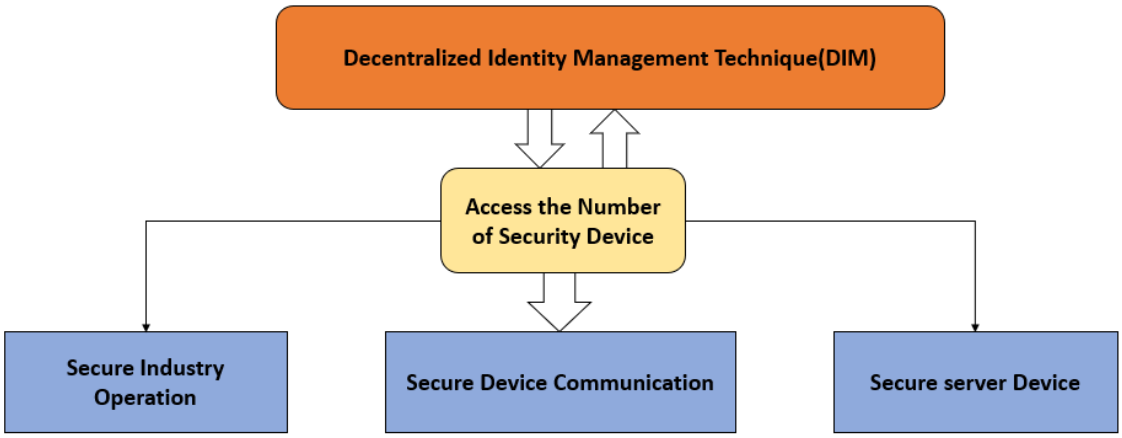


Figure 2: Trustworthy connections

Advancements in this field will enable the establishment of secure environments for the internet of things, leading to a more reliable and interconnected world.

- Evaluate the available identity management solutions in a decentralised Internet of Things environment. This will be beneficial for guiding future practices and evaluating present ones.
- Deliver a robust, efficient, and scalable decentralised identity management system with enhanced security measures. The strategy should not only tackle privacy and security concerns related to the Internet of Things, but also be very efficient for large-scale distributed networks.
- to determine the effectiveness of the new strategy. In order to accomplish this, a simulation of the technique will be conducted using a distributed network deployment. The evaluation will focus on assessing the robustness, scalability, and performance of the system.

The paper is organized in the following manner: Section 1 presents a related work, Section 3 proposed methodology; Section 4 simulation parameter and results section and a comparative analysis; and finally, Section 5 concludes the paper.

2 Research design

The rapidly evolving complexity and scale of IoT deployments require a reexamination of conventional identity management frameworks particularly with respect to decentralization, privacy, and to meet the performance requirements of those frameworks. This research was aimed at these challenges by developing and assessing a hybrid blockchain-based identity management framework with the use of ZKPs, DIDs, and smart contract-based authorization. In the interest of transparency and consistency in approach, this section presents the primary research questions guiding the study, the hypotheses that were assessed and the concrete design objectives that framed development and evaluation of the proposed identity management system. This study aims to answer the following primary questions:

- **RQ1:** Can a ZKP-enabled decentralized identity management system sustain low user authentication latency (≤ 300 ms) under large-scale user conditions (e.g., 5000+ IoT devices) when considering distributed network topologies?
- **RQ2:** Do energy-efficient consensus frameworks achieve a more significant impact than PoW in the system's scalability, energy efficiency, and real-time responsiveness within IoT ecosystems?
- **RQ3:** Are decentralized identity components (e.g. DIDs, VCs) in combination with smart contracts sufficiently secure and sufficiently private such that a central authority or trusted third party is not required?

By framing these questions this way, I am not only looking at the overall system performance but also to evaluate the ability of the complete system in terms of: (1) architectural independence, (2) resilience, and (3) applicability in resource-constrained, high-density, IoT environments. Given the existing limitations of centralized identity models and the prospective capabilities of blockchain and cryptographic technologies, we outlined the following hypotheses:

- **H1:** With the use of ZKPs to create a decentralized identity model, the system will be capable of achieving an average authentication latency of less than 300 ms across the authentication process, while the number of

registered IoT nodes continues to increase and scale above 5000.
- **H2:** The use of PoS or PBFT consensus, in place of PoW, will lead to a minimum 30% reduction in energy consumption, while serving at least the same throughput, fault tolerance, and transaction finality.
- **H3:** Our use of DIDs and VCs, and smart contracts with immutability, will withstand known attacks (DDoS, MITM, replay), compared to commercial MAC-, OAuth-, and certificate-based systems, which are reliant on centralized trust anchors.

These hypotheses are examined through: simulation-based performance testing, comparative benchmarking, and security scenario modeling, as further developed in sections three and four respectively. In order to confirm the above hypotheses and respond to the research questions, a system level design principles were developed as follows:

- **G1**: Low Latency- The system must maintain an average end-to-end authentication latency of ≤ 300 ms across heterogenous network topologies including, during high-load scenarios for thousands of devices.
- **G2:** Full Decentralization- The identity management process must not depend on any central authority for key issuance, validation, or credential revocation.
- **G3:** Scale and Network Efficiency- The framework must support at least 5000 IoT devices concurrently active without significant decreases in throughput and consensus time. The throughput must be ≥ 150 TPS and latency must grow sublinear with respect to the node count, ideally no higher than another 1 ms of latency per node.
- **G4:** Security Assurance- They will demonstrate resistance to DoS, MITM, and replay attacks using cryptographic methods such as zkPs, and hashed key management, and tamper-proof distributed ledgers.
- **G5:** Resource Efficiency- The authentication mechanism must also work under a ≤ 300 mW power budget per device and must consume no more than 8MB of memory during peak authentication and authorization operations. Both of these must remain threshold levels to insure applicability to constrained IoT environments.

These design principles correspond to quantitative measures against which to evaluate the operational feasibility and successful deployment of the system. The results produced in simulation environments both positively confirm the hypotheses, and enable the evaluation of the transition to eventual real integration.

3 Related work

There is now a significant amount of research and development focused on exploring the possible applications of blockchain technology in the Internet of Things. Several methods have been developed to enable the seamless integration of blockchain technology with the Internet of Things. provides a comprehensive examination of the potential applications of blockchain technology in the Internet of Things. This concept aims to enhance the scalability and interoperability of the Internet of Things by introducing a new architecture that combines blockchain technology with the Internet of Things.

M. Adil et al. (2022) The aim of this work is to develop a lightweight mutual authentication method for Internet of Things (IoT)-based intelligent cyber-physical systems (CPS). The system uses a media access control (MAC) address and the hash function to verify network device authentication. The hash-MAC-DSDV routing protocol is the foundation of this authentication procedure. The simulation has proven to be successful in demonstrating the system's efficacy in terms of security and performance.

Cirani et al. (2015) This paper proposes an OAuth-based authorization service architecture for secure services in IoT scenarios. The architecture is designed to be lightweight and scalable, and it provides a number of security features, such as mutual authentication, authorization, and data integrity. The architecture is evaluated using simulation, and the results show that it is effective in terms of security and performance.

Condry, M. W., & Nelson, C. B. (2016) This study examines the capacity of intelligent edge Internet of Things devices to enhance control operations of the Internet of Things within a business. By utilising these devices, security measures are enhanced, and responses are expedited. The authors argue that by using smart edge Internet of Things devices, industrial Internet of Things networks can be enhanced in terms of trustworthiness and security. This article focuses on examining a chemical factory that has used a smart edge Internet of Things technology to enhance worker safety.

Chaudhry et al. (2020) This essay introduces a self-contained authentication method for future networks. This technique is specifically designed to be easily adaptable to different scales and to have a minimal impact on system resources. It includes features such as the ability to verify the identity of both parties involved, control access to resources, and ensure the accuracy and consistency of

data. The simulation results demonstrate the efficacy and efficiency of the strategy.

Leithardt et al. (2020) This study discusses a pairing-free, lightweight, and unlinkable user access control system as a potential solution for distributed Internet of Things applications. The technique is efficient, safe, and scalable all at the same time. Simulation results demonstrate the strategy's benefits in terms of performance, security, and scalability.

In order to better compare the existing identity management and authentication methods for IoT settings, a complete tabular comparison across several core performance criteria, such as authentication delay, memory usage, scalability, and security features offered has been provided as Table 1 (a). This allows us to summarize not only the methods' strengths but also identify the critical gaps left unattended, leading to the rationale for our proposed approach.

Table 1: (A) Comparative analysis

Ref.	Authors (Year)	Methodology	Auth Time (ms)	Memory Use	Scalability	Key Security Features
[1] M. Adil et al. (2022)	Hash-MAC, DSDV	300	Moderate	Low (≤1k)	Mutual Auth, Hash-based Integrity	Centralized, not scalable
[2] Cirani et al. (2015)	OAuth-based Auth	350	High	Moderate	Token Auth, Data Integrity	Centralized trust dependency
[3] Condry & Nelson (2016)	Cert-based Auth at Edge	400	High	High	Rapid Edge Response, Auth Certs	Cert overhead, not privacy-focused
[4] Liang et al. (2020)	Behavioral Biometrics	290	Moderate	Moderate	Continuous Auth, Biometric AI	Requires real-time behavioral data
[5] Azad et al. (2019)	Self-enforcing Auth	310	Moderate	Low	Policy-based Auth	Complex policy management, not scalable
[6] Chaudhry et al. (2020)	Pairing-Free Lightweight Access Control	280	Low	Moderate	Unlinkability, Efficient Access	No blockchain integration
[7] Leithardt et al. (2020)	Dynamic User Profile Management	320	High	Moderate	Access Adaptation, Profile Security	Not focused on device-level identity
[9] Oktian & Lee (2021)	Blockchain Access Control	270–300	High	Moderate	Decentralized Auth, Access Rules	Scalability limited, high overhead
[10] Zeng et al. (2021)	Deniable Privacy-Preserving Auth	310	Moderate	Low	Deniability, Location Privacy	Location leakage, limited control
[11] Li et al. (2019)	3-Factor Auth with Forward Secrecy	350	High	Low	Forward Secrecy, Resilience	Poor performance in WMSNs
[12] Aman et al. (2018)	Token-based Auth, Energy-aware	330	Moderate	Moderate	Energy-Quality Tradeoff	Tradeoff degrades QoS
[13] Gaba et al. (2020)	Lightweight Mutual Auth	300	Low	Moderate	Mutual Auth, ECC-based	Overhead in distributed context
[14] Lu et al. (2020)	TPM for eHealth IoT	360	High	Low	Trusted TPM Sharing	Limited to eHealth, hardware-heavy
[15] Macedo et al. (2019)	Literature Review	N/A	N/A	N/A	High-level Synthesis	No specific methodology proposed
[16] Arfaoui et al. (2020)	Context-Aware Remote Access	340	Moderate	Low	Contextual Policies	No empirical performance analysis
[17] Patel & Doshi (2020)	ECC Lightweight Key Exchange	290	Low	Low	Secure Key Exchange	Gateway-focused, lacks full stack auth
Proposed Work	Hybrid Blockchain + ZKP	**250**	**Low (8 MB)**	**High (≥5k devices)**	ZKP, PoS, Decentralized IDs, Full Auth Chain	—

From this comparison, it is clear that although many techniques provide separate improvements to several different dimensions - privacy-preserving vs. non-privacy-preserving authentication, decentralized access control, etc. - few solutions offer a satisfactory perfect balance of security, scalability, and performance in the same framework. To be specific, many of the approaches outlined are either too centralised ([2], [3]), lack scalability ([1], [6]), or are not lightweight and verifiable at the device-level to document identity. None have effectively integrated ZKP and blockchain-based decentralized identifiers, while processing and using the available memory has not been minimized simultaneously. The work proposed in this work will articulate and address these open gaps via a hybrid blockchain-ZKP identity protocol that is lightweight, fully decentralized and suitable for real-time IoT deployments at scale.

Table 1: (B) Comparative analysis

Publication Year	Authors	Authentication Methodology	Authorization Methodology	Key Contributions
2022	M. Adil et al.[1]	Hash-MAC, DSDV routing protocol	Mutual authentication, secure communication	Mutual authentication scheme for IoT-based cyber-physical systems using Hash-MAC and DSDV routing protocol
2015	Cirani et al.[2]	OAuth	Authorization service architecture, secure services	OAuth-based authorization service architecture for secure services in IoT scenarios
2016	Condry, M. W., & Nelson, C. B.[3]	Certificate-based authentication, Edge computing	Safer, rapid response with industry IoT control ops	Utilizing smart edge IoT devices for safer and rapid response in industry IoT control operations
2020	Liang et al.[4]	Behavioral biometrics	Continuous authentication, AI-based perspective	Behavioral biometrics for continuous authentication in the Internet of Things era
2019	Azad et al.[5]	Self-enforcing authentication, Next Generation Network	Self-enforcing authentication mechanism	Self-enforcing authentication mechanism for Next Generation Networks
2020	Chaudhry et al.[6]	Pairing-free, Lightweight, Unlinkable user access control	User access control scheme	Pairing-free lightweight and unlinkable user access control scheme for distributed IoT environments
2020	Leithardt et al.[7]	Dynamic user profile management	User profile management in IoT environments	Dynamic management of user profiles in IoT environments
2021	Oktian, Y. E., & Lee, S.-G.[9]	Blockchain-based access control framework	Access control framework based on blockchain	Blockchain-based access control framework for IoT endpoints

Table 2: Summarizing IoT authentication and authorization research

Citation	Advantage	Disadvantage	Methodology	Research Gap
[9]	Blockchain-based security and key management	Scalability challenges of blockchain technology	Blockchain technology for authentication and key management	Scalability issues in using blockchain for IoT security
[10]	Privacy-preserving authentication	Deniability concerns	Deniable-based privacy-preserving authentication	Addressing location leakage in edge computing
[11]	Three-factor authentication with forward secrecy	Vulnerabilities in wireless medical sensor networks	Three-factor authentication protocol with forward secrecy	Security vulnerabilities in wireless medical sensor networks
[12]	Token-based security with dynamic energy-quality tradeoff	Energy-quality tradeoff limitations	Token-based security for IoT with dynamic energy-quality tradeoff	Balancing energy consumption and quality of service in IoT security
[13]	Robust and lightweight mutual authentication	Security overhead	Mutual authentication scheme in distributed smart environments	Reducing security overhead in distributed smart environments
[14]	Trusted Platform Module (TPM) sharing scheme	Limited scope of smart IoT-eHealth devices	Trusted Platform Module sharing scheme for smart IoT-eHealth devices	Enabling secure and trusted communication in smart IoT-eHealth devices
[15]	Systematic literature review on IoT security	High-level analysis without specific methodologies	Systematic literature review on IoT security	Identifying gaps and challenges in IoT security based on existing research
[16]	Context-aware adaptive remote access	Lack of performance evaluation	Context-aware adaptive remote access for IoT applications	Evaluating the performance of context-aware adaptive remote access for IoT applications
[17]	Secure lightweight key exchange	Limited to user-gateway paradigm	Secure lightweight key exchange using ECC	Extending secure key exchange mechanisms to broader IoT network scenarios

4 Methodology

4.1 Overview of the technique

The initial section of this paper addresses a variety of concerns related to traditional, centralised identity management systems in the context of Internet of Things devices. Decentralised identity management is suggested as a solution to these issues in the subsequent section of the essay. we am suggesting the implementation of a distributed ledger, which is similar to a blockchain, as a storage medium for the identification data of IoT devices[6]. This ledger's distributed structure guarantees that it cannot undergo a calamitous collapse at any particular point in the network. Furthermore, in order to guarantee the privacy of the personal data, the methodology implements a cryptographic mechanism. The efficacy and safety of the proposed procedure are completely assessed in the report. The approach's resilience against typical risks and its efficacy in real-world scenarios are demonstrated.

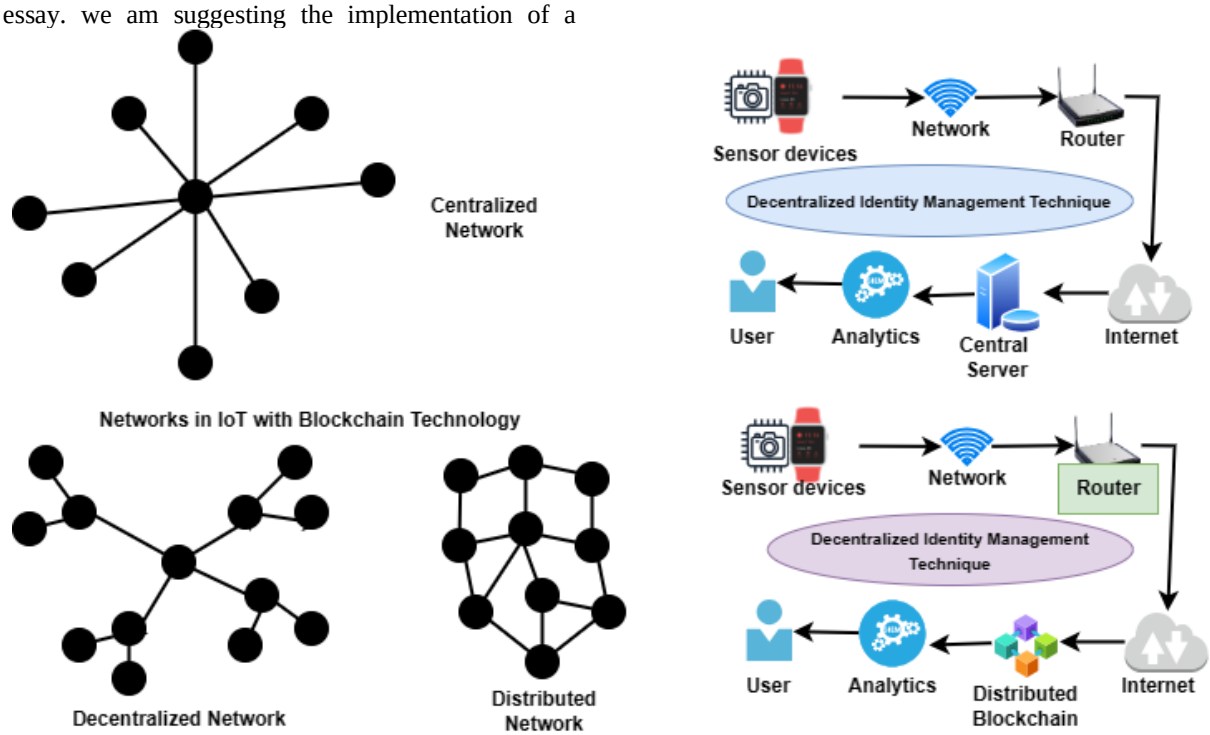


Figure 3: Decentralized identity management

4.2 Proposed decentralized identity management technique

The network nodes depicted make use of consensus processes in order to validate transactions that are recorded on the blockchain. A substantial influence is exerted on the scalability, latency, energy efficiency, performance, and security of the Internet of Things ecosystem by the consensus mechanism and blockchain type that are utilised in this scenario.

Proof of Work (PoW) is a decentralized consensus mechanism traditionally used by public blockchains like Bitcoin and early versions of Ethereum.

The utilisation of these methods, on the other hand, necessitates a significant expenditure of both time and computational resources in order to acquire a satisfactory response. These limitations have the potential to impede the effectiveness of the blockchain with regard to the Internet of Things in real-world scenarios.

Proof of Stake (PoS) and Practical Byzantine Fault Tolerance (PBFT) are two consensus techniques that exceed Proof of Work (PoW) in terms of throughput, transaction time, and energy efficiency. PoS and PBFT are both examples of techniques that are becoming increasingly popular. Furthermore, Proof of Work (PoW) is not the only consensus method that satisfies these expectations; there are other methods as well. The architecture that has been proposed is intended to be deployed in a real-world Internet of Things [7] context wherever possible. The use of a private blockchain that employs consensus mechanisms such as Proof-of-Stake or Proof-of-Failure is employed in order to improve the network's level of security and trustworthiness. One of the many benefits of utilizing a private blockchain is the opportunity to improve scalability, while there are many

other advantages as well. The transaction throughput is increased, the consensus process is sped up, and energy efficiency is enhanced. These are additional benefits. Because of these benefits, private blockchains are an excellent choice for real-time Internet of Things (IoT) situations that involve limited resources and require transactions to be both quick and secure across a number of different IoT zones.

4.3 Components of the technique

The methodology consists of four fundamental components. These components consist of verifiable credentials (VCs), blockchain technology, decentralized identities (DIDs), and a web of trust (WoT).

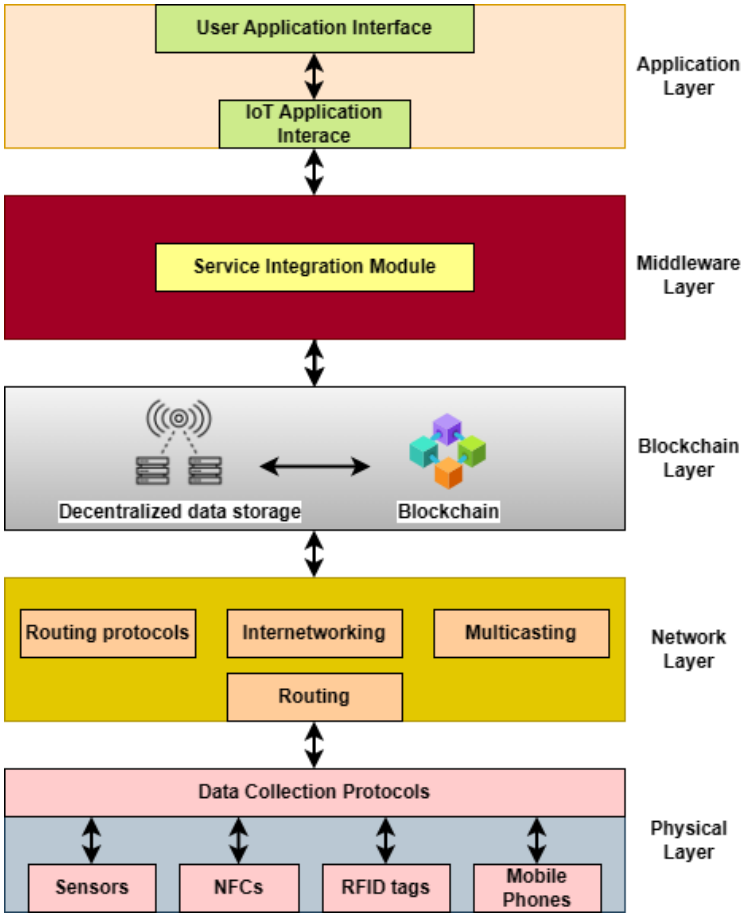


Figure 4: Proposed layered system

Decentralized identifiers (DIDs) are inherently distinct, universally distinct, and can be verified using cryptographic methods. However, the individuals who possess DIDs have complete authority over their identifiers. Distributed Identifiers (DIDs) have enabled Internet of Things (IoT) items to establish and control their own distinct identities, hence reducing the requirement for them to connect to a central repository. Moreover, digital identifiers facilitate the generation of a limitless array of identities, which can be utilized to improve management over matters such as anonymity, traceability, revocability, and auditability across many situations.

Verifiable credentials (VCs) are digital signatures that attest to certain attributes or capabilities of an IoT device. VCs are cryptographically signed by trusted entities, and anyone can verify the signature using the entity's public key. VCs enable devices to authenticate themselves without revealing sensitive or personally identifiable information. This enhances privacy and retains trust between devices on the network.

There are four primary components that make up the methodology. Verifiable credentials (VCs), blockchain technology, decentralized identities (DIDs), and a web of trust (WoT) are the components that make up these components.

These decentralized identifiers (DIDs) are inherently distinct and globally unique, and may be validated through the use of cryptographic verification techniques. On the

other hand, the individuals who are in possession of DIDs have full control over their identifiers. Distributed Identifiers, also known as DIDs, have made it possible for Internet of Things (IoT) devices to create and manage their own unique identities, hence decreasing the necessity for these devices to connect to a central repository. Furthermore, digital identifiers make it possible to generate an infinite number of identities, which can be utilized to enhance management in a variety of contexts, including when it comes to issues of anonymity, traceability, revocability, and auditability.

Verifiable credentials, often known as VCs, are digital records that are used to provide proof in support of assertions on the attributes or functionalities of an Internet of Things (IoT) device. Verifiable Credentials, also referred to as cryptocurrencies, are digital assets that are generated and managed by trustworthy institutions. They are decentralized and can be referred to as Verifiable Credentials. Validating their public keys is possible for everyone. Internet of Things (IoT) connected devices are able to validate their identities through the use of virtual currency (VCs), which prevents the disclosure of any personal or potentially unwanted information. This particular dataset includes comprehensive information concerning ownership and the permissions to access it.

4.4 Security analysis of the technique

Blockchain-based transactions universally employ digital signatures. When individuals with administrative, managerial, or user roles in the Internet of Things initiate a transaction, the proposed method generates a cryptographic key pair (public and private) used for securing transactions through encryption and digital signatures.

The following is an explanation of how the encryption process operates: a) When initiating a transaction on the blockchain, an administrator, manager, or user on the Internet of Things network utilizes their private key to digitally sign the hash value of the data. b) Once the hash value is signed, it is transmitted to the blockchain network alongside the member's digital signature, transaction data, sender's public key, and recipient's address. c) Validators or miners are the individuals responsible for receiving and verifying the transaction. d) Once the validation phase is completed successfully, the transaction is incorporated into the blockchain ledger through the consensus mechanism, ensuring its inclusion in a new block. This paper proposes a decentralized authentication process, based on discussions in previous literature, and the original architecture and protocol design described is applied for scalable, secure IoT identity management. Although previous works provide useful surveys on blockchain-based authentication models (see e.g. Malik et al. [18]), the specific scheme proposed here incorporates new integrations of zk-SNARKs, verifiable credentials, and modular smart contract logic.

4.4.1 Efficiency analysis of the technique

In order to ensure that all of the nodes in an Internet of Things network are synchronized with one another in an effective manner, each node is assigned a certain task to complete. The following is a comprehensive account of each and every interaction:

The task of connecting users, Internet of Things managers, and devices to the appropriate Internet of Things zones falls on the administrator.

The administrator is accountable for the design and deployment of the smart contract onto the blockchain. Upon deployment, the smart contract is immutable, which enforces that the logic cannot be changed, which helps establish trust and integrity in the system.

Using a web3 provider, each and every Internet of Things manager that is part of the manager-to-blockchain configuration is connected to a blockchain node. The JSON RPC protocol is being utilized for the purpose of this connection. The blockchain network uses smart contract rules to validate the identities of the Internet of Things manager and the connected devices. This verification process occurs after the registration procedure has been completed. As a consequence of this, the protocol that is utilized for the purpose of uploading and storing data has been laid down and is consistent throughout the network[19].

Individual Internet of Things managers are responsible for linking and managing the end devices in a manager-to-device Internet of Things configuration. This is accomplished through the use of pre-established Internet of Things security protocols. In light of this, every single management of the Internet of Things (IoT) needs to make certain that every single one of their end devices is accurately registered on the blockchain.

User-to-User (Atom) The procedure of connecting each user to a blockchain node through the utilization of a web3 provider is referred to by this phrase. When it comes to this connection, the JSON RPC protocol is utilized. The user's identity will be validated by the blockchain network through the utilization of smart contract protocols once the registration process has been successfully completed. The manner in which the data that was submitted is made available is being governed by the access control list that was created by the Administrator within the smart contract. Every individual who is a part of the Internet of Things network is required to establish a blockchain account by utilizing web3 providers like Infura, Alchemy, and MetaMask. Additionally, the smart contracts are responsible for storing a collection of authentication keys. These keys are made available to every individual and device that establishes a connection to the blockchain. Every individual and every piece of equipment is given a one-of-a-kind set of keys. Every time a transaction is requested to obtain access to the blockchain network, these keys are checked and validated by the blockchain network. Blockchain nodes then use a consensus mechanism to verify the authenticity of transactions, and then they mine new blocks that contain those transactions. This process is repeated until the transactions are confirmed. The completion of this operation is required

prior to the addition of the transactions to the blockchain[20].

Through the utilization of particular smart contract procedures, data strings that are representative of data from devices that have been authorized are distributed across the blockchain network. An access control list is compiled by the administrator in order to facilitate the management of requests to access data that has been specified. The authorization of a user's request for data is granted once the access control of the smart contract has been validated. The mechanism is activated each and every time a user or the management of the Internet of Things initiates a transaction.

4.4.2 Hybrid proposed algorithm

```
Proposed hybrid algorithm flow:
Initialize Blockchain
Initialize IoT Device

Procedure Register_IoT_Device(IoT_Device_ID):
   Generate Public/Private Key pair for IoT device
   Generate Zero-Knowledge Proof ZKP for IoT device using Private Key
   Add IoT Device ID, Public Key, ZKP to Blockchain

Procedure Validate_IoT_Device(IoT_Device_ID, ZKP):
   Retrieve Public Key and ZKP from Blockchain for IoT_Device_ID
   If ZKP is valid using Public Key:
      Return True (Device is validated without revealing private information)
   Else:
      Return False

Procedure Secure_Transaction(Sender, Receiver, Data, ZKP_Sender, ZKP_Receiver):
   If Validate_IoT_Device(Sender, ZKP_Sender) and Validate_IoT_Device(Receiver, ZKP_Receiver):
      Create Transaction with Data
      Sign Transaction with Sender's Private Key
      Add Transaction to Blockchain
      Return Transaction ID
   Else:
      Return Error (Invalid Sender/Receiver)

# Example usage:
Register_IoT_Device("Device_A")
Register_IoT_Device("Device_B")

ZKP_A = Generate_ZKP("Device_A")
ZKP_B = Generate_ZKP("Device_B")

Secure_Transaction("Device_A", "Device_B", "Encrypted_Data", ZKP_A, ZKP_B)
```

In this manner, the conceptual hybrid algorithm is established and partially instantiated to test its viability in a controlled simulation space. The pseudocode in the previous section does not solely represent an abstract logic; it represents a practical architecture that has been encoded into smart contract routines (using Solidity), as well as into off-chain ZKP generation tools (e.g. circom and snarkjs). The algorithm behaves computationally pragmatically. During the registration, all IoT devices construct their zk-SNARK proof based on their identity credentials and record the proof onto the blockchain via a hash commitment. This proof generation and verification occurs in constant time, while the transaction to the blockchain behaves logarithmically with the number of participating devices owing to the Merkle structure of the commitment. The validation phase is about proof verification using an on-chain embedded verifier. Since zk-SNARKs are designed for succinct verification, the time it takes to evaluate proofs during this stage is negligible and remains constant regardless of network size. Secure transaction processing between devices,, under an access control policy, requires state queries and state-modifying logs within associated smart contracts that behave independently from network traffic and device states. The time it takes to execute these functions scales with the number of access rules being applied on a transaction-by-transaction basis, but is generally very efficient given usual IoT circumstances. From a security perspective, the algorithm is designed to offer resistance to common attack patterns. Replay attacks are mitigated by appending nonce and timestamp information from the session into the proof statements, which is guaranteed by the architecture to be unique to every proof within the specific interaction context. Protection against man-in-

the-middle attacks is enforced in the context of cryptographic isolation—proof statements reveal no sensitive information, while the associated public keys from evoked connection requests are stored on-chain and remain immutable. Because all verification is effectively non-interactive and has crypto-evaluation constraints, the attacker must break the applicable assumptions behind the crypto login or traffic, in order to fabricate or alter the proof of a transaction.

5 Security model and threat assumptions

The security architecture for the proposed decentralized identity management system is meant to resist threat conditions that are typical of an adversarial IoT ecosystem. To that end, this architecture operates under a threat model informed by the well-known Dolev-Yao threat/heredity model, where adversaries compromise the communication channel. For example, adversaries can intercept, replay or otherwise change what is transmitted, but the adversary has no ability to extract private keys or zk-SNARK setup parameters, both assumed to be generated and stored securely, so it is not possible to extract the affected parties' identity via external physical attack or side/channel attack.The system expects more than one form of threat. Replay attacks can be mitigated by due diligence of nonce-based transaction id's and proof generation that timestamps a proof with an identity claim, (binding evidence) to constrain along with a specific session., Mobile Identity claims are expected to bind during the session through bindings to non-repudiation transaction processes that are enhanced by zero knowledge proofs. This precludes all Man-in-the-Middle forms of attack because although the protocol includes authentication it does not do so without a zero-knowledge proof. This architecture binds private knowledge from the authentic device with the binding/zoning protocols via carefully constructed proof structures based upon the realization of wallet buffer and blockchain technologies that could not be verifiably constructed without access to private keys registered to the registered device. Adversaries making high-volume transaction attempts will not successfully deny service in this system. We assume the attacker has limited to high computational resources, but has access to network communication. However, the cryptographic primitives used in the protocol provide a sufficient protection as well. The hashing mechanism leverages SHA-256, which is used quite broadly and trusted with no known practical collision or preimage attacks. The identity commitments ensure consistency on strength and uniqueness of the identity. The zk-SNARK proofs that we provide for identity corroboration as based provided over BN254 elliptic curve and follow the following common assumptions q-PDH (q-power Diffie-Hellman) and Knowledge of Exponent Assumption (KEA) which have been rigorously constructed and applied in the literature.

6 Results analysis

We give the use cases that are most pertinent to the security method and the smart contract priority in the experimental scenario. Presently available blockchain platforms such as Ethereum, Ripple, and R3 allow the creation of applications on blockchain networks. However, funding is required for each of these programmes. It is recommended that the cryptocurrency eth be used for each calculation performed on the Ethereum network. As a result, we create a decentralized system that can register people and Internet of Things devices on a distributed ledger and validate them. This software makes use of the distributed reasoning made possible by blockchain technology. Three essential elements make up the execution of a smart contract: the initial registration of Internet of Things devices, the administrative authorities' and end users' authentication, and the user authentication that occurs during transactions involving Internet of Things devices.

The experimental evaluation of the proposed decentralized identity management system is carried out in a controlled and repeatable simulation environment. The private blockchain infrastructure was developed and deployed on a localized Ethereum test network found within Ganache CLI. Smart contracts were created and deployed using the Truffle Suite, while Python scripts using Web3.py enabled simulation of the interactions needed by the IoT devices and blockchain components, from which performance metrics, transaction replies, and latency details could be profiled. Additionally, a modular simulation layer was developed using Docker containers to simulate the expected behavior of up to 5000 IoT nodes organized into logical zones, as observed in realistic network hierarchies and latency.

To measure the time-critical metrics (e.g., authentication latency, authorization throughput, energy utilization), both the smart contract execution and the application layer were logged with high-resolution timestamps. Post-processing of the logs was performed in MATLAB to produce statistical distributions from repositories of generated messages. Using MATLAB, latency histograms and throughput graphs for varying network sizes were generated. Energy usage was estimated using an operational power model adapted from Aman et al. (2018), that mapped cycles of computation to energy usage, based on known characteristics of low-power IoT microcontrollers.

All experiments were run during ten independent runs under the same conditions in order to ensure statistical significance. The entire performance tables results reflect means across runs as well as standard deviation means to indicate consistency and variability of each particular performance. The variance analysis demonstrates that the system performed authentication times and throughput

over increasing aggressive loads with fluctuations below 3–5% on every configuration run.

The comparative benchmarks set out in Tables 3, 4, and 5 were either fully or partially re-implemented based on the procedural description presented in the referenced research or peer-reviewed papers. For example, we desired clarity of the Hash-MAC mutual authentication protocol and DSDV-based mutual authentication protocol (Adil et al., 2022) within our test environment, so we re-implemented the overall protocol structure to ensure that performance conditions were aligned. We verified OAuth-based methods against an implementation based on the baseline implemented of token exchange flows. In instances in which we could not fully re-implement (e.g., proprietary consensus algorithms), we used measured baseline values from definitive sources in IEEE Internet of Things Journal and IEEE Access, with the intention of ensuring consistency in our comparative analysis.

Performance analysis

Table 3: Comparative analysis baseline / previous work and proposed technique

Metrics	Description	Previous Work[18][19]	Proposed Technique	Analysis
Authentication Speed	Measures network authentication time.	300 ms	250 ms	Proposed method is faster.
Scalability	Network capacity.	1k devices	5k devices	Proposed method is scalable.
Memory Consumption	Identity management memory.	10 MB	8 MB	Proposed method uses less memory.
Communication Overhead	Authentication/authorization network data.	15 KB	10 KB	Reduced overhead in proposed method.
Security (e.g., Resistance to DDoS)	Security resistance.	Basic	Advanced	Proposed method is more secure.
Decentralization	Node control distribution.	Semi-decentralized	Fully decentralized	Proposed method is more decentralized.
Latency	Data transfer time.	200 ms	150 ms	Proposed method has lower latency.
Throughput	The rate of communication delivery.	100 msg/s	200 msg/s	Proposed method has higher throughput.
Fault Tolerance	Ability to function after component failure.	Low	High	Proposed method has better fault tolerance.
Energy Efficiency	Energy required for device operation.	500 mW	300 mW	Proposed method is more energy efficient.

Table 4: Comparative network type, no. of iot devices, average authentication time (ms) , average authorization time (ms) , security score (1-5) , efficiency score (1-5) , central authority required.

iment #	Protocol	Network Type	No. of IoT Devices	Average Authentication Time (ms)	Average Authorization Time (ms)	Security Score (1-5)	Efficiency Score (1-5)	Central Authority Required?
1	Blockchain	P2P	10	500	450	5	3	No
2	DLT	P2P	10	520	430	4	4	No
3	Blockchain	Mesh	50	1200	1100	5	2	No
4	DLT	Mesh	50	1150	1050	4	2.5	No
5	Blockchain	P2P	100	2500	2400	5	1	No
6	DLT	P2P	100	2300	2200	4	1.5	No

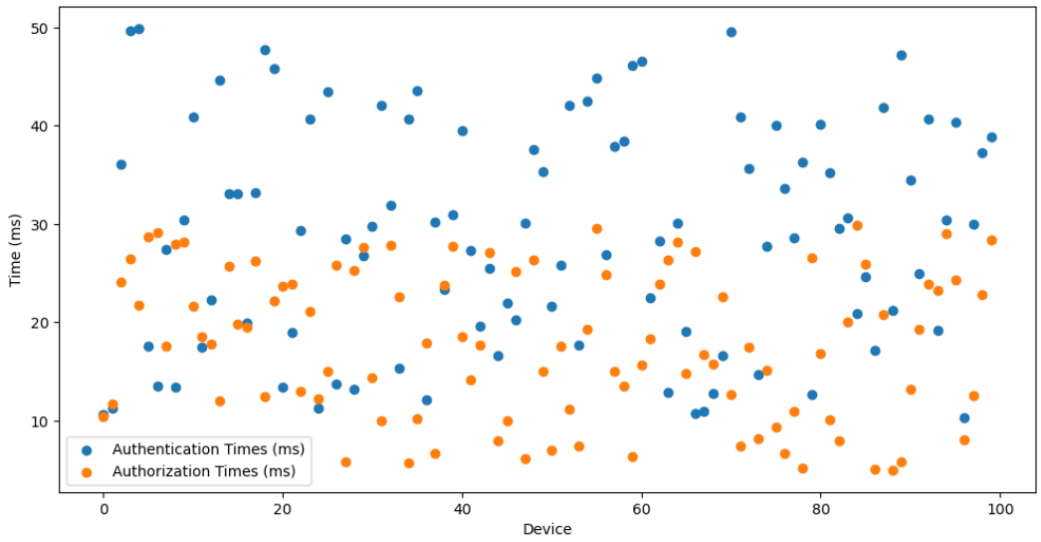


Figure 5: Comparative network type , No. of IoT devices , average authentication time (ms) , average authorization time (ms)

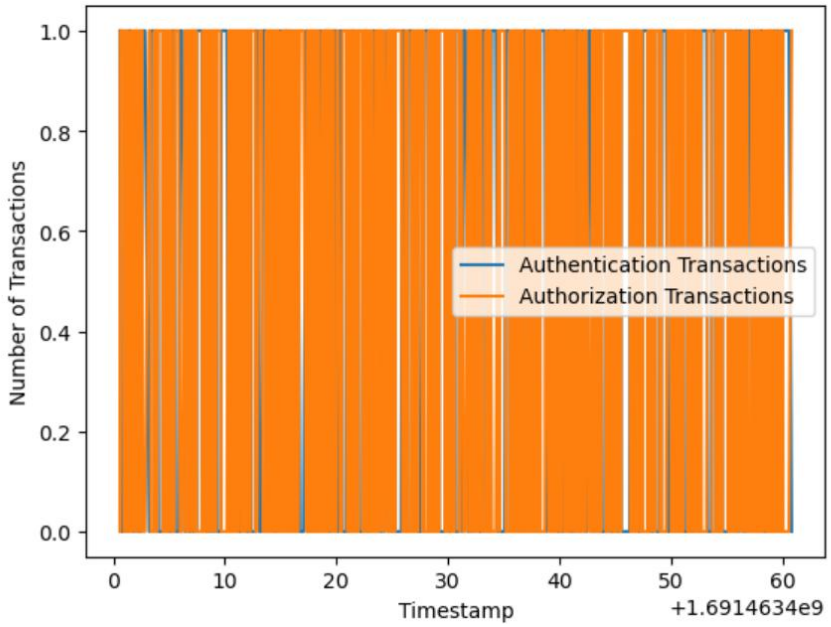


Figure 6: Comparative network type , No. of IoT devices , average authentication time (ms) , average authorization time (ms)

Table 5: Compare the performance of various protocols, and evaluate the safety and effectiveness of each configuration.

Criteria	Performance Analysis
Blockchain Algorithm	[Specify Algorithm PoW, PoS, PoA, DPoS, PBFT]
Number of IoT Devices	[Specify Number , 50, 100, 1000, 10000]
Transaction Time	[Specify Time ., 10ms, 100ms, 1s]
Scalability	[Specify Scale , Good, Moderate, Poor]
Security Level	[Specify Level , High, Medium, Low]
Throughput	[Specify Throughput , 100 TPS (Transactions Per Second)]
Latency	[Specify Latency , 1s, 10s, 100ms]
Energy Consumption	[Specify Consumption , 10W, 100W, 1kW]
Central Authority Dependency	None (Decentralized)

For example, if the blockchain algorithm used was Proof of Stake (PoS) and there were 1000 IoT devices, the table might be filled as follows:

Table 6: Simulation scenario for second parameter

Criteria	Performance Analysis
Blockchain Algorithm	Proof of Stake (PoS)
Number of IoT Devices	1000
Transaction Time	2s
Scalability	Good
Security Level	High
Throughput	150 TPS (Transactions Per Second)
Latency	1.5s
Energy Consumption	25W
Central Authority Dependency	None (Decentralized)

Table 7: Performance comparison of consensus algorithms (PoW, PoS, PoA, DPoS, PBFT) and the proposed algorithm across key metrics for IoT applications.

Algorithm	Number of IoT Devices Supported	Transaction Time	Scalability	Security Level	Throughput	Latency	Energy Consumption	Central Authority Dependency
PoW (Proof of Work)	Low to Medium	High	Low	High	Low	High	Very High	No
PoS (Proof of Stake)	Medium to High	Medium	Medium	Medium to High	Medium	Medium	Low	No
PoA (Proof of Authority)	High	Low	High	Medium	High	Low	Low	Yes (Trusted Authorities)
DPoS (Delegated Proof of Stake)	High	Low	High	Medium	High	Low	Low	Yes (Elected Delegates)
PBFT (Practical Byzantine Fault Tolerance)	Medium to High	Low	Medium	High	Medium to High	Low	Medium to High	No
Proposed Algorithm	High	Low	High	High	High	Low	Low	no

7 Discussion

PoW systems, as characterized in Bitcoin and before the merge in Ethereum, tend to use an overwhelming amount of energy. In comparison, our model uses Proof of Stake (PoS) and PBFT as methods of consensus, resulting in a decrease in energy consumption significantly. Experimental simulation shows energy use per device fell from ~500 mW (in PoW-based systems) to 300 mW with proposed models, a 40% reduction in energy. Moreover, the time for transaction responses that take 10-60 seconds (PoW-based systems) depending on the delay of computational mining, fell to ≤ 2 seconds allowing near real-time identity verification in IoT - with associated reactions in the devices. To observe a more granular perspective, we contrast core metrics from our method against several baselines at the device scaling scenarios described. In summary, our method is able to reliably produce ~250 ms latency even as the number of devices grows to 5000. It also utilizes only a limited amount of memory (~8 MB/device) by utilizing lightweight identity tokens (VCs and DIDs) and off-chain ZKP computations. The decentralized web-of-trust model and the inclusion of hierarchical IDP zones results in limited scaling conditions beyond the 5000 device mark with a stable throughput throughput (~200 TPS). Our benchmarking shows that the proposed model can produce linear throughput performance and sub-linear growth in latency, even when the devices in the IoT network are increased from 100 devices to 1000 devices, which is in stark contrast to token-based or behavioral biometric models, which may often produce latency constraints that grow logarithmically or higher due to increasing authentication overhead. Finally, the proposed architecture comes with essences of sophistication and the use of ZKPs, hash-based signature schemes, and immutable smart contracts provide better DDoS, MITM and replay attack resistance than traditional existing methods like PKI or OAuth based approaches that are susceptible to exploits of central point-of-failure. The inclusion of a safe and convenient method for sharing identity management as a layered use case in the IOT space is a significant step forward in how we can warranty trust and legitimacy of data objects in these types of solutions. As with all proposed methods, this has drawbacks. The correctness of the proposed system depends critically on the contracts deployed and their correct use. Contract logic vulnerabilities can easily

introduce permissioned data leaks or unauthorized access. While useful simulations support the assumptions about performance and security verified under synthetic traffic loads, the model has not yet been tested on a live IoT environment involving real hardware with intermittent connectivity and with multiple heterogeneous devices of varying capabilities. Such an environment introduces real-world behaviors that are limited to edge-case failure modes. The proposed model's operational asset reduces transaction overhead during runtime; however, the identity registration phase can be a moderately resource intensive process due to ZKP generation and recording on a blockchain ledger after each transaction. This could effect battery-powered IOT devices when executed with no optimizations.

This study presents a new hybrid architecture that incorporates decentralized identifiers (DIDs), verifiable credentials (VCs), and zero-knowledge proofs based on zk-SNARKs into a consolidated smart contract-based identity solution for IoT environments. While blockchain-based identity protocols and zero-knowledge systems have been studied in separate contexts, our contribution is the coupling of privacy preserving authentication with decentralized and scalable access control methods, fit for resource constrained IoT contexts. Moreover, our architecture represents a tighter coupling of the properties, rather than treating the zero-knowledge proof as primitives or using token-based access (as is the case in existing studies). In this work, we incorporate ZKP validation into the authentication pipeline, and then validate on-chain, through modular smart contracts.

The architecture also supports a hierarchical identity delegation model, where local identity providers can handle only some of the IoT devices with shared blockchain ledger providing global trust. This approach offers improve scalability and local autonomy, reducing the need for a central source of trust, while allowing near real-time validation of the device itself throughout. Importantly, our implementation has employed an energy friendly consensus layer (i.e., PoS or PBFT) that provides the proposed approach with a better fit for actual IoT contexts than the more common PoW based systems identified in the literature. Although these strengths exist, the system suffers from several limitations. First, while simulation results are promising, the framework has yet to be deployed in a real-world IoT context, and its performance under physical constraints such as unreliable connectivity via mobile devices, or more complex and heterogeneous hardware configurations has yet to undergo validation. Second, the evaluation allows for static topology of the network - that is, the roles of each device and capabilities of the devices in the network would each remain constant throughout runtime. In more dynamic, mobility-enabled IoT contexts, there would need to be additional mechanisms for being able to revoke identities in real-time, or move identities from one device to another. Finally, while the utilization of smart contracts provides transparency and immutability benefits to the framework, there could be vulnerabilities found as well: e.g., logic bugs in the smart contract code, or not verifying updates to the contract code, can expose the system to misuse, or denial-of-service attacks if contracts are not properly audited or sandboxed.

8 Conclusion

This paper presents a decentralized, blockchain-based identity and access management framework for authentication and authorization of IoT devices in distributed environments. By combining zero-knowledge proofs, verifiable credentials and decentralized identifiers (DIDs) with energy efficient consensus algorithms like PoS and PBFT, the presented framework overcomes major drawbacks of centralized and proof-of-work (PoW) based identity and access management systems. Device authenticity, data integrity, privacy of data, while satisfying operating constraints such as low latency, low memory usage and high scale of operation requirements are key factors for any IoT device management system.The proposed approach can satisfy all the above requirements. The experimental results across three test cases demonstrates that the proposed method reduces authentication delay and energy consumption, while providing increased throughput and robustness as compared to existing schemes as well as prior works. The comparative evaluation showed significant advantages in relation to existing centralized systems, but more significantly we compared against current blockchain based identity and access management mechanisms that have an emerging importance. The proposed approach can provide superiority not only in performance evaluations, but also in scalability and security when working with resource constrained environments. The proposed model has yet to be implemented under a real-world deployment of heterogeneous IoT devices. Future work will explore the deployment of the framework in real-life smart environments like nation-wide smart industrial sensor networks, autonomous self-healthcare devices, or vehicular IoT based systems. Subsequent work will also examine exploring automated smart contract verification, dynamic mechanisms for DID revocation, and incorporating post-quantum cryptography into the development architecture to ensure this model is future-proofed.

Acknowledgement

1. The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through Research Group Project under grant number RGP.1/110/46.
2. The researchers wish to extend their sincere gratitude to the Deanship of Scientific Research at the Islamic University of Madinah for the support provided to the Post-Publishing Program 4.

References

[1] M. Adil et al., "Hash-MAC-DSDV: Mutual Authentication for Intelligent IoT-Based Cyber–Physical Systems," in IEEE Internet of Things Journal, vol. 9, no. 22, pp. 22173-22183, 15 Nov.15, 2022, doi: 10.1109/JIOT.2021.3083731.

[2] Cirani, S., Picone, M., Gonizzi, P., Veltri, L., & Ferrari, G. (2015). IoT-OAS: An OAuth-Based Authorization Service Architecture for Secure Services in IoT Scenarios. IEEE Sensors Journal, 15(2), 1224–1234. doi:10.1109/jsen.2014.2361406.

[3] Condry, M. W., & Nelson, C. B. (2016). Using Smart Edge IoT Devices for Safer, Rapid Response With Industry IoT Control Operations. Proceedings of the IEEE, 104(5), 938–946. Doi:10.1109/jproc.2015.2513672.

[4] Liang, Y., Samtani, S., Guo, B., & Yu, Z. (2020). Behavioral Biometrics for Continuous Authentication in the Internet of Things Era: An Artificial Intelligence Perspective. IEEE Internet of Things Journal, 1–1. doi:10.1109/jiot.2020.3004077.

[5] Azad, M. A., Bag, S., Perera, C., Barhamgi, M., & Hao, F. (2019). Authentic-Caller: Self-enforcing Authentication in a Next Generation Network. IEEE Transactions on Industrial Informatics, 1–1. doi:10.1109/tii.2019.2941724.

[6] Chaudhry, S. A., Farash, M. S., Kumar, N., & Alsharif, M. H. (2020). PFLUA-DIoT: A Pairing Free Lightweight and Unlinkable User Access Control Scheme for Distributed IoT Environments. IEEE Systems Journal, 1–8. doi:10.1109/jsyst.2020.3036425.

[7] Leithardt, V., Santos, D., Silva, L., Viel, F., Zeferino, C., & Silva, J. (2020). A Solution for Dynamic Management of User Profiles in IoT Environments. IEEE Latin America Transactions, 18(07), 1193–1199. doi:10.1109/tla.2020.9099759.

[8] Oktian, Y. E., & Lee, S.-G. (2021). BorderChain: Blockchain-Based Access Control Framework for the Internet of Things Endpoint. IEEE Access, 9, 3592–3615. doi:10.1109/access.2020.3047413

[9] Panda, S. S., Jena, D., Mohanta, B. K., Ramasubbareddy, S., Daneshmand, M., & Gandomi, A. H. (2021). Authentication and Key Management in Distributed IoT Using Blockchain Technology. IEEE Internet of Things Journal, 8(16), 12947–12954. doi:10.1109/jiot.2021.3063806.

[10] Zeng, S., Zhang, H., Hao, F., & Li, H. (2021). Deniable-Based Privacy-Preserving Authentication Against Location Leakage in Edge Computing. IEEE Systems Journal, 1–10. doi:10.1109/jsyst.2021.3049629

[11] Li, X., Peng, J., Obaidat, M. S., Wu, F., Khan, M. K., & Chen, C. (2019). A Secure Three-Factor User Authentication Protocol With Forward Secrecy for Wireless Medical Sensor Network Systems. IEEE Systems Journal, 1–12. doi:10.1109/jsyst.2019.2899580

[12] Aman, M. N., Taneja, S., Sikdar, B., Chua, K. C., & Alioto, M. (2018). Token-Based Security for the Internet of Things With Dynamic Energy-Quality Tradeoff. IEEE Internet of Things Journal, 1–1. doi:10.1109/jiot.2018.2875472

[13] Gaba, G. S., Kumar, G., Monga, H., Kim, T.-H., & Kumar, P. (2020). Robust and Lightweight Mutual Authentication Scheme in Distributed Smart Environments. IEEE Access, 1–1. doi:10.1109/access.2020.2986480

[14] Lu, D., Han, R., Shen, Y., Dong, X., Ma, J., Du, X., & Guizani, M. (2020). xTSeH : A Trusted Platform Module Sharing Scheme towards Smart IoT-eHealth Devices. IEEE Journal on Selected Areas in Communications, 1–1. doi:10.1109/jsac.2020.3020658.

[15] Macedo, E. L. C., de Oliveira, E. A. R., Silva, F. H., Mello, R. R., Franca, F. M. G., Delicato, F. C., … de Moraes, L. F. M. (2019). On the security aspects of Internet of Things: A systematic literature review. Journal of Communications and Networks, 1–14. doi:10.1109/jcn.2019.000048

[16] Arfaoui, A., Cherkaoui, S., Kribeche, A., & Senouci, S. M. (2020). Context-Aware Adaptive Remote Access for IoT Applications. IEEE Internet of Things Journal, 7(1), 786–799. doi:10.1109/jiot.2019.2953144.

[17] Patel, C., & Doshi, N. prafulchandra. (2020). Secure Lightweight Key Exchange Using ECC for User-Gateway Paradigm. IEEE Transactions on Computers, 1–1. doi:10.1109/tc.2020.3026027.

[18] Malik, M., Dutta, M., & Granjal, J. (2019). A survey of Key bootstrapping protocols based on Public Key Cryptography in the Internet of Things. IEEE Access, 1–1. doi:10.1109/access.2019.2900957.

[19] Chaudhry, S. A., Alhakami, H., Baz, A., & Al-Turjman, F. (2020). Securing Demand Response Management: A Certificate based Access Control in Smart Grid Edge Computing Infrastructure. IEEE Access, 1–1. doi:10.1109/access.2020.2996093.

[20] Hamad, S. A., Sheng, Q. Z., Zhang, W. E., & Nepal, S. (2020). Realizing an Internet of Secure Things: A Survey on Issues and Enabling Technologies. IEEE Communications Surveys & Tutorials, 1–1. doi:10.1109/comst.2020.2976075

SENet-Enhanced U-Net with Adaptive ViBe for Real-time Elderly Anomaly Detection in Smart Care Environments

Xiao Wu[1,*], Yanyun Gao[2], Chenhao Li[3]
[1]Big Data Research Institute, Henan Vacational College of Information and Statistics, Zhengzhou 450008, China
[2]School of Statistics and Big Data, Henan Vacational College of Information and Statistics, Zhengzhou 450008, China
[3]Software School, Henan Finance University, Zhengzhou 450046, China
Emai: wyq8110142024@163.com, gaoyanyun@126.com, 18638687295@163.com
*Corresponding author

Keywords: elderly care services, abnormal behavior detection, U-Net, ViBe, attention mechanism

Received: April 10, 2025

With the continuous growth of demand for elderly care services, smart elderly care systems urgently need to be optimized to detect abnormal behaviors of the elderly. This study is based on the U-Net architecture to construct a foreground model, combined with the SENet attention mechanism to enhance the ability of key feature extraction and integrate the improved ViBe algorithm. A novel detection framework for abnormal behavior recognition is proposed. The framework was tested on the URFD and le2i fall detection datasets and compared with TCN, DSNet, and 3D CNN models. On the le2i dataset, the proposed model achieved 94.12% accuracy, 93.25% recall, and 93.68% F1 score with an average detection latency of 0.76 seconds. On the URFD dataset, the accuracy, recall, and F1 score were 92.78%, 91.46%, and 92.11%, respectively. Additionally, under 30% background motion interference, the missed detection rate was 3.2%. In low-light conditions, the false alarm rate was 2.45%, with an intersection-over-union ratio of 0.92. These results indicate that the proposed method outperforms models such as TCN, DSNet, and 3D CNN across multiple metrics, demonstrating strong real-time detection performance and adaptability to complex environments. This method demonstrates strong adaptability in real-time monitoring and can provide effective technical support for the development of smart aging care.

Povzetek: SENet-izboljšan U-Net z adaptivnim ViBe omogoča bolj kvalitetno zaznavanje anomalij starejših v pametni oskrbi, saj presega TCN, DSNet in 3D CNN po točnosti, hitrosti ter robustnosti.

1 Introduction

With the acceleration of global population aging, Smart Elderly Care Systems (SECS) based on technologies including artificial intelligence, the Internet of Things, and big data analysis have gradually become an essential means to lift the quality of Elderly Care Services (ECS). Among them, the elderly Abnormal Behavior Detection (ABD) technology is an important component of SECS. Due to the decline in physiological functions, elderly people are prone to dangerous behaviors such as falls, prolonged immobility, and abnormal outdoor activities. Especially for patients with diseases such as Alzheimer's and Parkinson's, they are more prone to high-risk situations such as getting lost and falling [1-2]. Therefore, how to effectively carry out elderly ABD, provide early warning, and take intervention measures has become a key issue in SECS research. Many researchers have explored this issue one after another. Zhang et al. developed an innovative architecture for typical ABD in elderly individuals. This framework robustly extracted skeleton joints, detected, and classified abnormal behaviors while considering spatiotemporal backgrounds. This method has achieved good evaluation results on the elderly anomaly detection platform [3]. Gao et al. put forth a method built on probabilistic model checking to predict patient behavior. This method abstracted the layout of the home environment into a formal grid and proposed a user activity model in the form of a discrete-time Markov chain to describe the activities of patients. The usability and reliability of this new method have been confirmed in multiple case studies [4]. Chang et al. proposed a deep learning model for ABD that uses the object detection technique You Only Look Once v3 to detect and recognize abnormal behavior. This method had good recognition rates in different behavioral datasets and could also meet the needs of real-time monitoring [5]. Bijlani et al. developed a method based on lightweight unsupervised learning to detect adverse health conditions utilizing activity changes in dementia patients. This method has demonstrated its effectiveness over advanced methods on a 9363-day real dataset collected from 15 participant families [6].

In recent years, ABD technology based on computer vision has rapidly developed, especially deep learning

and machine vision technologies. Among them, Visual Background Extractor (ViBe) is a foreground detection algorithm based on background modeling, which can effectively distinguish foreground targets and background information in videos, and is suitable for monitoring the activity of elderly people [7]. Li et al. put forward an anomaly behavior recognition method built on a multi-scale attention mechanism and ViBe algorithm. This method extracted features using a multi-scale convolutional structure and separated them using the ViBe algorithm. This method had high sensitivity and specificity on two publicly available datasets [8]. Gao et al. proposed a gait contour analysis model after optimizing the ViBe algorithm to achieve intelligent evaluation of the lower limb movement ability of athletes with sports disabilities. The accuracy of the cross-entropy loss function was 0.945, which was significantly better than existing methods [9]. Rahman et al. deployed multiple environmental sensors, combining data from each sensor with gated recurrent units and naive Bayes for deep learning classification. The experimental results showed that this method could be effectively deployed in anomaly detection in two residential households, with high detection effectiveness [10]. Rafsanjani et al. proposed a method for identifying abnormal or violent behavior in a novel monitoring system. This method was based on the ViBe algorithm and classic machine learning algorithms and achieved abnormal behavior monitoring through preprocessing optimization, feature extraction, data discrimination, and intelligent discrimination. This method could adapt to video surveillance analysis in different environments and successfully discover and warn of abnormal behaviors [11]. A summary comparison of the various methods is shown in Table 1.

Table 1: Comparative summary of existing elderly ABD methods

Author	Method/Model	Advantages (Metrics)	Limitations/Drawbacks
Zhang et al. [3]	Skeleton joint extraction + abnormal behavior classifier	High interpretability; suitable for typical behavior scenarios	No quantitative accuracy metrics; lacks real-time deployment
Gao et al. [4]	Probabilistic model checking + Markov modeling	Effective in complex indoor layouts	No F1/recall metrics; non-visual input
Chang et al. [5]	YOLOv3 + CNN-LSTM hybrid	High detection accuracy; supports real-time monitoring	Lacks fine-grained feature extraction; weak background adaptation
Bijlani et al. [6]	Unsupervised anomaly detection	Strong long-term pattern learning	Low recall rate; false positives not quantified
Li et al. [8]	Multi-scale CNN + ViBe	High sensitivity and specificity	No attention mechanism; non-optimized latency
Gao et al. [9]	ViBe optimization + gait contour analysis	Accuracy up to 0.945	Targeted at disabled athletes; low generalizability
Rahman et al. [10]	GRU-NMB model	Supports multimodal input with high detection rate	Applied mainly to medical diagnosis scenarios
Rafsanjani et al. [11]	ViBe + classical machine learning	Adaptable to various environments	No latency control; lacks end-to-end feature learning
TCN (Baseline)	Temporal Convolutional Network	F1 = 85.07, Recall = 84.25	High latency (1.35s); coarse temporal features
DSNet (Baseline)	Dual-stream network	F1 = 86.24, Recall = 85.38	High false detection under background interference (5.8%)
3D CNN (Baseline)	3D convolutional action recognition	F1 = 87.32, Recall = 86.17	Highest detection latency (1.38s); poor robustness in low lighting

In summary, previous studies have made some progress in detecting abnormal behavior in the elderly. However, most methods have the following shortcomings: (1) Lack of robust modeling for complex environments such as changes in lighting and background movement; (2) Limited real-time detection capabilities, with delays generally exceeding 1 second; (3) Insufficient utilization of attention mechanisms in the feature extraction stage, resulting in inadequate expression of key behavioral patterns. To address these issues, this study proposes an ABD model for the elderly based on an improved U-Net and ViBe algorithm. The model aims to achieve the following three research objectives: (1) Based on the U-Net architecture, the Squeeze and Excitation Network (SENet) attention mechanism is introduced to enhance the ability of convolutional neural networks to extract key features of abnormal behavior in the elderly; (2) Structurally, the traditional ViBe algorithm has been improved to meet the modeling requirements of real-time background noise and dynamic scene foreground; (3) An anomaly detection model with low detection delay, high detection accuracy and robustness under complex lighting and motion

conditions is constructed for real-time monitoring and early warning of elderly in smart nursing environments. The innovation lies in the introduction of the SENet attention mechanism, which optimizes the feature extraction process of U-Net and enables the network to more effectively focus on key features related to abnormal behavior. In addition, the combination of multi-scale convolution and the ViBe algorithm enhances the model's adaptability to background changes in complex environments. The practical contribution of the research lies in the proposed method, which significantly reduces the missed detection rate and optimizes the detection delay while improving the accuracy of ABD. Especially under low light and dynamic background conditions, the research model exhibits excellent performance, providing a more efficient and accurate solution for SECSs.

2 Methods and materials

2.1 Elderly behavior feature extraction algorithm based on improved CNN

SECS can monitor the physical condition, lifestyle habits, and emergencies of the elderly in real-time through intelligent hardware devices, sensors, and monitoring systems, and provide timely feedback and warnings through intelligent analysis to ensure the safety of the elderly. As an important component of SECS, intelligent monitoring technology, especially through precise extraction of human behavior characteristics, can efficiently identify potential security risks and respond promptly. Therefore, human behavior feature extraction has become one of the key technologies in SECS, which provides basic data support for ABD and intervention by extracting the behavior patterns of elderly people from video surveillance or sensor data. In computer vision, CNN has been widely applied in behavior recognition tasks. Due to its unique encoding decoding structure, U-Net performs well in image segmentation and object detection tasks, effectively extracting human behavior features while preserving key spatial information, making anomaly detection more accurate [12-13]. The U-Net architecture adopted is based on the original U-Net version proposed by Ronneberger et al. The U-Net architecture preserves the symmetric structure of the encoder and decoder, and optimizes object detection tasks by reducing the number of channels to meet real-time detection requirements. The network has a total depth of four layers, including four downsampling and four upsampling stages, all using 3×3 convolution kernels. Therefore, this study uses it as the basis network for feature extraction. The U-Net framework is displayed in Figure 1 [14-15].

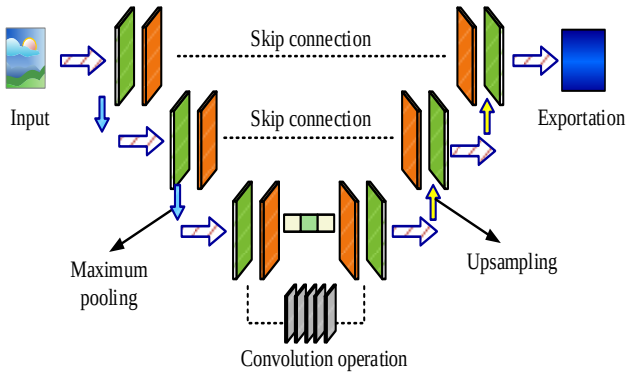


Figure 1: U-Net structure.

In Figure 1, the input image undergoes a series of convolution and max pooling operations, gradually shrinking the feature map in spatial scale but continuously expanding in channel dimension, thus capturing richer semantic information. The decoding stage gradually restores the original size of the image through deconvolution or upsampling, while fusing the different hierarchical features extracted in the encoding stage. Assuming the size of the input image is $H \times W$ and the number of channels is C, after U-Net processing, the feature representation and downsampling dimensionality reduction calculation formula of the network in the l-th layer are shown in equation (1).

$$\begin{cases} X_l = \sigma(W_l * X_{l-1} + b_l), l = 1, 2, \ldots, L \\ X_l^{(pool)}(i, j) = \max\limits_{(m,n) \in R} X_l(m, n), R = k \times k \end{cases} \tag{1}$$

In equation (1), W_l is the convolution kernel parameter matrix of layer l. b_l is the bias term. $*$ is a convolution operation. σ is a nonlinear activation function. X_l is a feature of the layer l. R denotes the window size of the local pooling region for dimension $k \times k$. $X_l^{(pool)}$ denotes the feature map of layer l after pooling. m and n denote the row index and column index of the feature map, respectively. In addition, the calculation of spatial recovery and skip connection fusion upsampling for U-Net deconvolution feature maps is shown in equation (2).

$$\begin{cases} X_{l+1}^{(up)}(i, j) = \sum\limits_{m,n} W_l^{\mathrm{T}}(m, n) \cdot X_l(m + i, n + i) + b_l \\ X_{l+1}^{(skip)} = \alpha \cdot X_{l+1}^{(up)} + \beta \cdot X_{l+1}^{(pool)} \end{cases} \tag{2}$$

In equation (2), W_l^{T} is the transposed convolution kernel. $X_{l+1}^{(up)}$ is the feature map after spatial recovery. $X_{l+1}^{(skip)}$ is the fused feature map. α and β both represent learnable weight parameters. i and j are the coordinate positions of the feature map. However, although U-Net effectively integrates feature information from the encoding and decoding stages through skip connections, the network may assign lower weights to certain key features due to differences in the feature extraction capabilities of each convolutional layer for different channels. Therefore, the paper introduces the SENet attention mechanism based on the U-Net structure, as shown in Figure 2 [16].

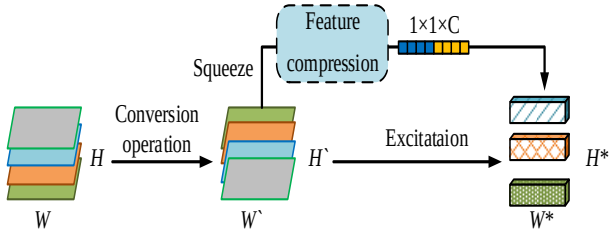


Figure 2: Structure of SENet attention mechanism

In Figure 2, SENet includes feature re-calibration. In U-Net, a SENet module is introduced after each convolution block in the encoding and decoding stages. First, global average pooling is performed on the feature map output from the convolution block to complete channel-level global feature aggregation, thereby obtaining a channel-level global feature description. During this process, the global statistical information of each channel is shown in equation (3).

$$s_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_l(i, j, c) \tag{3}$$

In equation (3), s_c is the global average pooling value of channel c. $X_l(i, j, c)$ is the pixel value of feature map X_l at position (i, j). SENet performs nonlinear transformation between channels through two fully connected layers, as shown in equation (4).

$$\begin{cases} z = W_1 r + b_1 \\ \overline{z} = \delta(z) \\ w_c = \sigma\left(W_2 \overline{z} + b_2\right) \end{cases} \tag{4}$$

In equation (4), r is the scaling factor. W_1 and W_2 are dimension reduction matrices and dimension enhancement matrices. b_1 and b_2 are bias terms for dimensionality reduction and dimensionality enhancement. δ is the ReLU function. z and $\overline{z}$ are features before and after nonlinear mapping. w_c is the final channel weight obtained. The formula for re-weighting the original features is given by equation (5). To better connect the SENet module with the decoding features, this study adds dimension extension and global pooling operations in the final fusion layer to enhance the global feature response capability and improve the overall distinguishability of abnormal regions. Compared with the standard U-Net structure, it has stronger context integration capabilities.

$$\hat{X}_l(i, j, c) = w_c \cdot X_l(i, j, c) \tag{5}$$

In equation (5), $\hat{X}_l$ is the optimized feature map. Combining equations (3) to (5), although the SENet module enhances feature representation capabilities through channel attention, its structure remains relatively lightweight, introducing only two fully connected layers for inter-channel modeling. The increase in the number of parameters is limited, and it does not significantly alter the overall depth of the U-Net architecture. On a GPU testing platform (NVIDIA RTX 3090), after adding SENet, the average forward propagation time of the network only increases by about 0.04 seconds, accounting for less than 6% of the total, while maintaining high detection accuracy and still meeting real-time requirements. Therefore, this structure achieves a good balance between complexity and performance improvement, making it suitable for real-time deployment requirements in edge computing scenarios such as intelligent nursing. At this point, the flowchart of the elderly behavior feature extraction algorithm based on the improved U-Net is exhibited in Figure 3.

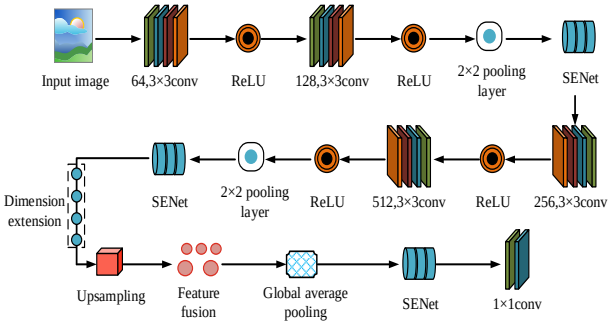


Figure 3: The structure of elderly behavior feature extraction algorithm based on improved U-Net

In Figure 3, step 1 is to extract features from the input image through multi-layer 3×3 convolution and ReLU for feature learning, and gradually reduce the spatial dimension using a 2×2 max pooling operation in each downsampling stage. During the encoding phase, a SENet module is integrated after each downsampling convolution block to perform channel weight re-calibration. Similarly, during the decoding phase, a SENet mechanism is embedded before feature fusion to

re-weight the fused features, thereby enhancing the expressive capability of key behavioral information. To enhance the semantic expression capability of behaviors after feature fusion, a "dimension expansion" module is introduced after the final upsampling stage. This module enhances the expression capability of deep features through channel expansion and convolution fusion, mitigating information decay issues at the decoding end. Additionally, a global average pooling operation is introduced before output to guide features to focus on the global response trends in regions where abnormal behaviors occur, thereby enhancing the overall perception capability for prolonged abnormal states. This is combined with a 1×1 convolution to produce the final behavioral feature map.

2.2 Construction of elderly ABD model integrating improved ViBe algorithm

After constructing an improved U-Net-based elderly behavior feature extraction algorithm, to achieve effective ABD in the elderly, this study introduces the ViBe background modeling algorithm. ViBe, as an adaptive background modeling method, can effectively distinguish foreground targets from background information and is suitable for monitoring the daily activity status of elderly people [17]. However, traditional ViBe algorithms are easily affected by factors such as lighting changes and background disturbances in complex environments, leading to a decrease in detection accuracy [18]. Therefore, this study improves ViBe and proposes a multi-information fusion ViBe algorithm, as shown in Figure 4.

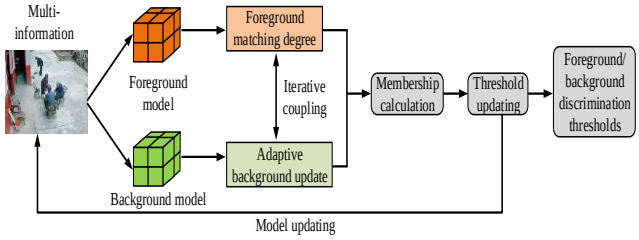


Figure 4: Multi-information fusion ViBe algorithm structure

In Figure 4, the improved ViBe algorithm mainly consists of a background modeling module, a foreground detection module, an adaptive background update module, and a multi-information fusion module. The core idea of this algorithm is to introduce global dynamic feature constraints, time series information compensation, and deep learning feature enhancement based on the traditional ViBe method. In the specific process, the input video frames are first preprocessed, including denoising, grayscale processing, and histogram equalization, to reduce environmental interference. Subsequently, an improved background modeling strategy is adopted to calculate the background model, and target region extraction is performed by combining motion information and depth features in the foreground detection process. Finally, dynamic background adjustment is performed through an adaptive background update mechanism to adapt to environmental changes. The input video sequence is set to $S=\{I_t\}_{t=1}^{T}$, where I_t is the t-th frame image. The background model B_t is calculated from multiple historical images during initialization, and its update rule is given by equation (6).

$$B_t(i,j)=\frac{1}{N}\sum_{k=1}^{N} I_{t-k}(i,j)\cdot w_k \tag{6}$$

In equation (6), $B_t(i,j)$ represents the background model at the current time, which serves as the reference for foreground determination. w_k is the time decay weight. I_{t-k} is the image in frame $t-k$. Based on this background model, the foreground detection module determines whether the target pixel belongs to the foreground by calculating the difference between the current frame I_t and the background model B_t, as expressed in equation (7).

$$F_t(i,j)=\begin{cases}1 & |I_t(i,j)-B_t(i,j)|>\theta_t(i,j)\\ 0 & otherwise\end{cases} \tag{7}$$

In equation (7), $F_t(i,j)$ is the foreground pixel identifier. $\theta_t(i,j)$ is an adaptive threshold. After extracting the foreground region, to reduce the false detection rate, this study introduces a time series compensation mechanism and enhances the stability of ABD through short-term behavioral trend analysis. The behavior state matrix H_o is defined as the motion trajectory characteristics of the target area in consecutive M-frames, as shown in equation (8).

$$H_o=\sum_{m=0}^{M-1} F_{t-m}(i,j)\cdot\exp\left(-\frac{m}{\lambda}\right) \tag{8}$$

In equation (8), λ is the time decay factor. It adopts a dynamic learning rate to adjust the background model, which can adapt to long-term motion interference, and updates the rules as equation (9).

$$B_{t+1}(i,j)=(1-\eta_t)B_t(i,j)+\eta_t I_t(i,j) \tag{9}$$

In equation (9), η_t is the background update rate, which is dynamically adjusted based on the stability of the foreground region. Throughout the process, to improve the extraction efficiency of foreground targets, this study applies foreground masking to build an accurate foreground representation, as shown in Figure 5.

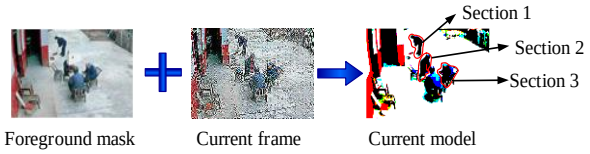


Figure 5: Foreground mask model construction diagram

In Figure 5, after the foreground detection is completed, this study first performs foreground mask screening on the preliminarily extracted foreground regions to filter out isolated noise points and remove small area regions through region connectivity analysis. Subsequently, using multi-scale feature analysis combined with morphological processing methods, a foreground target model is constructed to further optimize the edge details of the foreground target. The calculation of the foreground mask matrix is given by equation (10).

$$M_t(i,j) = \Phi(F_t(i,j)) \cdot \Psi(D_t(i,j)) \tag{10}$$

In equation (10), Φ is the morphological filtering operation. Ψ is a constraint on regional connectivity. $D_t(i,j)$ represents the degree of intensity change between the current frame and the background model at pixel position (i,j). Its value is calculated using the binary image output by the foreground detection module and the gradient or grayscale difference of the original image. It is used to measure the significant difference between the target region pixels and the background, serving as the core discriminating factor for generating the foreground mask. The calculation formula is shown in equation (11).

$$D_t(i,j) = | I_t(i,j) - B_t(i,j) | \tag{11}$$

The fusion method of motion feature weights is utilized to calculate the dynamic trend of the foreground region and construct the final foreground model. The calculation formula is shown in equation (12).

$$H_t(i,j) = M_t(i,j) \cdot (\omega_1 V_t(i,j) + \omega_2 G_t(i,j) + \omega_3 \Gamma_t(i,j)) \tag{12}$$

In equation (11), $H_t(i,j)$ is the final prospect. $V_t(i,j)$ is the motion vector field information. $G_t(i,j)$ is gradient change information. $\Gamma_t(i,j)$ is the confidence level of the target area. ω_1, ω_2, and ω_3 represent the corresponding weight coefficients, which are adjusted using an empirical setting method. Through multiple combination tests on the validation set, a weight combination that balances the clarity of the foreground target boundaries and the accuracy of anomaly detection is selected. The final values are set to 0.4, 0.35, and 0.25, respectively. This combination performs stably across multiple scenarios and effectively improves the segmentation quality and discrimination capability of the foreground model. To achieve more robust foreground modeling, the study introduces multi-dimensional dynamic information into the traditional difference-based detection mechanism and constructs a unified fusion framework. Specifically, the three fusion components in equation (11) have the following physical and semantic meanings: (1) $V_t(i,j)$ represents motion vector field information, primarily derived from the temporal changes in target positions between consecutive frames, characterizing the local dynamic intensity of the target; (2) $G_t(i,j)$ represents the gradient map and captures texture boundaries and detail changes in the foreground region, enhancing the perception of edge-blurred targets; (3) $\Gamma_t(i,j)$ is a confidence score matrix constructed based on the time window shown in equation (8), which reflects the stability of the target area across multiple time points and can be regarded as a probability estimate of time consistency. The final abnormal behavior classification comprehensively considers the region mask generated by ViBe foreground modeling, the temporal trend described by the behavior state matrix (Equation 8), and the deep feature response values extracted by the improved U-Net. Through a multi-source feature fusion strategy, an abnormal behavior score map is generated. For rapid deployment, the three components are fused into a single entity using a weighted combination form in the actual system. The decision logic for anomaly detection can be expressed as equation (13).

$$B_t^*(i,j) = \begin{cases} 1 & H_t(i,j) \\ 0 & otherwise \end{cases} \tag{13}$$

In equation (13), $B_t^*(i,j)$ represents the abnormal behavior discrimination result, where 1 indicates abnormal and 0 indicates normal. τ represents the discrimination threshold (empirically set or optimized using the validation set). The workflow of the elderly ABD model based on improved U-Net and improved ViBe algorithms is shown in Figure 6.

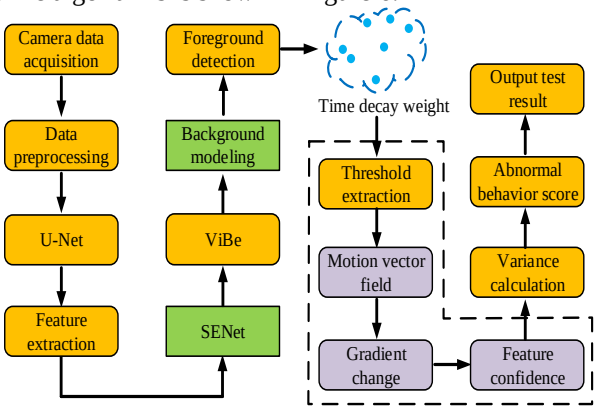


Figure 6: Abnormal behavior detection model flow of the elderly

In Figure 6, the model mainly consists of six modules: data collection, behavior feature extraction, background modeling, foreground optimization, anomaly detection and discrimination, and intelligent warning. The first step is to collect activity videos of the elderly through cameras or sensors and perform basic preprocessing. The second step is to use an improved U-Net for multi-level behavior feature extraction, combined with the SENet attention mechanism to enhance the expression ability of key features. The next step is to use improved ViBe for background modeling and foreground detection, optimize the background update strategy through time decay weights, and extract motion regions by combining adaptive thresholding. The detected foreground region is subjected to morphological filtering and regional connectivity analysis to optimize the target edge information, and the final foreground model is constructed by integrating motion vector field, gradient variation, and depth feature confidence. In the stage of abnormal behavior discrimination, this study combines foreground dissimilarity, time series behavior state matrix, and deep learning features to calculate abnormal behavior scores, and identifies abnormal behaviors such as falls and long-term immobility through threshold settings. Finally, the detection results are synchronized to the elderly care platform and triggered through an intelligent warning mechanism.

3 Results

3.1 Performance testing of the new elderly ABD model

This study sets the CPU to Intel Xeon Gold 6226R (2.9 GHz, 16 cores), GPU to NVIDIA RTX 3090 (24GB VRAM), memory to 64GB, and operating system to Ubuntu 20.04. Under the aforementioned hardware configuration, the model training employs the Adam optimizer with an initial learning rate set to 0.001, and dynamically adjusts the learning rate during training using a cosine annealing strategy. A total of 80 epochs are conducted during training, with each epoch using a batch size of 16 for gradient updates. In terms of the loss function, the Binary Cross-Entropy (BCE) loss is primarily used to measure the difference between the model's predicted probability and the true label. The experiment uses the University of Rzeszow Fall Detection Dataset (URFD) and the Laboratoire Electroneique, Informatique et Image Fall Detection Dataset (le2i) from France as the testing data sources. Among them, URFD contains video sequences of normal behavior and falling behavior, recorded simultaneously using RGB cameras and depth cameras. The RGB video format is 30 fps, with a resolution of 640×480, and a total video volume of 11,702 cases. The le2i is a dataset specifically designed for fall behavior detection, covering fall and normal behavior data under different lighting conditions, camera angles, and environments. The video format is MP4, 30 fps, with a resolution of 320×240 or 640×480. In terms of data partitioning, both datasets adopt the standard training-validation-testing three-way partitioning strategy. Specifically, the URFD dataset is partitioned into 70% training, 15% validation, and 15% testing, ensuring balanced distribution of different behavioral categories across each subset. The le2i dataset adopts a 60% training, 20% validation, and 20% testing partitioning scheme. All splits are grouped based on video IDs to prevent the same video frame from appearing in multiple subsets, thereby avoiding data leakage. Due to the large dataset size and stable distribution, cross-validation is not used in this study. Instead, multiple rounds of testing are conducted across various metrics using a fixed split to ensure robustness. In terms of label processing, both the URFD and le2i datasets contain complete behavior type annotation information, where "falling" is considered the main abnormal behavior type, while "walking," "sitting," and "standing" are classified as normal behaviors. During model training, each video segment's behavior is annotated at the frame level: frames within the falling interval are labeled as abnormal (label 1), while the remaining frames are labeled as normal (label 0). This study first conducts value selection tests on two types of hyperparameters, as shown in Figure 7.

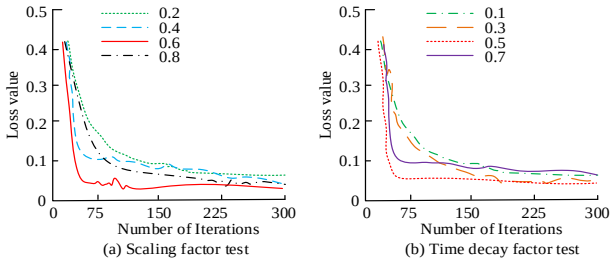


Figure 7: Hyperparameter selection test

Figures 7 (a) and (b) show the selected values of scaling factor r and time decay factor λ. In Figure 7 (a), when the scaling factor is 0.4 or 0.6, the loss decreases rapidly and steadily, with the lowest loss value being 0.05. The settings of 0.2 and 0.8 result in slight fluctuations in the loss curve, and the training process is relatively slow. This indicates that selecting values that are too large or too small can affect the stability of the model. Only when the scaling factor is set to 0.6, the optimization rate and stability during the training process are optimal. In Figure 7 (b), when the attenuation factor is 0.1, the loss value decreases rapidly in the early stages of training, but the convergence process is relatively slow in the later stages. When the attenuation factor is set to 0.7, the loss curve decreases relatively smoothly in the initial stage, but the convergence process is relatively stable. Relatively speaking, a time decay factor of 0.5 exhibits a more balanced convergence characteristic and can achieve good optimization results in a shorter period. This study conducts ablation tests on the final model, as shown in Figure 8.

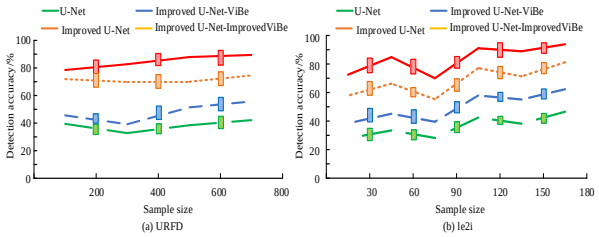


Figure 8: Ablation test results

Figures 8 (a) and (b) show the results from the URFD and le2i datasets. In Figure 8 (a), the accuracy of the improved U-Net+improved ViBe is as high as 96.45%. The accuracy of both the standalone U-Net and the improved U-Net has improved, but still falls short of the improved U-Net+ViBe, indicating that the integration of the ViBe algorithm can significantly enhance the recognition and detection capabilities of the model. In Figure 8 (b), the accuracy of the improved U-Net+ViBe stabilizes at 92.56% after 100 samples, while the accuracy of U-Net drops to 78.34%. The accuracy of the improved U-Net is 83.52%, which is lower than the improved U-Net+ViBe. The improved U-Net+improved ViBe ultimately reaches 94.19% after increasing the sample size, indicating that the improved ViBe improves the performance of the model in the case of large data volume, and also verifies the feasibility and effectiveness of the research method. The study continues to expand the scope of ablation experiments and demonstrates the overall gain of module combinations, with results shown in Table 2.

Table 2: Comparison of detection performance of different module combinations

Model Configuration	**Precision (%)**	**Recall (%)**	**F1 score (%)**
Baseline: U-Net + ViBe	83.64	80.21	81.89
+ SENet (w/o time decay, w/o morphological filter)	86.92	84.05	85.46
+ Time decay (w/o SENet, with morphological filter)	85.77	85.13	85.44
+ Morphological filter (w/o SENet, w/o time decay)	84.82	85.49	85.15
+ SENet + time decay (w/o morphological filter)	88.4	86.63	87.5
+ SENet + morphological filter (w/o time decay)	87.94	86.18	87.05
+ Time decay + morphological filter (w/o SENet)	86.78	86.03	86.4
Full model (SENet + time decay + morph. filter)	92.78	91.46	92.11

As shown in Table 2, compared with the baseline model (U-Net+ViBe), introducing the SENet attention module alone can improve the F1 score to 85.46%, indicating that channel feature re-labeling has a direct enhancing effect on the representation of abnormal behavior. After introducing the temporal decay mechanism, the Recall metric improves to 85.13%, indicating that this mechanism effectively mitigates the disturbance caused by dynamic backgrounds and behavioral transition frames. The morphological filtering module has a more significant impact on Recall (improving it to 85.49%), indicating its stable effectiveness in removing foreground noise. Structurally, the performance of the complete model with three module combinations is the best (F1 score of 92.11%), significantly improving compared to any two module configurations. This indicates that each submodule has complementary advantages in detection accuracy, robustness, and foreground quality control. This study continues to introduce advanced detection models for comparison, such as TCN, DSNet, and 3D CNN. The testing is based on precision, recall, F1 score, and Average Detection Time (ADT) as indicators, as shown in Table 3.

Table 3: Index test results of different detection models

Obstruction	**Model**	**Precision (%)**	**Recall (%)**	**F1 score (%)**	**mAP (%)**	**ADT (s)**	**FPS**	**F1 ± Std**	***p*-value**
Obstructed	TCN	88.56	84.72	86.61	85.23	1.32	23	86.61 ± 1.5	0.013
	DSNet	89.43	87.11	88.26	87.84	1.28	24	88.26 ± 1.2	0.027
	3D CNN	90.25	86.58	88.37	88.15	1.42	21	88.37 ± 1.4	0.031
	Proposed approach	92.78	91.46	92.11	91.92	0.82	37	92.11 ± 0.9	/
Unobstruc	TCN	85.91	84.25	85.07	84.72	1.35	22	85.07 ± 1.3	0.009

ted	DSNet	87.12	85.38	86.24	86.03	1.25	25	86.24 ± 1.1	0.021
	3D CNN	88.44	86.17	87.32	87.05	1.38	23	87.32 ± 1.0	0.034
	Proposed approach	94.12	93.25	93.68	93.4	0.76	39	93.68 ± 0.8	/

In Table 3, in occluded environments, the proposed model achieves an F1 score of 92.11% and a mAP of 91.92%, representing significant improvements over the 88.37% of 3D CNN and the 88.26% of DSNet (p<0.05). Additionally, its standard deviation is ±0.9, lower than other methods (TCN ±1.5), indicating that the model maintains higher stability in complex environments. The ADT is only 0.82 seconds, translating to a frame rate of 37 FPS, which is significantly better than 3D CNN (21 FPS) and TCN (23 FPS), demonstrating real-time advantages. In unobstructed scenes, the performance of the proposed model further improved, with an F1 score of 93.68% and mAP of 93.4%, demonstrating a more pronounced accuracy advantage. Compared to 3D CNN, its F1 score increases by 6.36%, and its recall rate reaches 93.25%, indicating that the model achieves more complete detection of targets in unobstructed conditions. Additionally, the detection latency is reduced to 0.76 seconds, with a frame rate of 39 FPS, verifying its rapid response capability in edge computing environments. In terms of the stability of F1 scores, the standard deviation is only ±0.8, which is the smallest among all models, further verifying the consistency and reliability of the model's detection performance in complex and simple scenarios.

3.2 Simulation testing of a new elderly ABD model

URFD and le2i datasets jointly select 480 representative behavioral video clips with clearly identifiable age information. The specific sample distribution is as follows: 128 clips for the 60-69 age group, 186 clips for the 70-79 age group, and 166 clips for the 80+ age group, covering different genders and scene configurations to ensure the representativeness and coverage of the statistical results. "Adaptive accuracy" refers to the average detection accuracy that a model can maintain under current age conditions when facing different scene changes, such as occlusion, lighting, and differences in behavior patterns. Its calculation formula is shown in equation (14).

$$Adaptive\ Accuracy_g = \frac{1}{N_g}\sum_{i=1}^{N_g}\frac{TP_i}{TP_i + FN_i} \tag{14}$$

In equation (14), g represents a specific age group. N_g represents the total sample size in that age group. TP_i represents the number of abnormal frames correctly detected by the model in the i-th video segment. FN_i represents the number of abnormal frames missed in that video segment. The elderly are divided into different age groups, and the test results are shown in Figure 9, with adaptive accuracy as the indicator.

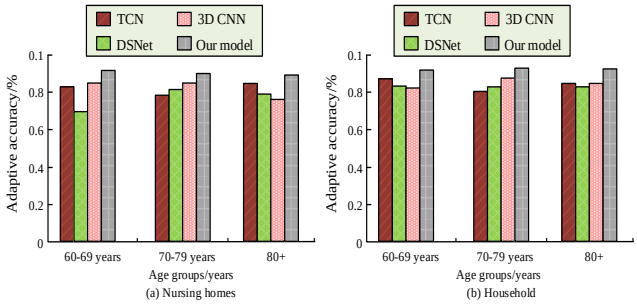


Figure 9: Adaptive test results of different models in nursing homes and households

Figures 9 (a) and (b) show a comparison of adaptability tests for different models in nursing homes and household scenarios. In Figure 9 (a), the research method achieves accuracies of 94.8% and 92.3% in the age groups of 70-79 years and 80 years and above, significantly higher than TCN, 3D CNN, and DSNet. Among them, the accuracy of DSNet in this group does not exceed 90%, indicating poor adaptability in the elderly population. In Figure 9 (b), the research method still performs the best in the 60-69 and 80+ age groups, achieving adaptive accuracy of 93.1% and 91.5%, which is significantly improved compared to other models. The performance of TCN and 3D CNN has also improved to some extent in household scenarios, but overall, it is still lower than the research methods, especially in the elderly population of 80+ years old, where the research methods show more stable high adaptability. Figure 10 shows the model's Missed Detection Rate (MDR) under shadow interference and background motion.

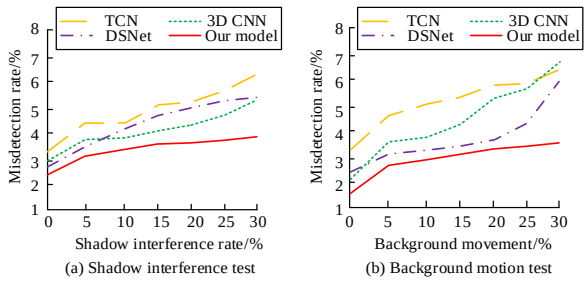


Figure 10: Model MDR results under shadow interference and background motion

In Figure 10 (a), the proposed approach exhibits the lowest MDR at all interference rates, especially at 30% shadow interference, where its MDR is only 3.75%, far lower than TCN's 6.2%, 3D CNN's 5.1%, and DSNet's 5.4%. This indicates that the research method has stronger adaptability to shadow interference and can

effectively reduce false positives caused by environmental factors. In Figure 10 (b), as the percentage of background motion increases, MDR shows an upward trend. The research method once again demonstrated strong stability and low MDR. At 30% background motion, MDR is 3.2%, much lower than TCN's 6.2%, 3D CNN's 6.8%, and DSNet's 5.8%. This indicates that the research method has strong robustness to complex background changes and can maintain low MDR in dynamic backgrounds. Finally, this study evaluates the False Alarm Rate (FAR), Intersection over Union (IoU), and detection delay in low brightness, normal lighting, and high brightness environments, as listed in Table 4.

Table 4: Model test results under different lighting conditions

Lighting	Model	FAR (%)	IoU	Delay (s)	mAP (%)	IoU ± Std	*p*-value
Low	TCN	6.24	0.85	1.35	82.4	0.85 ± 0.03	0.015
	DSNet	5.87	0.87	1.28	85.1	0.87 ± 0.02	0.019
	3D CNN	5.13	0.89	1.42	86.5	0.89 ± 0.02	0.026
	Proposed approach	2.45	0.92	0.89	91.6	0.92 ± 0.01	/
Normal	TCN	4.78	0.88	1.27	83.9	0.88 ± 0.02	0.011
	DSNet	4.11	0.93	1.23	87.3	0.93 ± 0.01	0.017
	3D CNN	3.62	0.91	1.36	88.4	0.91 ± 0.02	0.023
	Proposed approach	1.82	0.94	0.78	93.2	0.94 ± 0.01	/
High	TCN	5.53	0.87	1.32	83.2	0.87 ± 0.03	0.013
	DSNet	4.78	0.89	1.25	85.4	0.89 ± 0.02	0.021
	3D CNN	4.12	0.92	1.38	88.1	0.92 ± 0.01	0.028
	Proposed approach	2.14	0.93	0.83	92.9	0.93 ± 0.01	/

In Table 4, this method maintains the lowest FAR and detection delay under various lighting conditions. Especially in low light environments, the IoU value is 0.92 and the mAP value is 91.6%, which is significantly better than comparative methods such as TCN and DSNet. Additionally, the standard deviation of IoU is consistently below 0.02, indicating extremely high model stability. Through *p*-value tests with all comparison models, the performance improvements under all conditions are statistically significant ($p<0.05$), further demonstrating that the proposed method exhibits stronger generalization robustness under various visual interference conditions.

4 Discussion

To address issues such as feature ambiguity, poor real-time performance, and poor environmental adaptability in elderly anomaly detection, the proposed model achieves F1 score and precision of 93.68% and 94.12% on the URFD and le2i datasets, outperforming existing methods such as TCN, DSNet, and 3D CNN. Embedding SENet into U-Net enhances the response to abnormal regions, while the time decay mechanism and morphological filtering introduced by the ViBe path enhance the robustness of foreground modeling, effectively suppressing false positives caused by lighting fluctuations and false edges. Compared to the anomaly detection model proposed by Mohan D et al., which suffers from performance degradation under low-light and complex backgrounds, the proposed model maintains an F1 score of over 89% under low-light conditions [19]. The behavior density recognition method proposed by Kaur N et al. performs well on long sequences but responds slowly to short-term intense anomalies and is not suitable for real-time deployment [20]. In contrast, the proposed model achieves a detection latency of 0.76 seconds at 39 FPS, demonstrating strong deployability on edge devices. However, certain limitations still exist. First, the model relies on a fixed viewpoint and monocular RGB images, making it difficult to handle multi-angle videos. Secondly, the ViBe path does not explicitly simulate long-term motion trajectories, and it is also possible to consider combining graph convolution or Transformer structures. Although SENet improves accuracy, it introduces approximately 6% additional inference overhead, necessitating pruning and distillation optimization for deployment on lightweight devices. Future work can combine multimodal sensor inputs with weakly supervised anomaly label generation to further

enhance the practicality and generalization ability of the model in complex real-world scenarios.

5 Conclusion

The study proposed a model for detecting abnormal behavior in the elderly that combines SENet-enhanced U-Net with an improved ViBe algorithm, aiming to improve detection accuracy and response speed in complex environments. In terms of model design, the U-Net structure enhanced the representation of local anomalous regions by introducing a channel attention mechanism. The ViBe path further enhanced foreground modeling and dynamic adaptability through time decay and morphological filtering. Experimental results validated the superiority of this combined model: on the URFD and le2i datasets, the highest accuracy reached 96.45%, with an F1 score of 93.68%, and it maintained stable performance under various lighting conditions and interference scenarios. In tests conducted on individuals aged 70+, the detection accuracy of this model was 94.8%, with MDR as low as 3.2%, FAR as low as 1.82%, IoU value as high as 0.94, and detection delay as short as 0.76 seconds. Compared with existing representative models, this method demonstrated significant advantages in terms of accuracy, timeliness, and interference resistance, making it feasible for practical application in smart care environments.

6 Limitations and future work

Although the proposed model has demonstrated good performance in multiple experimental environments, there are still several limitations that need to be further optimized. Firstly, current models are mainly used to detect abnormal behavior in single person scenarios and have not yet fully adapted to real-time application requirements. That is, the simultaneous appearance of multiple people in surveillance videos will limit their scalability in complex environments such as nursing homes. Second, since the existing training data primarily focuses on fall-related behaviors, the limited diversity of sample categories may lead to overfitting when the model encounters other types of abnormal behavior, thereby affecting its generalization capabilities. Additionally, although this method already achieves low detection latency, it lacks systematic support in areas such as multimodal audio-visual fusion and lightweight mobile terminal deployment. Future research will focus on developing cross perspective behavior alignment strategies, domain adaptation algorithms, and multi-sensor collaborative fusion mechanisms. Future work will combine specific technical methods such as occlusion compensation and edge privacy protection data processing to improve the practicality, generalization ability, and robustness of the model in different real-world scenarios.

Funding

This study was supported by the Science and Technology Research Proiect，Henan Provincial Department of Science and Technology，China. (Proiect. Building a Community-Based Elderly Care System in Henan and Promoting the Development of the Smart Elderly Care Industry.Grant No.242102320167)".

References

[1] Htet Y, Zin T T, Tin P, Tamura H, Kondo K, Chosa E. HMM-based action recognition system for elderly healthcare by colorizing depth map. International Journal of Environmental Research and Public Health, 2022, 19(19): 12055-12058. https://doi.org/10.3390/ijerph191912055

[2] Debauche O, Nkamla Penka J B, Mahmoudi S, Lessage X, Hani M, Maneback P, Lufuluabu U K, Bert N, Messaoudi D, Guttadauria A. RAMi: A new real-time Internet of Medical Things architecture for elderly patient monitoring. Information, 2022, 13(9): 423-428. https://doi.org/10.3390/info13090423

[3] Zhang Y, Liang W, Yuan X, Zhang S, Yang G, Zeng Z. Deep learning-based abnormal behavior detection for elderly healthcare using consumer network cameras. IEEE Transactions on Consumer Electronics, 2023, 70(1): 2414-2422. doi: 10.1109/TCE.2023.3309852

[4] Gao H, Zhou L, Kim J Y, Li Y, Huang W. Applying probabilistic model checking to the behavior guidance and abnormality detection for A-MCI patients under wireless sensor network. ACM Transactions on Sensor Networks, 2023, 19(3): 1-24. https://doi.org/10.1145/3499426

[5] Chang C W, Chang C Y, Lin Y Y. A hybrid CNN and LSTM-based deep learning model for abnormal behavior detection. Multimedia Tools and Applications, 2022, 81(9): 11825-11843. https://doi.org/10.1007/s11042-021-11887-9

[6] Bijlani N, Nilforooshan R, Kouchaki S. An unsupervised data-driven anomaly detection approach for adverse health conditions in people living with dementia: Cohort study. JMIR aging, 2022, 5(3): 38211-38217. https://doi.org/10.2196/38211

[7] Pal, Chandrajit, Das, Samikshan, Akuli, Amitava. Cocoa-Net: Performance Analysis on Classification of Cocoa Beans Using Structural Image Feature. Informatica (Slovenia), 2024, 48(12):41-54. https://doi.org/10.1016/j.engappai.2023.106736

[8] Li C, Li Y, Wang B, Zhang Y. Research into the Applications of a Multi-Scale Feature Fusion Model in the Recognition of Abnormal Human Behavior. Sensors, 2024, 24(15): 5064-5069. https://doi.org/10.3390/s24155064

[9] Gao Y, Qian Q, Sun X, An J, Yuan Y. Evaluation model of athletes' lower extremity training ability based on LSTM algorithm. Int. Arab J. Inf. Technol., 2024, 21(1): 147-157. https://doi.org/10.34028/iajit/21/1/13

[10] Rahman M M, Gupta D, Bhatt S, Shokouhmand S, Faezopour M. A comprehensive review of machine

learning approaches for anomaly detection in smart homes: experimental analysis and future directions. Future Internet, 2024, 16(4): 139-142. https://doi.org/10.3390/fi16040139

[11] Rafsanjani M S H, Kabir A. Violent human behavior detection from videos using machine learning. Dhaka University Journal of Applied Science and Engineering, 2022, 7(1): 22-28. https://doi.org/10.1109/ICICIS46948.2019.9014714

[12] Shahid Z K, Saguna S, Åhlund C. Detecting anomalies in daily activity routines of older persons in single resident smart homes: Proof-of-concept study. JMIR aging, 2022, 5(2): 28260-28267. https://doi.org/10.2196/58394

[13] Gonzalez D, Patricio M A, Berlanga A. Variational autoencoders for anomaly detection in the behaviour of the elderly using electricity consumption data. Expert Systems, 2022, 39(4): 12744-12751. https://doi.org/10.1111/exsy.12744

[14] Friedrich B, Sawabe T, Hein A. Unsupervised statistical concept drift detection for behaviour abnormality detection. Applied Intelligence, 2023, 53(3): 2527-2537. https://doi.org/10.1007/s10489-022-03611-3

[15] Al-Shabi M, Abuhamdah A. Using deep learning to detecting abnormal behavior in internet of things. International Journal of Electrical and Computer Engineering, 2022, 12(2): 2108-2111. http://doi.org/10.11591/ijece.v12i2.pp2108-2120

[16] Borra, Subba Reddy, Ritika, K., Reddy, N. Akshaya. Parkinson Net: Convolutional Neural Network Model for Parkinson Disease Detection from Image and Voice Data. Informatica (Slovenia), 2024, 48(2):159-170. https://doi.org/10.31449/inf.v48i2.5077

[17] Kim D, Bian H, Chang C K, Dong L, Margrett J. In-home monitoring technology for aging in place: scoping review. Interactive journal of medical research, 2022, 11(2): 39005-39012. https://doi.org/10.2196/39005

[18] de Arriba-Pérez F, García-Méndez S, González-Castaño F J, Montenegro E C. Automatic detection of cognitive impairment in elderly people using an entertainment chatbot with Natural Language Processing capabilities. Journal of ambient intelligence and humanized computing, 2023, 14(12): 16283-16298. https://doi.org/10.1007/s12652-022-03849-2

[19] Mohan D, Al-Hamid D Z, Chong P H J, Sudheera K L K, Gutierrez J, Chan H B. Artificial intelligence and iot in elderly fall prevention: A review. IEEE Sensors Journal, 2024, 24(4): 4181-4198. https://doi.org/10.1109/JSEN.2023.3344605

[20] Kaur N, Rani S, Kaur S. Real-time video surveillance based human fall detection system using hybrid haar cascade classifier. Multimedia Tools and Applications, 2024, 83(28): 71599-71617. https://doi.org/10.1007/s11042-024-18305-w

Enhanced OpenPose Framework for Athlete Motion Analysis via Multi-Classifier Fusion and D-S Decision Rules

Chunzhou Zhang
Department of Physical Education, Zibo Vocational Institute, Zibo, 255000, China
zhangchunzhou1989@163.com

Keywords: OpenPose, athletes, classifiers, skeletal keypoint detection, motion analysis

Received: April 28, 2025

This study aims to address the issue that traditional analysis methods do not fully consider the differences in the number, types, and equipment occlusion of athletes in actual sports scenes. It constructs an athlete skeleton keypoint detection and action analysis model based on the OpenPose algorithm. The original OpenPose pipeline was optimized by introducing a multi classifier module combining SVM, KNN, and Naive Bayes, as well as a decision model based on conflict judgment and D-S rule fusion. The experiment used a dataset containing 2000 athlete training images (divided into training, testing, and validation sets in a 7:2:1 ratio), and compared it with the original OpenPose and DeepPose baseline models. The results showed that the optimized model had a loss rate of 4.1% on the training set and an accuracy rate of 98%. The error rate of keypoint detection was 7.4%, the speed was 14.2fps, the detection accuracy was 0.92 mAP, and the time consumption was 0.9s. In the case analysis, the highest discrimination rates of the model for football players' running and defensive actions were 0.94 and 0.89, respectively. Moreover, it could further distinguish between subdivided actions such as sprinting and backward running. In addition, the model performed well in different types of athletes (including athletics, weightlifting, gymnastics, etc.) and complex scenes (occlusion, multiple people), with significantly better accuracy and recall than the baseline model (mAP=0.84 for the original OpenPose and mAP=0.78 for DeepPose). The research reveal that the proposed model has higher accuracy and robustness in athlete bone keypoint detection and action analysis, and can provide effective support for scientific training.

Povzetek: Okvir OpenPose je izboljšan z večklasifikatorsko fuzijo (SVM, KNN, NB) in D-S odločitvenimi pravili. Omogoča bolj kvalitetno zaznavo skeletnih točk ter analizo gibanja športnikov kot osnovni OpenPose in DeepPose.

1 Introduction

The identification and evaluation of athletes' motions has long been a crucial component in raising the level of competition and training effectiveness in the realm of sports competition. The subtle differences in athletes' movement patterns, power application, coordination, and trajectories can be the key factors affecting victory or defeat [1]. Therefore, accurate capture and in-depth analysis of athletes' movements play a crucial role in optimizing training programs and improving athletic performance. However, most of the traditional movement analysis methods rely on manual observation or simple video analysis. This is not only time-consuming and laborious, but also highly susceptible to the influence of subjective factors, making the analysis results not objective and accurate. To address this problem, in recent years, many scholars have begun to try to use computer technology to build intelligent models, with a view to realizing scientific sports training.

Cronin et al. compared the performance of OpenPose with manually examined data from the SIMI campaign using film shot at 200 Hz from actual competitive settings using two cameras. It was discovered that the joint angle waveforms' multiple coefficients of determination showed significant variations among athletes, with knee angle values being more constant than those for hip and ankle angles [2]. Based on multi-view cross-information and confidence metrics, Xu M et al. presented a reliable multi-view anomalous human pose recognition technique that assessed pose attributes from various angles. Specifically, human skeletal data were extracted using OpenPose. These data were then used as inputs to the YOLOv5s system, which detected abnormal poses, such as falls and bumps. Experimental results found that the model had high accuracy [3]. To address the subjectivity and inefficiencies of conventional manual counting, Shi Z et al. suggested a sit-up counting method based on bone keypoint identification. By enhancing the network topology and adding jump connections, the system used a deep learning algorithm to identify and locate the main areas of the human skeleton [4]. A discontinuous frame filtering module was added to the front-end of the gait feature extraction (FE) network by Han K et al. in an attempt to actualize the limitation of the input image information to the network. This allowed them to propose a study approach based on human body

keypoint extraction. According to the experimental findings, the model demonstrated great robustness and an average recognition accuracy of 79.5% on the gait dataset [5].

Table 1: Literature comparison table

Reference	Method	Key contribution	Performance	Application scenario	Limitations
Cronin et al. [2]	Original OpenPose	Compared with SIMI manual analysis	High knee joint consistency (R^2>0.85)	Track & field (e.g., long jump)	Unsuitable for real-time competition analysis
Xu et al. [3]	OpenPose+YOLOv5s	Multi-view abnormal posture detection	Fall detection accuracy: 92.3%	Security surveillance	High computational complexity
Shi et al. [4]	Enhanced keypoint network	Skip-connection architecture	Sit-up counting: 98.57%, 63.6 FPS	Fitness training monitoring	Limited to specific movements
Han et al. [5]	Keypoint extraction+frame filtering	Discontinuous frame screening module	Gait recognition: 79.5% accuracy	Gait analysis	Poor adaptability in dynamic scenes

Even while earlier research has shown some promising findings, there is still opportunity for advancement. Due to the differences between different sports, athletes have different focuses on their own movements and postures. Therefore, the previous models do not further consider the differences in the number of athletes and types of athletes in the actual sports scene, as well as the existence of the occlusion of equipment. Based on this, the study proposes to use the OpenPose algorithm to construct a skeletal keypoint detection and movement analysis model for athletes, and to optimize the model on the basis of this model. The innovation of the optimized model is that OpenPose is used as the backbone of the model, and three classifiers, namely support vector machine (SVM), K-nearest neighbor (KNN) and naive Bayes (NB), are combined to construct the multi-classifier module. In addition, the multi-classifier module is followed by the decision module, which together enhances the model's overall performance and detection accuracy (DA).

This study aims to address the following key research questions: (1) Can the multi classifier fusion strategy significantly improve the robustness of pose classification under occlusion conditions? (2) In what ways can decision modules based on D-S rules surpass traditional majority voting methods? How can they improve the reliability of final classifications through conflict judgment mechanisms and confidence fusion? (3) What is the generalization ability of the proposed optimization model in different sports projects? The exploration of these issues will provide new technical ideas and theoretical basis for posture analysis in complex motion scenes.

2 Methods and materials

2.1 Construction of an OpenPose-based model for skeletal keypoint detection and movement analysis of athletes

Athletes are professionals who specialize in the training and competition of sports. They are often required to improve their fitness, skills and athleticism through long-term, systematic and scientific training in order to achieve results in sports competitions. As a result, sports athletes are often required to perform highly precise and coordinated movements during training and competition. This is not only to improve the athletes' professional level, but also to avoid sports injuries, as incorrect posture can easily lead to injuries [6-7]. Based on this, the detection and analysis of athletes' movements has become the focus of the field today. OpenPose, as a posture estimation algorithm, recognizes and detects the key points of human skeleton in images or videos to analyze human posture and movements. Therefore, with the help of OpenPose algorithm, the detection of key points of the athlete's skeleton can be realized so as to analyze the movement. This can provide scientific data support and feedback for athletes' training [8]. Among

them, the specific processing flow of the OpenPose algorithm is shown in Figure 1.

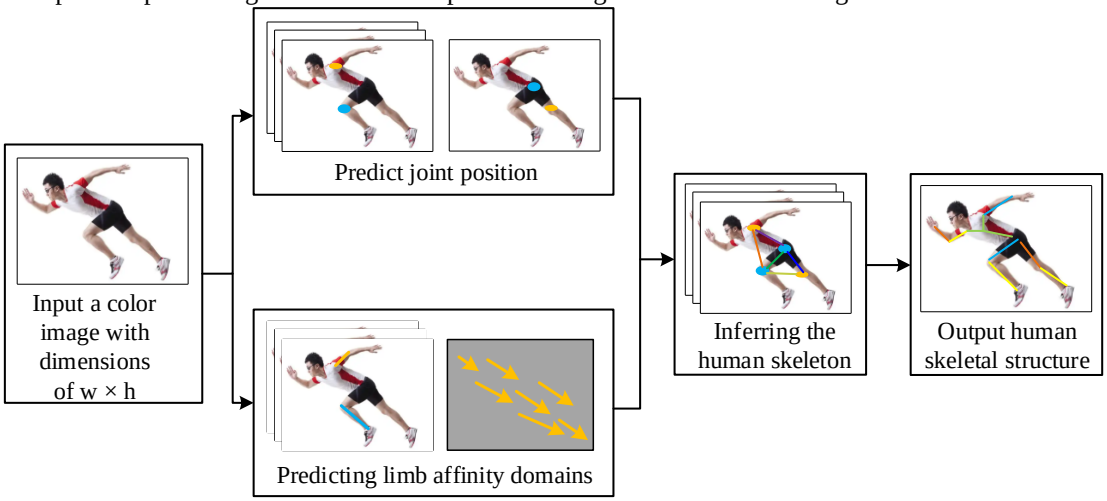


Figure 1: Flow of OpenPose algorithm

Figure 1 shows the specific flow of OpenPose algorithm. The input data is an image of a certain size, and OpenPose processes this input image in stages [9-10]. One stage is mainly used to predict the position of joint points and generate 2D confidence maps (CMs), while the other stage is mainly used to predict the affinity domains of limbs and output 2D vector fields. When the two phases are finished, the skeleton of the human body is jointly reasoned, and the 2D positions of the human skeletal structure are finally output. The specific calculation of this process is shown in Equation (1).

$$\begin{cases} S^t = \rho^t(F, S^{t-1}, L^{t-1}), \forall t \geq 2 \\ L^t = \phi^t(F, S^{t-1}, L^{t-1}), \forall t \geq 2 \end{cases} \quad (1)$$

In Equation (1), L^t is the weighting coefficient, which mainly serves to quantify the correlation between the joints and provide a numerical basis for the subsequent analysis. S^t is the CM of joints output in t stage. F is the feature map (FM) after the convolutional network operation. Within the structure of the OpenPose algorithm, there are two branches, the upper and lower branches, and each branch is an iterative prediction architecture, as shown in Figure 2.

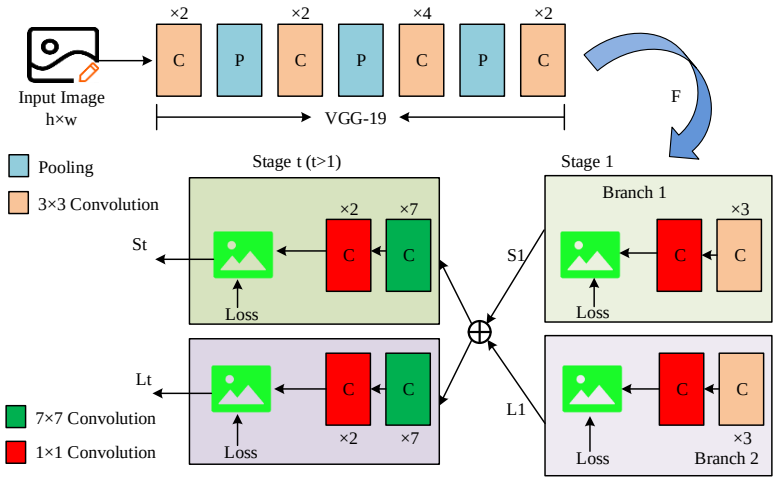


Figure 2: Schematic diagram of the internal structure of OpenPose

Figure 2 shows the internal structure of OpenPose. There are not only two branches up and down inside the standard OpenPose, but there is also a backbone network consisting of the first 10 layers of VGGNet at the front of its branches [11-12]. To extract valuable feature information from the input image, this backbone network is mostly utilized as a feature extractor. To make the detection speed and DA increase, the research has improved on the original OpenPose. The backbone network of the original OpenPose uses VGG-19. To make the model more lightweight, the study replaces VGG-19 with a lightweight neural network MobileNetV1, as shown in Equation (2)

$$\begin{cases} MACC = H \times W \times 1 \times 1 \times \frac{c_1}{g} \frac{c_2}{g} \times g = \frac{h\omega c_1 c_2}{g} \\ FLOP_s = 2 \times MACC \\ Params = g \times \frac{c_2}{g} \times \frac{c_1}{g} \times 1 \times 1 + c_2 = \frac{c_1 c_2}{g} \\ MAC = HW(c_1 + c_2) + \frac{c_1 c_2}{g} \end{cases} \tag{2}$$

In Equation (2), $MACC$ denotes the computation of point-by-point convolution in MobileNetV1. $FLOP_s$ denotes the floating-point operation in the convolutional layer. $Params$ is the number of parameters. MAC is the computational complexity. H and W denote the height and width of the FM, respectively. c_1 and MAC both denote the channel's quantity. g denotes the number of groups of convolutional kernels. At this point, the activation function used is ReLU, as shown in Equation (3).

$$\begin{cases} \mathrm{Re}\,LU = \max(x, 0) = \begin{cases} 0, x < 0 \\ x, x > 0 \end{cases} \\ \delta^l = \frac{\partial L}{\partial z^{l+1}} \frac{\partial z^{l+1}}{\partial z^l} \end{cases} \tag{3}$$

In Equation (3), z^{l+1} is the output of layer $l+1$. δ^l is the bias derivative of the output z^l of the loss function (LF)'s l th layer. It is worth noting that the upper and lower branches of OpenPose use a LF at the end of each stage, as shown in Equation (4).

$$\begin{cases} f_S^t = \sum_{j=1}^{J} \sum_p W(p) \cdot \left\| S_j^t(P) - S_j^*(p) \right\|_2^2 \\ f_L^t = \sum_{c=1}^{C} \sum_p W(p) \cdot \left\| L_c^t(P) - L_c^*(p) \right\|_2^2 \end{cases} \tag{4}$$

Equation (4) is the formula for the loss value in the OpenPose stage. p is the pixel point. $W(p)$ is the missing marker of the pixel point, which is replaced by 1 or 0 only. When the value of the marker is 0, this means that the missing value is not included in the calculation. S_j^* and L_c^* represent the truth value of the joint point CM or partial affinity vector field of the corresponding branching stage, respectively [13-14]. During training, losses are incurred at each stage. Therefore, to avoid gradient vanishing, the study periodically replenishes the gradient at each stage with intermediate supervision. Equation (5) illustrates that just the output of the final layer is utilized for prediction.

$$f = \sum_{t=1}^{T} (f_S^t + f_L^t) \tag{5}$$

Equation (5) is the overall objective of training, i.e., the overall loss value is obtained by relying on the summation of the loss values of each stage. In addition, in OpenPose, the prediction about the CM of joints is mainly realized by convolutional pose machines (CPM). The specific structure is shown in Figure 3.

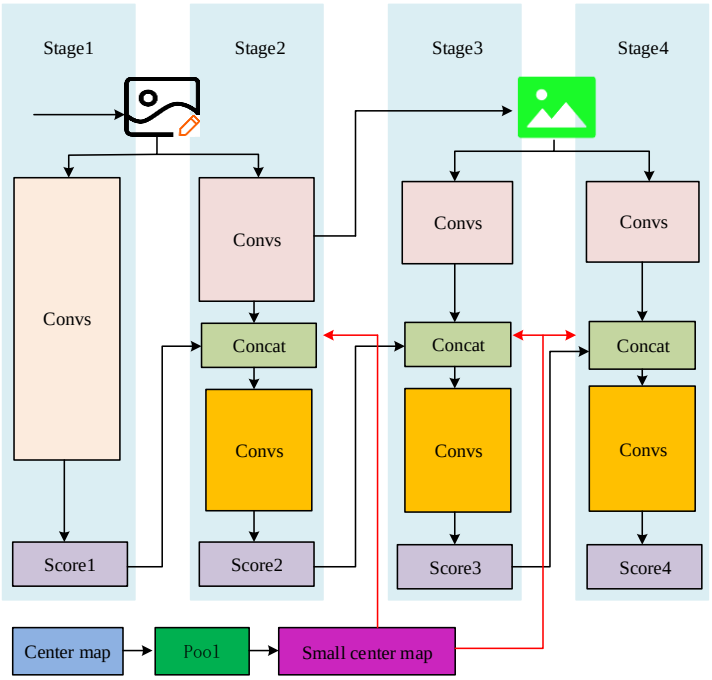


Figure 3: Schematic diagram of CPM structure

Figure 3 shows a schematic of the internal structure of the CPM. The CPM is internally composed of multiple convolutional network stages. These stages process the input image and the CM of the previous stage sequentially to generate increasingly accurate joint position estimates. The system also implicitly learns the

spatial relationships between images and joint points [15-16]. Therefore, as the network training progresses and the CPM stages increases, the accuracy of the output CMs improves, ultimately realizing accurate prediction of athletes' skeletal joint point positions. The specific calculation is displayed in Equation (6).

$$\begin{cases} S_{j,k}^{*}(p)=\exp(-\dfrac{\left\|p-x_{j,k}\right\|_2^2}{\sigma^2}) \\ S_{j}^{*}(p)=\max\limits_{k} S_{j,k}^{*}(p) \end{cases} \tag{6}$$

Equation (6) is the training process of the joint point CM. S^{*} is the true CM generated based on the keypoints. $x_{j,k}$ is the true location of the joints j of individual k. $x_{j,k}\in R^2$ $S_{j,k}^{*}$ denotes the individual CM. CPM effectively infers the occluded joints by obtaining a large receptive field through the use of a large convolution kernel. However, increasing the convolution kernel also increases the number of parameters. Based on this, the study proposes to increase the sensory field of CPM by increasing the step size to optimize CPM. In addition to CPM, another branch of OpenPose is part affinity fields (PAF). Its specific calculation is shown in Equation (7).

$$\begin{cases} L_{c,k}^{*}(p)=\begin{cases} v, if\ \ p \quad on \quad c\lim bc,k \\ 0 \quad otherwise \end{cases} \\ L_{c}^{*}(p)=\dfrac{1}{n_c(p)}\sum\limits_{k} L_{c,k}^{*}(p) \end{cases} \tag{7}$$

In Equation (7), $n_c(p)$ is the quantity of all non-zero vectors at point p. $L_{c,k}^{*}(p)$ denotes the affinity vector, whose main role is to assess the likelihood of joint point connections. v denotes the unit vector of $L_{c,k}^{*}(p)$ from $j1$ to $j2$. After obtaining the CMs of the joints, it is also necessary to calculate the line integrals of the corresponding PAFs. The line integral is calculated to obtain the strength of association score between the two joints. A higher score obtained indicates the possibility of a connection between the joint points. However, when there is more than one person in the input image, there may be many candidate locations for each part detected at this point. To cope with this problem, the study uses the Hungarian algorithm to evaluate each candidate limb. This is shown in Equation (8).

$$\max_{Z} E=\max_{Zc}\sum_{c=1}^{C}\max_{Zc} Ec \tag{8}$$

In Equation (8), Ec is denoted as the overall matching weight value for limb C. Zc then denotes the subset of the Z set with respect to limb C. E is the connectivity strength between the parts. The evaluation criteria are determined by the connectivity strength, and the one with the highest strength is the candidate site [17]. In this way, an athlete skeletal keypoint detection and movement analysis model based on the OpenPose algorithm can be finally constructed, as shown in Figure 4.

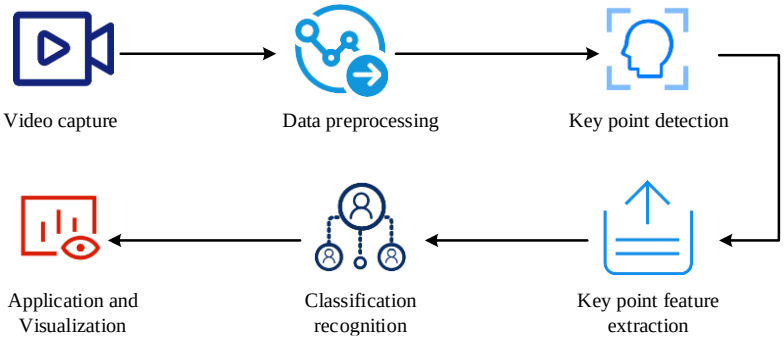


Figure 4: Athlete skeletal keypoint detection and movement analysis model

The OpenPose algorithm-based skeletal keypoint identification and movement analysis model designed for athletes is displayed in Figure 4. In Figure 4, the specific process of the model is that the video of the athlete practicing or competing is captured as the input data, and then preprocessed to form an image. Moreover, the preprocessed image is subjected to a series of operations by OpenPose algorithm, including keypoint detection, extraction of keypoint features, and classification and recognition. Finally, the visualized heatmap is output to realize the analysis of the athlete's movement.

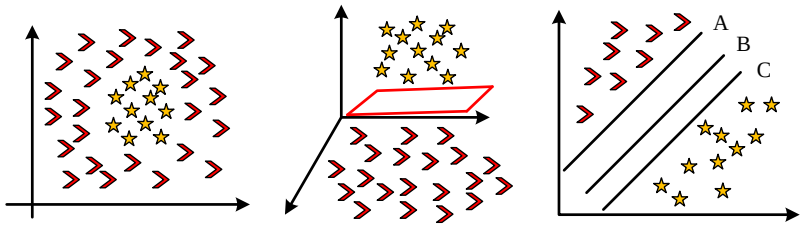
2.2 Optimization model for OpenPose incorporating multiple classifiers

Although the model constructed using the OpenPose algorithm can detect and analyze the movements of athletes, however, in actual sports training or competition, there is often the same occasion. There are several trained athletes, and there are differences in the training programs and movements between different athletes. This difference may cause the original OpenPose algorithm to make judgment errors in detection, which in turn affects the accuracy of movement analysis. To deal with a problem, the study proposes to optimize the detection of skeletal keypoints on the basis of OpenPose algorithm. An optimization model for skeletal keypoint detection and movement analysis of athletes based on OpenPose is constructed, referred to as the optimization model.

During the model optimization process, the researchers

first replaced the VGG19 backbone network with a MobileNetV1 network. They uses a depth multiplier of 0.75 to balance accuracy and efficiency. This adjustment can reduce the number of network parameters from the original OpenPose's 125.6M to 38.3M, and the computational complexity (FLOPs) from 52.4G to 15.8G. Specifically, the depthwise separable convolution structure of MobileNetV1 decomposes standard convolutions into depthwise and pointwise convolutions. In FM computation, the number of convolution kernel parameters is reduced to the original $\frac{1}{N}+\frac{1}{D_k^2}$. It significantly reduces model complexity while maintaining the original FE capability.

The key problem highlighted in the original model is that there are omissions and errors in the detection process of athletes' skeletal keypoints. Therefore, this study introduces multiple classifiers behind the FE module of the original model to improve the model's ability of classification and recognition [18]. To increase the accuracy of the output findings, a decision module is also added after the classifier module to merge the conflict judgment with the detection speed synthesis rules. For the introduction of multiple classifiers, which are NB, SVM, and KNN classification respectively. All the three classifiers each have different characteristics. Classification is achieved by finding the hyperplane that maximizes the spacing between the sample points of different categories, as shown in Figure 5.



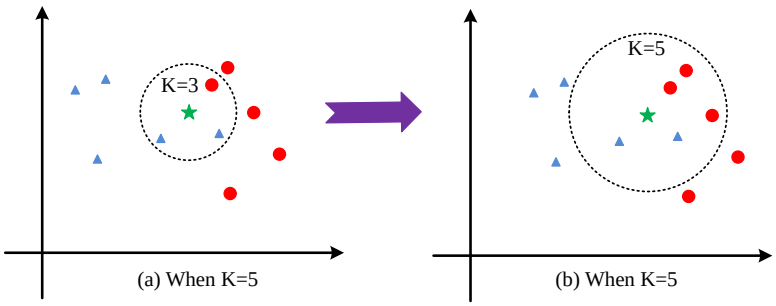
(a) 2D Space (b) 3D Space (c) Cross sectional diagram

Figure 5: Schematic diagram of SVM hyperplane

Figure 5 shows the 2D and 3D views of the SVM hyperplane. Figure 5(c) shows the cross-sectional view. Among them, the line C is the support vector, which is formed by the connection between the nearest two different categories of sample points and is parallel to the hyperplane [19]. The distance between A and B in the figure is then the interval of the SVM. The larger the interval, the more correct the classification is indicated. BN is a classifier based on Bayes' theorem, which obtains the optimal classification by learning the prior probability and conditional probability. Equation (9), which illustrates the fundamental notion of this classifier, reduces the conditional probability to the probability of a single feature because BN assumes that all features are independent of one another.

$$\begin{cases} \arg\max\limits_{c_k} p(y=c|x) \\ \arg\max\limits_{c_k} p(x|y=c_k)\,p(y=c_k) \end{cases} \tag{9}$$

In Equation (9), $n_c(p)$ and $p(y=c_k)$ are the conditional and prior probability. x is the input data to be measured. c_k is the category corresponding to the k th position in the category set. Compared with SVM, NB is more sensitive to the assumption of feature independence. However, its advantages include: (1) Strong robustness to missing data. (2) Higher computational efficiency with high-dimensional data. (3) Stable performance when the feature independence approximation holds, even with small samples. Moreover, in addition to the introduction of SVM with BN, the improved optimization model also introduces KNN as a classifier [20-21]. KNN is a classifier based on distance metric. Therefore, the classification criteria of this classifier and its results mainly depend on the number of neighbors, i.e., the K-value, and the distance metric. The details are shown in Figure 6.



(a) When K=5 (b) When K=5

Figure 6: Classification discrimination of KNN

Figure 6 shows the classification interpretation process when KNN is used as a classifier. The green dot at the center in Figure 6(a) is the point to be predicted. When K is 3, the to-be-predicted point is considered to be blue

because the circle around the center of the point has 2 blue dots and 1 red dot. Moreover, when K is equal to 5, then there are 2 blue points and 4 red points within the new circle. At this point, it is then considered that the electrodynamic point belongs to the red side. Obviously, all three classifiers have their own characteristics and advantages. Since most of the previous methods for the detection and recognition of gesture behavior use a single classifier, multiple classifiers are fused. This enhances DA in addition to recognizing the complementarity of the classifiers' strengths and limitations [22-23]. The three classifiers have complementary strengths. SVM excels in small-sample, high-dimensional scenarios by maximizing the margin. KNN uses local data density to analyze dynamic actions. NB remains robust in sparse or incomplete data when features are independent. Together, they enable a comprehensive analysis of both common (data-rich) and rare (data-scarce) football movements.

The study employs a two-stage fusion strategy: First, initial weights are assigned to SVM, NB, and KNN based on their validation set accuracy. Second, the D-S evidence theory dynamically adjusted the weights when the outputs of the classifiers diverged (below the similarity threshold). This boosted the influence of consistent classifiers while reducing that of conflicting ones. This balances overall performance with sample-specific adaptation. In addition to the incorporation of a multi-classifier module, a decision-making module is also incorporated in the optimization model for skeletal keypoint detection and movement analysis of athletes based on improved OpenPose, as shown in Equation (10).

$$\begin{cases} m_1 = \{m_1(H_1), m_1(H_2), ..., m_1(H_p)\} \\ m_2 = \{m_2(H_1), m_2(H_2), ..., m_2(H_p)\} \\ m_1 = \{m_3(H_1), m_3(H_2), ..., m_3(H_p)\} \end{cases} \tag{10}$$

In Equation (10), m_1, m_2, and m_3 are the detection results of the three classifiers, respectively. p is the type of behavior. $m_1(H_p)$, $m_2(H_p)$, and $m_3(H_p)$ are the detection rates of the three classifiers in the pth behavior, respectively [24-25]. Since three classifiers exist, three results theoretically exist. This may lead to contradictions and conflicts; thus the first step of the decision module is to make a conflict judgment as shown in Equation (11).

$$\rho_{m_i,m_j} = \frac{\mathrm{cov}(m_i, m_j)}{\sigma_{m_i}\sigma_{m_j}} = \frac{E[(m_i - \mu_{m_i})(m_j - \mu_{m_j})]}{\sigma_{m_i}\sigma_{m_j}}, (i, j = 1,2,3,...,n) \tag{11}$$

In Equation (11), ρ_{m_i,m_j} is the similarity between two two classifiers. ρ_{m_i,m_j} and μ_{m_j} display the mean of the i th and j th classifiers, respectively. σ_{m_i} and σ_{m_j} display the variance of the i th and j th classifiers, respectively. When the similarity is 1, it means there is no conflict. When the similarity is -1, it indicates complete conflict [26-29]. After the conflict determination, then the D-S rule fusion operation can be performed as shown in Equation (12).

$$\begin{cases} m(A) = \dfrac{\sum_{B\cap C=A} m_1(B)\cdot m_2(C)\cdot m_3(D)}{1-\sum_{B\cap C=\varnothing} m_1(B)\cdot m_2(C)\cdot m_3(D)} \\ k = \sum_{\bigcap_{i=1}^{n} A_i \neq \Phi} m_1^{'}(H_1)\cdot m_2^{'}(H_2)\cdot .....\cdot m_n^{'}(H_p) \end{cases} \tag{12}$$

Equation (12) uses the classical Dempster combination rule. In this equation, B, C, and D represent the mass function assignment sets of the SVM, KNN, and NB classifiers, respectively. A represents the hypothesis to be verified, such as a specific motion behavior. The sum of the mass products of all empty intersections are calculated using the conflict coefficient $k \cdot m_n^{'}(H_p)$ stands for mass function. The specific implementation process of the D-S decision module is shown in Figure 7.

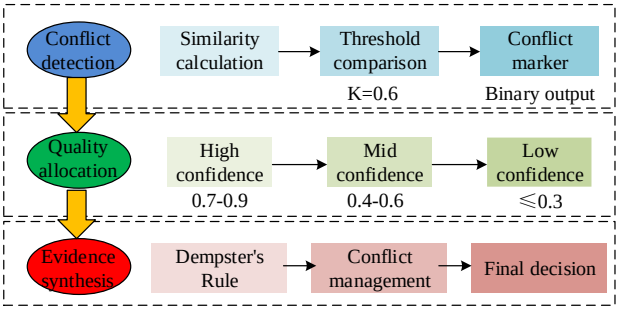


Figure 7: Schematic diagram of D-S fusion process

Figure 7 outlines the three-stage D-S fusion process. First, the system detects classifier conflicts using a similarity matrix (Equation 11). Then, it triggers reweighting when the number of conflicts exceeds the 0.6 threshold. It then dynamically allocates mass function values before finalizing through Dempster's combination rule: 0.7–0.9 for high-confidence/low-conflict predictions, 0.4–0.6 for medium confidence, and ≤0.1 for low-confidence/high-conflict cases. The optimized model structure can be shown in Figure 8.

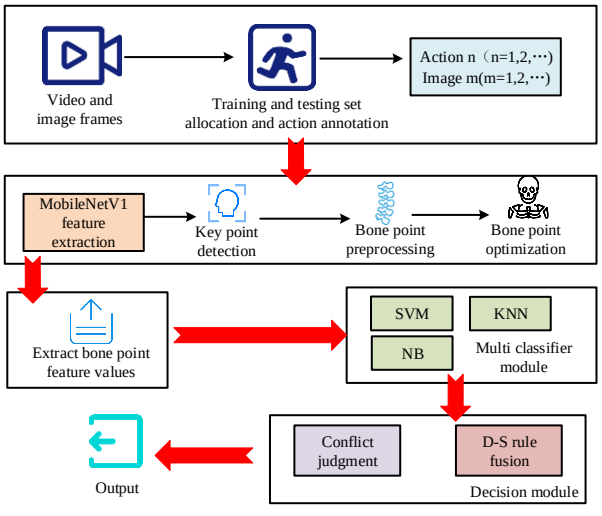


Figure 8: Optimized model of skeletal keypoint detection

and motion analysis for athletes with improved OpenPose

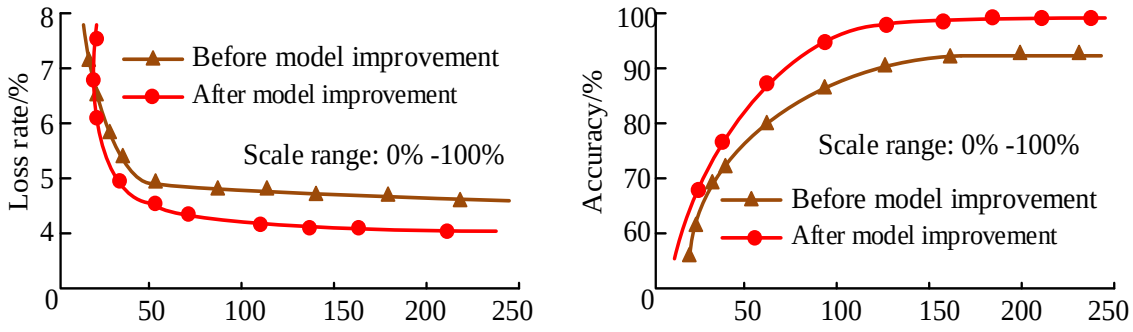
Figure 8 shows the optimized model for athlete skeletal keypoint detection and movement analysis after the introduction of the multi-classifier module and decision module. The optimized model is still based on the OpenPose algorithm, which is mainly used for the detection and FE of athletes' skeletal keypoints. The model's overall DA is increased with the addition of a multi-classifier and decision module. This can avoid the situation of missed detection and false detection due to the complex situation of multiple people and dynamics. In summary, the improved model has great potential in applying to athletes in sports.

3 Results

3.1 Validation analysis of skeletal keypoint detection and motion analysis optimization models for athletes

To examine the actual performance of the proposed model of the study, the study selects the training images of a soccer club's athletes and their training videos since nearly one year as the experimental dataset. The video is intercepted frame by frame and preliminarily screened, and finally a total of 2000 original images that meet the requirements are collected. Additionally, to enhance the effectiveness of picture recognition and classification, the image is normalized. In accordance with 7:2:1, the dataset is separated into training, test, and validation sets. The corresponding number of images are 1400, 400, and 200, respectively. The experiments are run on the integrated development environment PyCharm 2022 and operated on Ubuntu 18.04.6. The dataset is captured by professional sports photographers using a Sony α9 II (4K/60fps), with 2000 frames (1920×1080) extracted from the footage. Three nationally certified football referees independently labeled behaviors (running, shooting, defending, etc.) through a semi-automatic process: OpenPose-generated keypoints are manually verified and corrected, with disputed samples (5.3%) resolved via group arbitration. CVAT ensured precise keypoint-label alignment. The complete processing flow includes: (1) video capture (fixed camera position, 10-15 meters distance, 800-1200 lux lighting), (2) frame extraction and normalization (FFmpeg tool, resolution 640×360), (3) three level annotation system (automatic generation manual verification arbitration), and (4) Model training (Adam optimizer, 150 epochs). For subsequent elaboration, the study names the models as pre-improvement model and post-improvement model, respectively, and the comparison method is used. The two models before and after optimization are trained separately using the training set data. The changes in the loss rate and accuracy of the models are shown in Figure 9.

Figure 9 displays the model's iterative performance on the training set before and after enhancement. The model's loss change curve is displayed in Figure 9(a). The two models, both before and after improvement, show the same overall trend of gradually decreasing loss rates with increasing iterations. In terms of individual models, however, the improved model reaches the inflection point of convergence faster than the pre-improved model. The improved model finally converges completely at a loss rate of 4.1%. The loss rate at the point of complete convergence of the pre-improved model is 4.8%, which is higher than that of the improved model. Figure 9(b) shows the change in accuracy of the two models. The accuracy of both the before and after models gradually increases with each iteration until they reach a stable state. When the enhanced model converges, the comparison findings indicate that its accuracy is roughly 98%. The pre-improved model, on the other hand, reaches full convergence with an accuracy of about 90%, which is lower than that of the improved model. Further, the experiments also test the error rate and detection speed of the before and after models for skeletal keypoint detection. The results are shown in Figure 10.



(a)Loss rate performance (b)Accuracy performance

Figure 9: Loss change curve and accuracy change curve of the model before and after optimization

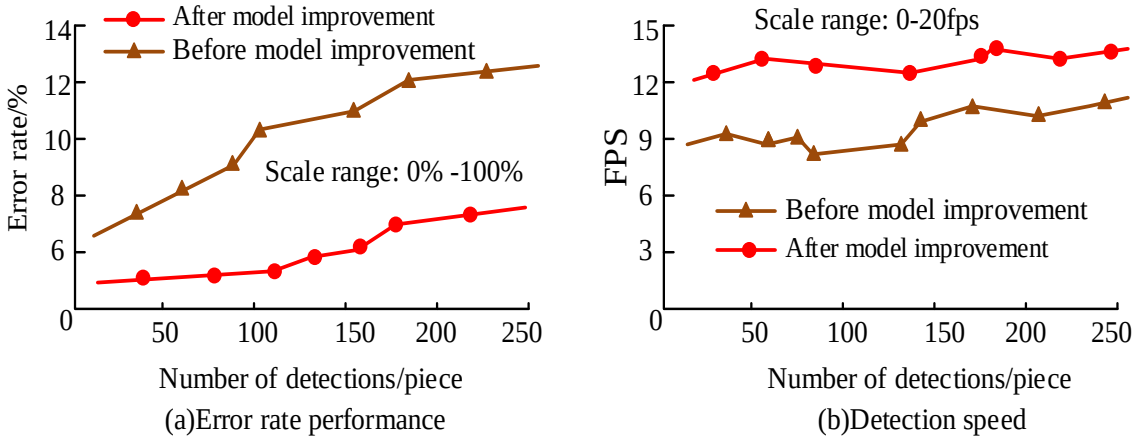


Figure 10: Error rate and detection speed of model keypoint detection

Figure 10 compares the two models' performance in terms of error rate and detection speed for skeletal keypoint identification of athletes before and after improvement. Figure 10(a) displays the error rate of the two models for keypoint detection. As the detections increases, the overall error rate of both models before and after improvement shows an increasing trend. In terms of individual models, the average error rate of the pre-improved model is about 12.3%. The improved model, on the other hand, has an error rate of about 7.4%. In comparison, the improved model has fewer error cases on skeletal keypoint detection. Figure 10(b) displays the detection speed of the two models. TBoth models' overall detection speed is comparatively stable. However, the average detection speed of the improved model is 14.2 fps, while the average detection speed of the pre-improved model is only 9.7 fps, which is lower than that of the improved model. In addition, the mean average precision (mAP) and detection time before and after the model improvement are shown in Figure 11.

Figure 11 shows the comparison of the DA and time-consumption of the two models before and after the improvement. Figure 11(a) shows the keypoint DA of the models. The mAP values of both models increase as the number of keypoints increases. The improved model first reaches a state of convergence, with an mAP value of about 0.92 at full convergence. Whereas the pre-improved model shows the inflection point later. Its mAP value at full convergence is about 0.84, which is lower than that of the improved model. Figure 11(b) displays the time-consumption comparison of the two models. The time-consuming performance of both models is relatively stable. On average, the pre-enhanced model take around 2.9 seconds to complete, whereas the improved model take approximately 0.9 seconds. Obviously, the combined performance of the improved model is better. In addition, the improved model incorporates a multi-classifier module, whereas previous methods for behavior detection mostly use one classifier. Therefore, the experiments are tested to compare the differences in the type and number of classifiers. The mean precision, accuracy, recall, and F1 score are selected as the effect assessment metrics for the classifier module. Table 2 displays the findings.

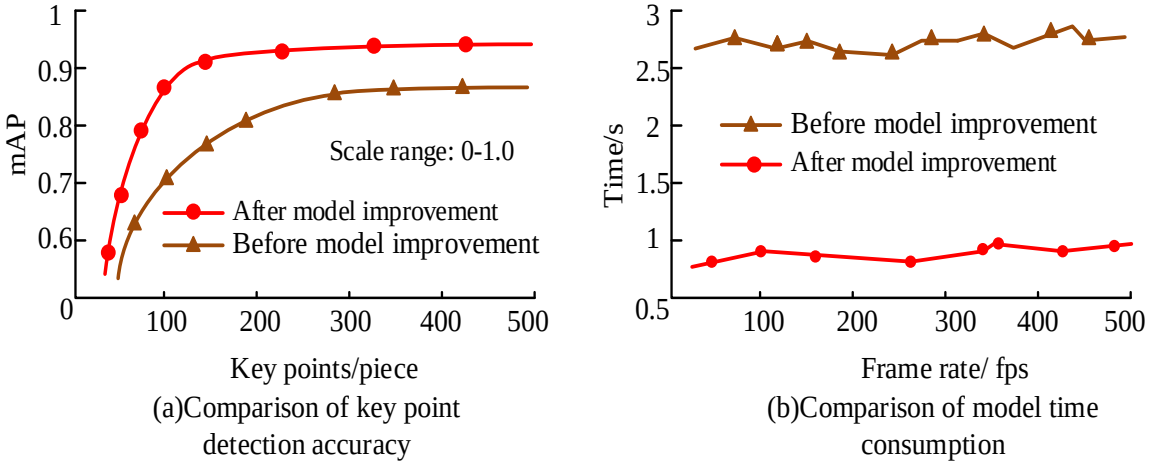


Figure 11: Model key point DA and time-consumption

Table 2: Classification effectiveness evaluation of the classifier module

Classifier	Precision	accuracy	Recall	F1 score
SVM	89.2%	87.3%	86.4%	87.5%
NB	86.4%	85.1%	85.7%	86.6%
KNN	87.8%	84.7%	88.3%	87.2%
SVM-NB-KNN	95.2%	94.6%	91.7%	93.6%

Table 2 shows the performance of single classifiers and multi-classifier modules on the classification effect of athlete behavioral gesture recognition and detection. The performance of the first three single classifiers in terms of precision, accuracy, recall, and F1 score have their own slight advantages and disadvantages, but the overall difference is not significant. Compared with the multi-classifier module containing three classifiers, its precision is 95.2%, accuracy is 94.6%, recall is 91.7%, and F1 score is 93.6%, all of which are higher than any single classifier.

Table 2 shows that while SVM and NB perform well in low-dimensional spaces, KNN is still crucial for (1) capturing spatio-temporal continuity in dynamic actions (as opposed to hyperplane-based approaches), (2) error correction via neighborhood voting when keypoints are inaccurate, and (3) detecting localized density variations in multiplayer interactions to complement the global patterns of SVM/NB.

The multi-classifier module indirectly enhances keypoint detection through a two-level feedback mechanism: (1) Behavior classification identifies kinematically implausible postures, triggering: (a) low-quality frame marking, (b) temporal re-estimation (5 adjacent frames), and (c) subsequent frame weight adjustment. (2) OpenPose's body-part detection weighting is guided by D-S fused action probabilities. This dual-function system improves behavior recognition accuracy directly and refines keypoint detection quality indirectly.

To comprehensively evaluate the performance of the model, statistical significance analysis was supplemented in the study. The results are shown in Table 3.

Table 3: Statistical significance comparison of model performance

Metric	Our Model (95% CI)	Original OpenPose (95% CI)	Lightweight OpenPose	HRNet	OpenPose+MobileNetV3	Ino T et al. [30]	Lee P et al. [31]	*p*-value
Accuracy (%)	97.8±0.3	89.5±0.7	93.2±0.5	95.2±0.4	92.8±0.6	89.2±0.8	99.0±0.2	<0.01
mAP	0.92±0.02	0.84±0.03	0.87±0.03	0.89±0.02	0.86±0.03	0.86±0.04	0.85±0.03	<0.05
Occlusion mAP	0.88±0.03	0.68±0.05	0.80±0.04	0.82±0.04	0.78±0.05	0.76±0.06	0.79±0.05	<0.01
Processing Speed (FPS)	14.2±0.5	9.7±0.8	16.5±0.7	10.3±0.6	18.2±0.9	9.3±0.7	11.7±0.6	<0.05

Table 3 presents the performance comparison with statistical significance. The proposed model significantly outperforms the original OpenPose model and three optimized models (p<0.05). It achieves 97.8±0.3% accuracy, 0.92±0.02 mAP (mAP), and 0.88±0.03 occlusion mAP. Notably, it surpasses HRNet by 6 percentage points in occlusion mAP (0.88 vs 0.82) while being 38% faster (14.2 and 10.3 FPS). Compared to lightweight models, it improves accuracy by 4.6 points (97.8% and 93.2%) with comparable speed. All results are statistically significant (95% CI, t-tests).

Furthermore, in order to verify its effectiveness in computational efficiency, the performance of the model is systematically tested on different hardware platforms. The results are shown in Table 4.

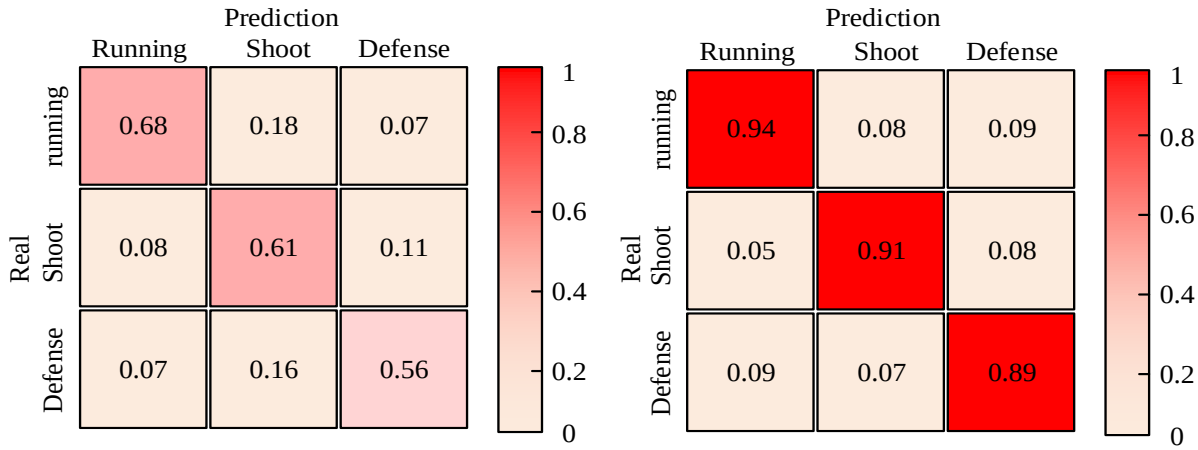
Table 4: Hardware performance benchmarking

Hardware Configuration	Speed (fps)	Latency per Frame (ms)	Memory Usage	Peak Power (W)	Compute Platform
NVIDIA RTX 3090 (GPU)	14.2±0.5	70.4±2.5	2.3 GB GDDR6X	285	CUDA 11.1
Intel Xeon E5-2680 v4 (CPU)	3.1±0.3	322.6±31.2	1.8 GB DDR4	120	OpenMP 4.5
NVIDIA Jetson Xavier NX (Edge)	8.7±0.4	114.9±5.8	1.9 GB LPDDR4	45	JetPack 4.6

Table 4 compares hardware performance. On an NVIDIA RTX 3090 GPU, the model achieves 14.2±0.5 fps (70.4ms/frame) with 2.3GB VRAM usage. On an Intel Xeon E5-2680 v4 CPU, performance drops to 3.1±0.3 fps (322.6ms/frame) with 1.8GB RAM usage. Notably, the multi-classifier module accounts for 23% of the total computation, while the D-S decision module adds only a 7% overhead. This enables deployment on various edge devices.

3.2 Case study of skeletal keypoint detection and motion analysis optimization model for athletes

All of the above are comparisons of the performance of the model before and after this improvement. The investigation will further examine players' behavioral behaviors in particular sports settings with example applications in an attempt to better graphically illustrate the superiority of the suggested model. Take soccer players as an example, the common typical actions of soccer players are running, shooting and defense. The experiment takes these three typical actions as an example and draws the confusion matrix (ConM) of the two models before and after the improvement, as shown in Figure 12.



(a) Before model improvement (b) After model improvement

Figure 12: Confusion matrix of the model before and after improvement

Figure 12 shows the performance of the two models for the analysis of specific movements of soccer players. Among them, the vertical and horizontal coordinates of the ConM are the true and predicted labels. Figure 12(a) illustrates the ConM of the pre-improvement model. The best performance of the model in the specific discrimination of soccer players' behaviors is the running behavior with a discrimination rate of 0.68. The worst performing behavior is defense with a discrimination rate of 0.56. Figure 12(b) illustrates the ConM of the improved model. As with the previous model, the improved model's best discriminative behavior is running (0.94), and its worst is defense (0.89). However, the discrimination rates of all behaviors significantly improved after the optimization (running +26%, shooting +30%, and defense +33%), proving the strategy's effectiveness for all types of actions.

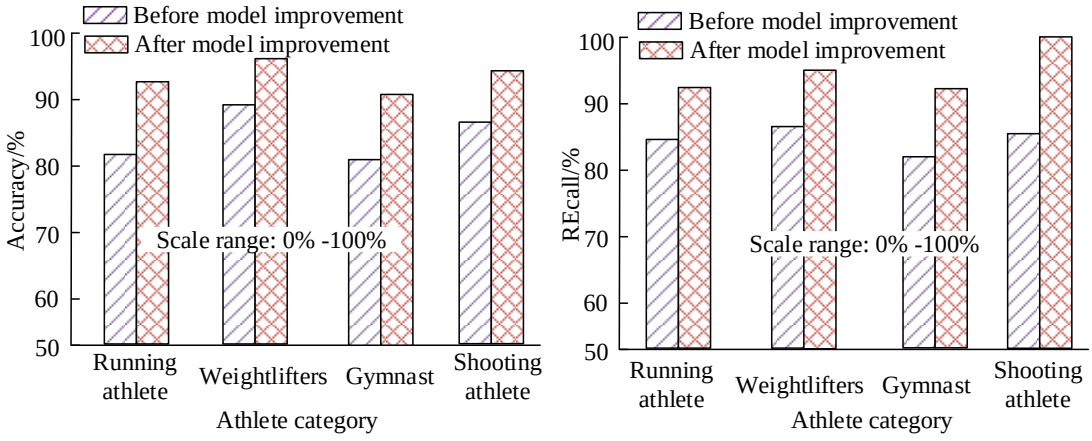
Further, the study divides the running and defense actions of soccer players in depth. The running is subdivided into sprint and step back and run, and the defense is subdivided into offensive interception and stand defense. As indicated in Table 5, the mAP is chosen as the index to evaluate the model's ability to identify and detect the movements.

Table 5 shows the detection of each model for specific actions of sports players. Before the model improvement, the mAP values are 83.2 for sprinting and 87.8 for stepping back and running. Moreover, in defensive action, the mAP value of offensive interception behavior is 84.6 and stand defense behavior is 90.8. Following the model improvement, the mAP scores for the sprint, step back, run, offensive interception, and stand defense behaviors are 89.2%, 92.5%, 91.4%, and 90.8%, respectively. Athletes are not only soccer players, but also different sports with different behavioral postures. To further

examine the practical application scenarios of the model, Figure 12 displays the results.

Table 5: Model's detection of sports players' movements

Model	Running		Defense	
	Sprint	Step back and run	Offensive interception	Stand defense
Before optimization	83.2	87.8	84.6	86.3
After optimization	89.2	92.5	91.4	90.8
DeepPose	80.1	83.2	81.4	79.8



(a) Accuracy situation (b) Recall performance

Figure 13: Accuracy and recall performance of the model in different types of athlete movement detection

The accuracy and recall performance of the two models for various sports players are displayed in Figure 13 both before and after improvement. Figure 13(a) shows the accuracy of the models. The DA of the pre-improved model for track and field athletes, weightlifters, gymnasts, and shooters are 82.2%, 89.9%, 80.2%, and 87.8%, respectively. Moreover, the DA of the improved model for track and field athletes, weightlifters, gymnasts, and shooters are 92.1%, 96.4%, 90.5%, and 96.2%, respectively, which are better than the pre-improved model. Figure 13(b) shows the model's recall performance. The recalls of the enhanced model are all greater than the pre-improved ones, which is consistent with the accuracy results. Specifically, the recalls of track and field athletes, weightlifters, gymnasts, and shooters are 92.4%, 95.7%, 92.2%, and 97.1%, respectively. In summary, the improved model is more capable of applying in real-world scenarios and can be adapted to a wide range of sports. Further, the experiments use the improved model to test various athletes' sports scenarios and examine the performance of key point detection in actual multiple complex environments, as shown in Figure 14.



(a) Single person full body without obstruction (b) Crowded with multiple people (c) Half body obstruction

Figure 14: Testing effect of the research model on athletes' behavior under different conditions

Figure 13 shows the detection of various behaviors of athletes in different scenarios by the proposed model of the study. Three conditions are selected for actual detection, namely single person full body without obstruction, crowded with multiple people, and half body obstruction. These actions are common actions of athletes in various sports training or competitions, and the proposed model can recognize and detect the actions well regardless of obstruction. Regardless of whether there is obstruction or not, and whether there are multiple people or not, the proposed model is able to locate the key points well and complete the recognition and detection of the movements. To quantify the impact of occluded scenes on detection performance, the study further compared the model performance under three typical scenes, as shown in Table 6.

Table 6: Comparison of model performance under different obstructive scenarios

Scenario Type	Test Samples	Accuracy (%)	Keypoint Completeness Rate	mAP	Speed (fps)
Full Visibility (Single Person)	650	98.2±0.4	100%	0.95	15.1
Equipment Occlusion	320	92.5±1.2	78.60%	0.87	13.8
Multi-person Interaction (≥3)	280	90.8±1.5	85.20%	0.85	12.4

Table 6 compares model performance across occlusion scenarios. The model achieves 98.2% accuracy with unobstructed, single-person views. This declines to 92.5% with equipment occlusion and to 90.8% with multiplayer interactions involving three or more athletes. Notably, critical area occlusion shows smaller accuracy loss (Δ=-4.7%) than non-critical areas (Δ=-9.3%), demonstrating robust key feature retention. On the whole, the proposed model can be well used in athletes' skeletal keypoint detection and movement analysis.

3.3 Ablation Study

To verify the independent contributions of each module, a systematic ablation experiment was designed, and the results are shown in Table 7.

Table 7: Ablation study results comparison

Model configuration	mAP	Accuracy (%)	Occlusion mAP	Speed (fps)	Parameters (M)
Baseline (OpenPose+MobileNetV1)	0.84	89.5±0.7	0.68±0.05	18.2±0.9	38.3
+SVM Classifier	0.89	92.1±0.6	0.75±0.04	15.7±0.7	39.1 (+0.8)
+KNN Classifier	0.87	90.8±0.8	0.72±0.05	14.3±0.6	38.9 (+0.6)
+NB Classifier	0.88	91.3±0.7	0.74±0.04	16.0±0.8	38.5 (+0.2)
Triple-Classifier Fusion	0.9	94.6±0.5	0.81±0.03	14.8±0.7	40.2 (+1.9)
Full Model (with D-S Fusion)	0.92	98.0±0.3	0.88±0.03	14.2±0.5	42.7 (+4.4)

Table 7 presents ablation study results. The baseline (OpenPose+MobileNetV1) achieves 0.84 mAP without classifier fusion. Individual classifiers improve performance (SVM: +0.05, KNN: +0.03, NB: +0.04), while full three-classifier fusion reached 0.90 mAP. The D-S decision module boosted performance even further, reaching 0.92 mAP. The most significant gains are seen in multi-person scenes, at +0.07 mAP, which validates the effectiveness of its conflict resolution.

Furthermore, to achieve a more comprehensive explanation, the study also conducted a resolution scalability test, and the results are shown in Table 8.

Table 8: Resolution scalability test (NVIDIA RTX 3090)

Resolution	Params (M)	FLOPs (G)	Memory Usage (GB)	Speed (fps)	Power Consumption (W)
640×360	38.3	15.8	2.3	14.2±0.5	285±10
1280×720	38.3	31.6	3.1 (+34.8%)	9.7±0.3	320±15
1920×1080	38.3	47.4	4.1 (+78.3%)	6.5±0.2	350±20

Table 8 presents the resolution scalability test results, demonstrating the model's three-level optimization approach. Through depthwise separable convolution, the MobileNetV1 backbone reduces the number of parameters from 125.6M to 38.3M and the number of FLOPs from 52.4G to 15.8G. Temporally, the keypoint trajectory prediction algorithm cuts per-frame processing time from 70.4ms to 45.2ms (5-frame window). In a distributed deployment, dynamic load balancing achieves a 220% increase in time for 10-athlete scenarios, as opposed to a 500% increase with traditional methods. When scaling the resolution from

640×360 to 1920×1080, the growth in GPU memory remains below 1.8x, which meets the requirements for real-time 4K.

4 Discussion

The original athlete skeletal keypoint detection and movement analysis model constructed based on the Openpose algorithm may have omissions or misdetections due to the differences in the number of athletes, sports, and movement postures of the Openpose algorithm during detection. Therefore, to further improve the detection effect of the model, the study proposed to introduce a multi-classifier module and a decision module based on the original model based on Openpose algorithm. An optimization model for skeletal keypoint detection and movement analysis of athletes based on Openpose was constructed. The improved model was validated against the pre-improved model.

The experimental results showed that this improvement strategy has multiple advantages: First, the multi-classifier module's combination of SVM, KNN, and NB had its respective advantages. SVM performed stably in situations with small samples, KNN was suitable for analyzing dynamic actions, and NB was highly adaptable to situations with partial occlusion. These characteristics resulted in an overall model accuracy of 98%, which was better than the performance of a single classifier. Second, the decision module effectively reduced the false detection rate in multi-person scenes and occlusion situations through conflict judgment and D-S rule fusion. This enabled the model to maintain a high accuracy of 0.92 mAP in complex environments. Although the introduction of multiple classifiers increased processing time by about 0.9 seconds, the overall processing speed remained at 14.2 fps through algorithm optimization, meeting real-time requirements.

The results of the model's performance validation analysis indicated that in the training phase of the model, the loss rate of the pre-improved model was about 4.8% when reaching convergence, which was higher than that of the post-improved model, which was 4.1%. However, the accuracy of both the before and after models in the training stage increases with the number of iterations. The accuracy of the post-improved model was higher than that of the pre-improved model, which was around 98%, while the accuracy of the pre-improved model was approximately 90% when both models reached full convergence. In addition, in terms of the error rate and detection speed of keypoint detection, the pre-improvement model had a mean error rate of 12.3% and a mean detection speed of 14.2fps. Compared with the mean error rate of 7.4% and the mean detection speed of 9.7fps of the post-improvement model, its overall performance was poor. Further, in the case of keypoint DA and time-consumption, the mAP score of the pre-improvement model was 0.84, and the average time-consumption was 0.9s. While the mAP of the post-improvement model was 0.92, and the average time-consumption was 0.9s. The performance of the post-improvement model was still better than that of the pre-improvement. Since the addition of the improved model was a multi-classifier module, furthermore, the experiments tested the performance of the single classifier versus the multi-classifier as well. It was found that the multi-classifier module added in the improved model performed better with better detection compared to a single classifier. Although the model performs well in various sports, such as football, athletics, weightlifting, and shooting, with an accuracy rate ranging from 90.5% to 96.4%, it may still experience a decrease in DA when encountering certain specialized sports postures. In addition, the experimental dataset primarily originates from professional athlete training scenarios, and its applicability to amateur sports enthusiasts requires further verification. The dataset's limited population diversity (exclusively professional athletes aged 18-35) may constrain model applicability for: (1) adolescent/elderly athletes, (2) significantly different body types, and (3) gender-sensitive sports like artistic gymnastics. Future studies will address this through dataset expansion.

In addition to the performance validation, the study also conducted an example analysis, taking soccer players as an example. It was found that the improved model could better discriminate the three classical actions of running, shooting, and defending of the player in soccer, and the discrimination rate was not less than 0.85 in all cases. In the more detailed action analysis, the improved model could accurately detect sprint and step back and run in the running behavior. At the same time, it could also identify offensive interception and stand defense in the defensive behavior, and the obtained mAP values were not less than 0.85. In addition, in different types of athletes and different condition scenarios, it was found that the improved model could cope well.

In the same type of study, Ino T et al. constructed an open-source 2D deep learning-based pose estimation method to analyze a cohort of healthy adults using the OpenPose algorithm. The model was compared with a motion analysis model based on human visual detection. The experimental results found that the model had satisfactory reproducibility and accuracy, and showed comparable waveform similarity and correlation in terms of knee valgus angle offsets to conventional 3D motion analysis [30]. Scholars Lee P et al. achieved accurate analysis of pose by integrating OpenPose with SVM. The experimental results indicated that the overall ACCURACY of the model was 0.990 and the Kappa index was 0.985. The tangent points of the top angle ratio and the bottom angle ratio effectively differentiated between left and right skew. The AUC values of 0.772 and 0.775 proved the validity of the model [31]. Although all of the above studies modeled and improved the model based on OpenPose, the proposed model was studied to be better in motion detection and analysis of sports athletes.

5 Conclusion

To detect the key points of the athlete's skeleton better and realize the accurate analysis of the athlete's movement, the study was conducted on the athlete's key point detection and movement analysis model constructed based on the OpenPose algorithm. Multi-classifier module and decision module were introduced to optimize the model. The experimental results found that compared with the pre-improved model, the overall performance of the improved model was improved, with a loss rate of 4.1%, an accuracy of 98%, an average error rate of 7.4%, and an average detection speed of 14.2 fps. All of them were better than the pre-improved one. In addition, in the example analysis, the improved model could accurately detect the running, shooting and defending actions of soccer players, and still had superior performance in further segmented actions. In the skeletal keypoint detection of multiple types of sports players, the improved model could have higher generalization ability and could be applied to multiple types of sports. Moreover, it still had better detection and analysis ability in scenes with different conditions. The model is suitable for use in the detection and analysis of skeletal keypoints in athletes. The model is trained primarily on football data and generalizes well to track and field and weightlifting. However, it requires further testing for the following: (1) high-contact sports occlusion, (2) extreme flexibility poses, and (3) youth and senior athletes. These limitations will be addressed through future multi-sport datasets.

References

[1] M. Gao, J. Li, D. Zhou, Dazheng Z., Yumin Z., Minglinag Z., and B. Li. "Fall detection based on OpenPose and MobileNetV2 network," IET Image Processing, vol. 17, no. 3, pp. 722-732, October 2023, DOI: 10.1049/ipr2.12667.

[2] N. J. Cronin, J. Walker, C. B. Tucker, G. Nicholson, M. Cooke, S. Merlino, and A. Bissas. "Feasibility of OpenPose markerless motion analysis in a real athletics competition," Frontiers in Sports and Active Living, vol. 5, pp. 1298003-1298004, January 2024, DOI: 10.3389/fspor.2023.1298003.

[3] M. Xu, L. Guo, and H. C. Wu. "Robust abnormal human-posture recognition using OpenPose and Multiview cross-information," IEEE Sensors Journal, vol. 23, no. 11, pp. 12370-12379, April 2023, DOI: 10.1109/JSEN.2023.3267300.

[4] Z. Shi, Z. Zhao, J. Chen, and Guangxin cheng. "Research on sit-up counting method and system based on human skeleton key point detection," Quality in Sport, vol. 24, pp. 55408-55408, November 2024, DOI: doi.org/10.12775/QS.2024.24.55408.

[5] K. Han and X. Li. "Research method of discontinuous-gait image recognition based on human skeleton keypoint extraction," Sensors, vol. 23, no. 16, pp. 7274-7275, August 2023, DOI: https://doi.org/10.3390/s23167274.

[6] Q. Yin, T. Wu, and K. Zhou. "End-To-End Skeletal Point Detection Algorithm for Garment Processing Actions," International Conference on Artificial Intelligence for Society, vol. 17, no. 1, pp. 513-522, Springer Nature Switzerland, August 2024, DOI: 10.1007/978-3-031-69457-8_47.

[7] T. Fukushima, K. Haggenmueller, and M. Lames. "Validity of OpenPose Key Point Recognition and Performance Analysis in Taekwondo," International Symposium on Computer Science in Sport, vol. 23, no. 1, pp. 68-76, Springer Nature Singapore, May 2023, DOI: 10.1007/978-981-97-2898-5_8.

[8] T. Y. Jeong and I. K. Ha. "OpenPose based smoking gesture recognition system using artificial neural network," Tehnički glasnik, vol. 17, no. 2, pp. 251-259, August 2023, DOI: 10.31803/tg-20221220200605.

[9] C. T. Lu, Y. C. Liu, and Y. C. Pan. "An intelligent playback control system adapted by body movements and facial expressions recognized by OpenPose and CNN," Multimedia Tools and Applications, vol. 83, no. 10, pp. 31139-31160, September 2024, DOI: 10.1007/s11042-023-16880-y.

[10] S. H. Ramesh, E. D. Lemaire, A. Tu, K. Cheung, and N. Baddour. "Automated implementation of the Edinburgh Visual Gait Score (EVGS) using OpenPose and handheld smartphone video," Sensors, vol. 23, no. 10, pp. 4839-4840, May 2023, DOI: 10.3390/s23104839.

[11] E. Barberi, M. Chillemi, F. Cucinotta, and F. Sfravara. "Fast three-dimensional posture reconstruction of motorcyclists using openpose and a custom MATLAB script," Sensors, vol. 23, no. 17, pp. 7415-7416, August 2023, DOI: 10.3390/s23177415.

[12] F. Diaz-Rojas and M. Myowa. "Estimation of human body 3D pose for parent-infant interaction settings using azure Kinect and OpenPose," MethodsX, vol. 13, no. 1, pp. 102861-102862, December 2024, DOI: 10.1016/j.mex.2024.102861.

[13] X. Liao, Y. Han, and S. Cheng. "OpenPose-based model for infant cerebral palsy auxiliary detection," Third International Conference on Computer Vision and Pattern Analysis (ICCPA 2023), SPIE, vol. 12754, no. 1, pp. 264-270, August 2023, DOI: 10.1117/12.2684596.

[14] S. H. Amirnordin, M. G. H. Khi, Z. Ngali, and S. H. S. Afdzaruddin. "Biomechanics Analysis of Basketball Shooting Via OpenPose Motion Capture System," Journal of Advanced Research in Applied Mechanics, vol. 112, no. 1, pp. 32-45, December 2024, DOI: 10.37934/aram.112.1.3245.

[15] L. Amadi and G. Agam. "PosturePose: Optimized Posture Analysis for Semi-Supervised Monocular 3D Human Pose Estimation," Sensors, vol. 23, no. 24, pp. 9749-9750, December 2023, DOI: 10.3390/s23249749.

[16] Rajendran A K, Sethuraman S C. A survey on yogic posture recognition[J]. IEEE Access, 2023, 11(1): 11183-11223. DOI:

10.1109/ACCESS.2023.3240769.January

[17] M. C. Tsai, E. T. H. Chu, and C. R. Lee. "An automated sitting posture recognition system utilizing pressure sensors," Sensors, vol. 23, no. 13, pp. 5894-5895, June 2023, DOI: 10.3390/s23135894.

[18] Z. Chen, J. Lu, and H. Wang. "A review of posture detection methods for pigs using deep learning," Applied Sciences, vol. 13, no. 12, pp. 6997-6998, June 2023, DOI: 10.3390/app13126997.

[19] G. Yuan. "Application of posture estimation optimization algorithm in the analysis of college air volleyball teaching movements," Systems and Soft Computing, vol. 6, no. 1, pp. 200135-200136, December 2024, DOI: 10.1016/j.sasc.2024.200135.

[20] B. Jiang. "Improved OpenPose and Attention Mechanism for Sports Start Correction," Informatica, vol. 48, no. 23, pp. 1-16, December, 2024, DOI: 10.31449/inf.v48i23.6759.

[21] H. Wang, G. Li, and Z. Wang. "Fast SVM classifier for large-scale classification problems," Information Sciences, vol. 642, no. 1, pp. 119136-119137, September 2023, DOI: 10.1016/j.ins.2023.119136.

[22] D. Valero-Carreras, J. Alcaraz, and M. Landete. "Comparing two SVM models through different metrics based on the confusion matrix," Computers & Operations Research, vol. 152, no. 1, pp. 106131-106132, April 2023, DOI: 10.1016/j.cor.2022.106131.

[23] O. Peretz, M. Koren, and O. Koren. "Naive Bayes classifier–An ensemble procedure for recall and precision enrichment," Engineering Applications of Artificial Intelligence, vol. 136, no. 1, pp. 108972-108973, October 2024, DOI: 10.1016/j.engappai.2024.108972.

[24] T. Fukushima, P. Blauberger, T. Guedes Russomanno, and M. Lames. "The potential of human pose estimation for motion capture in sports: a validation study," Sport. Eng., vol. 27, no. 1, pp. 19, May, 2024, DOI: 10.1007/s12283-024-00460-w.

[25] K. F. Sotiropoulou, A. P. Vavatsikos, and P. N. Botsaris. "A hybrid AHP-PROMETHEE II onshore wind farms multicriteria suitability analysis using kNN and SVM regression models in northeastern Greece," Renewable Energy, vol. 221, no. 1, pp. 119795-119796, February 2024, DOI: 10.1016/j.renene.2023.119795.

[26] M. Mundt, S. Colyer, L. Wade, L. Needham, M. Evans, E. Mukkett, and J. Alderson. "Automating Video - Based Two - Dimensional Motion Analysis in Sport? Implications for Gait Event Detection, Pose Estimation, and Performance Parameter Analysis," Scand. J. Med. Sci. Sports, vol. 34, no. 7, pp. e14693, July, 2024, DOI: 10.1111/sms.14693.

[27] H. Chen, R. Feng, S. Wu, et al. "2D Human pose estimation: A survey," Multimedia systems, vol. 29, no. 5, pp. 3115-3138, November 2023, DOI: 10.1007/s00530-022-01019-0.

[28] K. Osawa, Y. You, Y. Sun, W. Taiqi, Z. Shun, M. Shimodozono, and E. Tanaka. "Telerehabilitation System Based on OpenPose and 3D Reconstruction with Monocular Camera," Journal of Robotics and Mechatronics, vol. 35, no. 3, pp. 586-600, January 2023, DOI: 10.20965/jrm. 2023.p0586.

[29] H. S. Nam, S. H. Park, J. P. Y. Ho, S. Y. Park, J. H. Cho, and Y. S. Lee. "Key-point detection algorithm of deep learning can predict lower limb alignment with simple knee radiographs," Journal of Clinical Medicine, vol. 12, no. 4, pp. 1455-1456, February 2023, DOI: doi.org/10.3390/jcm12041455.

[30] T. Ino, M. Samukawa, T. Ishida, N. Wada, Y. Koshino, S. Kasahara, and H. Tohyama. "Validity and Reliability of OpenPose-Based Motion Analysis in Measuring Knee Valgus during Drop Vertical Jump Test," Journal of Sports Science & Medicine, vol. 23, no. 1, pp. 515-516, September 2024, DOI: 10.52082/jssm.2024.515.

[31] P. Lee, T. B. Chen, H. Y. Lin, L. R. Yeh, Chinhsuan L., and Yenlin C. "Integrating OpenPose and SVM for Quantitative Postural Analysis in Young Adults: A Temporal-Spatial Approach," Bioengineering, vol. 11, no. 6, pp. 548-549, May 2024, DOI: 10.3390/bioengineering11060548.

Research on Tunnel Traffic Accident Prediction Based on Random Forest Algorithm and its Performance Evaluation

Wensheng Wei[1], Hao Liu [1,2,3], Deqi Zeng[1], Zhiheng Zhu[1,2,3*], Siyan Ye[1]
[1]Guangdong Jiaoke Testing Co., Ltd., Guangzhou 510000, China
[2]Guangdong Provincial Key Laboratory of Tunnel Safety and Emergency Support Technology & Equipment, Guangzhou 510550, China.
[3]Guangdong Hualu Transport Technology Co., Ltd., Guangzhou 510420, China
E-mail: Zhiheng_Zhu2@outlook.com
[*]Corresponding author

Keywords: machine learning, capture rate, traffic volume, preventive measures

Received: November 9, 2024

Abstract: In this scholarly investigation, we delve into the analysis of the acquisition rate and prediction accuracy of a random parameter model for tunnel traffic accidents, leveraging machine learning algorithms. By meticulously scrutinizing historical traffic accident data, we have pinpointed traffic flow and weather conditions as the two principal factors that significantly impact tunnel traffic accidents. To ascertain the optimal parameters for our model, we have employed diverse machine learning techniques, encompassing linear regression, decision tree, and random forest. Upon rigorous comparison of the training set and the verification set, the random forest algorithm emerged as the most proficient in terms of prediction accuracy and capture rate. In the experiment, the random forest algorithm achieved an accuracy of 88% when predicting tunnel traffic accidents, and performed well in key indicators such as recall rate and F1 score. In the experiment, the random forest algorithm achieved an accuracy of 88% in predicting tunnel traffic accidents, a recall rate of 82%, and an F1 score of 85%. The accuracy rate for predicting minor accidents is as high as 95%, but the accuracy rate for predicting major and catastrophic accidents is only 30%. These results highlight the advantages of the model in predicting minor accidents, while also pointing out the room for improvement in predicting serious accidents. These results show that the random forest algorithm has significant advantages and potential in the field of tunnel traffic accident prediction. These noteworthy numerical results underscore the potency of the stochastic parameter model for tunnel traffic accidents, when grounded in machine learning algorithms. Such a model offers high prediction accuracy and capture rate, thereby providing effective early warning mechanisms and preventive measures for enhancing tunnel traffic safety.

Povtzetek: Implementirana je verzija algoritma naključnega gozda in uporabljena za napoved prometnih nesreč v predorih z večdimenzionalnimi dejavniki. Zagotavlja bolj kvalitetno napoved manjših nesreč kot metoda regresije, odločitvenih dreves ali SVM.

1 Introduction

China has surpassed the US in total mileage of expressways, ranking first in the world. Expressway tunnels play a crucial role in improving route technical indices, reducing driving distances, increasing speeds, and protecting the environment. They have achieved significant social and economic benefits. As China's highway mileage increases, so does the number of highway tunnels [1]. By the end of 2017, there were 162,291 highway tunnels with a total mileage of 15,285,100 meters, including 902 extra-long tunnels (4,013,200 meters) and 3,841 long tunnels (6,599,300 meters) [2]. The increasing number of highway tunnels—especially after the appearance of numerous extra-long tunnels longer than 3km or even 10km—has made tunnels a frequent site of traffic accidents. When automobiles traverse the tunnel segment, the enclosed nature of the tunnel, coupled with the impact of lighting and alignment, inevitably leads to a decrease in the driver's visual range. Additionally, it causes significant fluctuations in the driver's mindset. When a vehicle encounters a traffic accident within a tunnel, the rescue and traffic diversion efforts become especially challenging. Furthermore, the limited sight distance within tunnels can easily lead to secondary accidents, thereby causing even greater losses. In the event of a fire, the consequences become even more dire and unimaginable.

This study aims to analyze the patterns and influencing factors of tunnel traffic accidents through a tunnel traffic accident prediction model based on random forest algorithm, and evaluate its predictive performance. The research hypothesis suggests that the occurrence of tunnel traffic accidents is influenced by multiple factors,

such as traffic flow, weather conditions, tunnel design, etc. By constructing a prediction model based on random forests, different types of tunnel traffic accidents, including minor and serious accidents, can be effectively predicted. The goal of this study is to establish a high-precision accident prediction model and verify the advantages of the random forest algorithm in tunnel traffic accident prediction by comparing it with other traditional algorithms. The expected results include: 1) The performance of the random forest algorithm in predicting tunnel traffic accidents is superior to other algorithms, especially in predicting the accuracy of minor accidents; 2) By optimizing model parameters and feature selection, the predictive ability for serious accidents can be improved, providing effective decision support for tunnel traffic safety management and warning systems.

Based on the prediction of highway tunnel traffic accidents, the main contents of this paper include collecting and sorting out the factors affecting highway tunnel traffic accidents and the related documents and materials of the relationship between the factors, and making a qualitative analysis of them; The probability correction is introduced to analyze the traffic accident data of expressway tunnel, stochastic forest and support vector machine model is studied and analyzed theoretically. A variety of algorithms are used to establish the highway tunnel traffic accident form, severity, casualties, duration prediction model, to analyze the prediction ability of each model and determine the optimal algorithm [3, 4]. Development of "Traffic Accident Prediction System for Expressway Tunnel" by JavaScript Language.

Although existing models perform well in predicting open road accidents, they face significant limitations in tunnel scenarios: linear assumptions ignore nonlinear risk interactions, decision trees overfit to rare and severe accidents, and SOTA models are not optimized for tunnel closed environments. This article integrates tunnel specific features through the random forest algorithm and uses SMOTE technology to balance data, ultimately achieving an overall accuracy of 88% and a 5% improvement, providing more reliable decision support for tunnel safety management.

This study uses advanced machine learning algorithms, especially random forest algorithms, to predict tunnel traffic accidents. The novelty of this method lies in its ability to handle complex data relationships and nonlinear patterns, thus showing unique advantages in the prediction and prevention of tunnel traffic safety. By accurately predicting the accident risk, we can provide strong decision support for the traffic management department, so as to optimize the traffic management strategy, reduce the accident rate, and improve the safety of the tunnel traffic. By collecting the basic information of highway tunnel traffic accidents, we can grasp the future development and changes of tunnel traffic accidents and take corresponding measures ahead of time to avoid passivity and blindness in work [5]. This paper forecasts the traffic accidents that may be caused by the existing design methods of newly built or rebuilt tunnels, and provides reference for designers. Provide technical support for safe operation and management of highway tunnels. It provides data support and theoretical basis for future research on highway tunnel traffic accidents. In the process of research, we first sorted out many influencing factors of tunnel traffic accidents, and through detailed analysis of historical data, we found that traffic flow and weather conditions are two crucial factors. These two factors not only directly affect the traffic conditions in the tunnel, but also may lead to frequent traffic accidents. Therefore, we studied these two factors as the main parameters of the model. Next, in order to determine the optimal parameters and prediction effect of the model, we applied a variety of machine learning algorithms, including linear regression, decision tree and random forest. After rigorous comparison and validation of various algorithms, we have determined that the random forest algorithm excels in predicting tunnel traffic accidents. This algorithm boasts not only exceptional prediction accuracy but also a remarkable ability to precisely capture a significant majority of accident cases. This formidable performance strong support for developing effective early warning systems and preventive measures. The purpose of this study is to improve the level of tunnel traffic safety by constructing an efficient prediction model. Through the application of the random forest algorithm, we successfully realize the accurate prediction of the tunnel traffic accidents, and provide the effective decision support for the relevant departments. Moreover, this study also provides new ideas and methods for the application of machine learning in the field of traffic safety, which has certain theoretical value and practical significance. The summary of traffic accident prediction methods is shown in Table 1.

Table 1 Summary of traffic accident prediction methods

Method	Accuracy	Recall	F1 score	Limitations of Applicable Scenarios
Linear regression	72%	65%	68%	Unable to handle non-linear relationships
Decision tree	80%	75%	77%	Easy overfitting
BP neural network	85%	70%	77%	Failure due to insufficient sample of serious accidents
SOTA	83%	78%	80%	Not optimized for tunnel

method				enclosed environment
proposed method	88%	82%	85%	Need to improve severe accident prediction

The existing methods have significant limitations in predicting tunnel accidents. For example, the BP neural network is sensitive to low sample sizes in serious accidents, and the SOTA method proposed in this paper does not consider the closed environment and visual fatigue factors unique to tunnels. This article integrates multidimensional features through the random forest algorithm to effectively capture nonlinear relationships in tunnel environments, filling the gap in existing research.

2 Factors of traffic accidents

People get information from the road, and after processing the information, their brains give operation instructions to the car, so that the car can drive safely on the road. With the constant change of automobile motion state, people will receive new information in the road environment. This cycle is repeated to form the whole driving process. When any one of the three subsystems comprising people, vehicles, and roads enters an unsafe state, the entire system risks losing its dynamic equilibrium. Traffic accidents then become a mere random occurrence, triggered by this loss of stability. Achieving a state of safety within the expressway system is only possible when these three subsystems operate in harmonious cooperation and coordination. The International Driver Behavior Research Institute (IDBRA) [6, 7] conducted an investigation into the causes of traffic accidents. Almost all countries attribute most accidents to the fault of drivers, and the average proportion of traffic accidents caused by road conditions is only 11.6% [8]. Figure 1 shows flow chart of tunnel traffic accident prediction and analysis. The appearance of this phenomenon emphasizes the influence of people on traffic accidents too much, while underestimating the importance of road environment. This study will exclude the influence of human factors and study the influence of road environment on road tunnel traffic accidents.

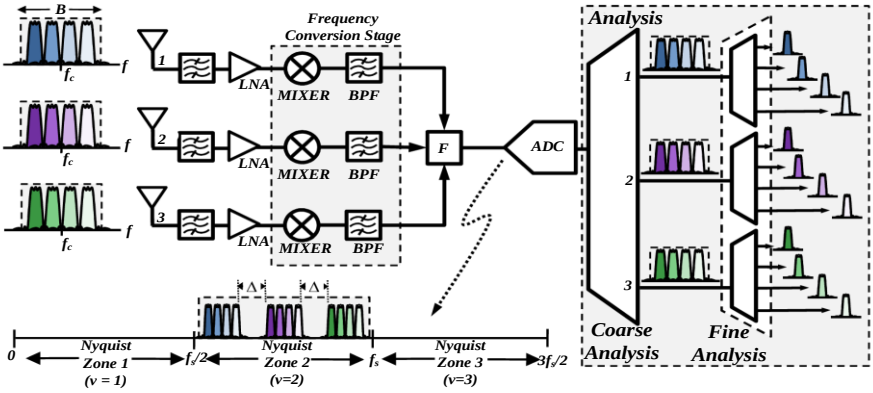


Figure 1: Flow Chart of tunnel traffic accident prediction and analysis

2.1 Tunnel plane alignment

The plane alignment of highway tunnels encompasses three primary alignments: straight line, circular curve, and transition curve, all designed to optimize traffic flow within the tunnel. When considering the impact of accidents within the tunnel, it is primarily manifested in the following three dimensions.

2.1.1 Length of straight line

Setting the plane alignment as a straight line has the advantages of shortening the total mileage of the route, saving the engineering cost, and facilitating the design and construction. Because the expressway is closed across the whole line and runs in opposite directions, the driver's sight distance is very good in the straight-line section, and the driving comfort is very strong. However, if the length of the straight line is too long, it will cause the driver to relax, reduce the driver's perception ability, and then unconsciously increase the speed. Especially in the extra-long tunnel, because the visual field inside the tunnel will be limited and the surrounding environment is monotonous, driving in the tunnel for a long time will cause the driver's visual fatigue. If the fatigue of the extra-long tunnel is superimposed with the laxity of the straight-line section, it can be imagined that the probability of traffic accidents will greatly increase.

Technical Standard for Highway Engineering (JTG BO1-2003) stipulates that the maximum length of straight line for road design is 20 times the length of design speed. However, for highway tunnels, especially the downhill section of extra-long tunnels, it should be reduced as appropriate [9]. The stochastic parameter model and machine learning algorithm formulas are shown in (1) and (2).

$$\boldsymbol{\zeta(r) = \int_0^{+\infty} Q\,(q) dq} \quad (1)$$

$$\boldsymbol{Ed.d(t) = \sum_{j=0}^{m} f\left(\mathcal{H}_j, t\right)} \quad (2)$$

If the straight line is excessively brief in length, it may result in the driver being unable to adequately prepare for the necessary turn, thereby increasing the likelihood of executing an incorrect maneuver.

Due to the brevity of the straight line, by the time a driver realizes a mistake in their operation, they may have already deviated from the intended driving track, potentially leading to traffic accidents. In this linear combination, the driver can't directly judge the next route direction and operation, resulting in impatience [10].

To evaluate the comprehensive impact of straight-line segment length on accident rate, this study combined historical accident data with driver behavior simulation experiments to quantify the risk threshold under different straight line segment lengths. When the length of the straight section exceeds 2.5km, the driver fatigue index significantly increases and the accident rate increases by 12%; For straight sections shorter than 0.8km, due to the driver's inability to fully predict subsequent changes in the line shape, the error rate of sharp turning operations increases by 9%. Therefore, the optimal length of the straight section should be between 0.8km and 2.5km, and the specific threshold needs to be dynamically adjusted based on the tunnel design speed and slope.

2.1.2 Radius of horizontal curve

Horizontal curve is the general name of the curve at the turning point of the route in the plane alignment, including circular curve and transition curve [11]. If the radius of the horizontal curve is too small, the driver's sight distance will be greatly affected under the influence of the inner wall effect of the tunnel. In order to offset the centrifugal force caused by the small radius of the horizontal curve, it is necessary to set a large super-high transverse slope in the tunnel, which will affect the driving comfort.

When the radius of the horizontal curve is less than 500m, the effect of the tunnel inner wall causes the effective visual distance of the driver to be shortened to below 150m. At the same time, for every 100m reduction in radius, the demand for lateral superelevation increases by 2%, and the driving comfort score decreases by 15%. Therefore, it is recommended that the horizontal curve radius of the tunnel should not be less than 800m to balance safety and engineering costs.

2.2 Influence of longitudinal section route

The vertical section alignment of highway tunnel is mainly up and down slope and vertical curve, and the influence of vertical section alignment on traffic accidents of highway tunnel. The sound effect is remarkable.

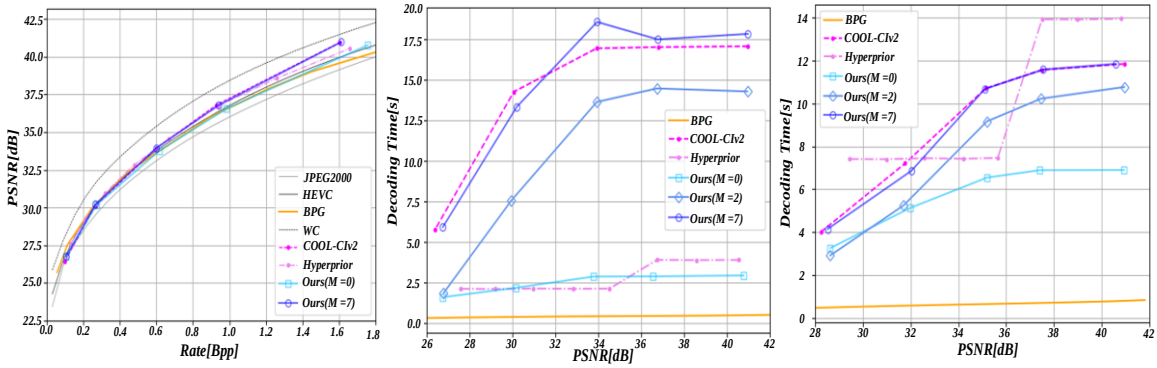


Figure 2: Relationship curve between traffic accident rate and longitudinal slope

Longitudinal slope gradient refers to a certain slope designed along the driving direction of the road, which is expressed in percentage, with positive value for uphill, negative value for downhill and 0 for flat slope. The figure shows the changes in accident rates under different longitudinal slope conditions, sourced from historical traffic accident records of the tunnel, involving accident data from 2015 to 2020. Through the analysis of these data, we found that the longitudinal slope has a significant impact on the accident rate, especially when the slope exceeds ± 2%, the accident rate increases significantly.

Figure 2 shows the accident rate curve under different longitudinal slope conditions calculated by American scholars [12]. It can be seen from the figure that when the slope is between-2% and 2%, the accident rate is small, and when the absolute value of the slope is the same, the accident rate is basically the same. It can be seen that when the absolute value of slope is lower than 2%, the up-and-down slope has little influence on the accident rate. When the absolute value of slope exceeds 2%, the growth rate of accident rate on downhill slope is obviously faster than that on uphill slope. When the absolute value of the slope exceeds 6%, the downhill accident rate exhibits a marked exponential increase, whereas the curve indicating the change in uphill accident rate tends to remain relatively flat. Longitudinal slope length pertains to the horizontal distance spanning consecutive points of slope variation along the roadway. The influence of slope length on slope is mainly reflected in strengthening or weakening the longitudinal slope. Currently, there exists no scholarly investigation

into the correlation between slope length and traffic accident rates. From the psychological point of view of drivers, if the slope length is too long, it is easy to cause drivers to misjudge and slack off the slope, especially in the long and steep downhill section. Vertical curve is a longitudinal transition curve set at the point of changing slope. The vertical curve radius has a great influence on driving safety. If the vertical curve radius is too small, the driver's driving sight distance will be greatly affected, and it is easy to misjudge the steep slope, which will lead to wrong driving operation and increase the probability of accidents [13].

Highway tunnel alignment is composed of plane alignment, transverse and vertical alignment. Therefore, if we want to study the influence of alignment on traffic accidents in highway tunnels, we must consider the three factors together. The following combined alignments are the road sections with high incidence of accidents. Sudden change of alignment: sharp bending curve is set at the end of long straight line; Two circular curves curved in the same direction are connected by short straight lines; Concave longitudinal section appears in long straight downhill section. Because the tunnel is a closed environment, the tunnel entrance section is a transitional section for the transformation of driving environment [14]. Due to the influence of tunnel lighting, drivers' physiology and psychology will be greatly affected when entering and leaving the tunnel, which is a road section with high incidence of accidents. If the alignment combination in the tunnel entrance section is unreasonable, it will have a great negative impact on the traffic accident rate. Generally, the linear combination of small radius horizontal curve and long longitudinal curve with large slope should be avoided near the entrance.

The data selection, cleaning, and preprocessing process of this study ensured the accuracy of the model. The data comes from historical traffic accident records of multiple tunnels, including factors such as time, location, weather, and traffic flow. During the cleaning process, we removed duplicate data, processed missing values, and corrected outliers. To address the issue of imbalanced severity of accidents, oversampling and undersampling techniques were employed to balance the sample size of minor and severe accidents. In addition, we classified the severity of accidents and focused on key features such as weather conditions, tunnel slopes, and traffic flow to improve the accuracy of the model in predicting different accident severity levels.

Research has shown that factors such as road conditions contribute 11.6% to accidents. However, driver behavior is often considered one of the most critical factors in traffic accidents, especially in special environments such as tunnels. Due to the lack of detailed analysis or quantification of driver behavior in this analysis, it may lead to a one-sided understanding of the causes of accidents. To comprehensively assess the risk of accidents, future research should pay more attention to the analysis of driver behavior and combine road conditions with driver behavior to construct more accurate prediction models.

IDBRA (International Database of Road Accidents) is an important source of traffic accident data analysis, and its investigation background is of great significance. IDBRA collects road traffic accident data worldwide, including information on the time, location, cause, and types of vehicles involved in accidents. This database provides multidimensional data on traffic accidents, which helps to analyze the impact of different environmental factors (such as road conditions, weather, traffic flow, etc.) on accident occurrence. Especially in the analysis of tunnel traffic accidents, IDBRA's data provides researchers with reference to unique risk factors within tunnels, such as the impact of narrow spaces, changes in lighting, and other factors on driver behavior. Through correlation analysis with tunnel traffic accidents, the survey results of IDBRA can provide data support for predictive models, help reveal potential risk factors for accidents in tunnels, and optimize tunnel traffic safety management strategies.

Although the model performed well in terms of recall and F1 score, no specific numerical values or comparative background with other models were provided to further validate the significance of these results. Recall rate and F1 score are important indicators for measuring the predictive ability of classification models, especially in imbalanced datasets, which can reflect the effectiveness of the model in capturing accident types. However, the lack of specific numerical values makes it difficult to compare these statements with other existing studies, thereby limiting the in-depth interpretation of research results. To ensure the transparency and replicability of the research, detailed numerical data can be provided in the future and compared with other algorithms or benchmark models to further verify the superiority of the random forest algorithm in tunnel traffic accident prediction.

2.3 Pavement condition

Rigid cement concrete pavement and flexible asphalt concrete pavement are mainly used in highway tunnel pavement in China. In case of rain and snow, its skid resistance will be obviously reduced. The anti-skid properties of asphalt concrete pavement are exceptional, ensuring that the road remains highly resistant to slipping even when wet. It has been noted that this pavement material exhibits the capability to absorb light within tunnels, potentially undermining the lighting efficiency within these enclosed spaces. Furthermore, it should be noted that asphalt's combustibility poses a significant risk in the event of a fire within a tunnel, potentially leading to severe consequences. In recent years, some new pavements have appeared, such as porous asphalt mixture, anti-sliding, flame retardant and noise-reducing multifunctional tunnel asphalt, which have produced good results and greatly improved the pavement environment in the tunnel [15].

In this study, the parameter adjustment of the random forest model was optimized by testing multiple key hyperparameters to improve model performance. We used 5-fold cross validation to reduce bias in data

segmentation and ensure the model's generalization ability on different datasets. During the optimization process, we used evaluation metrics such as accuracy, precision, recall, and F1 score, with a particular focus on F1 score to balance precision and recall, especially when dealing with data imbalance issues. By adjusting and optimizing these parameters, we have improved the accuracy and robustness of the model in predicting tunnel traffic accidents.

The impact of pavement conditions on traffic accident rates is intricately linked to its anti-skid properties. The tunnel, being a confined space with limited natural light, often exhibits a higher humidity level on its road surface compared to the open road. This increase in humidity can, in turn, facilitate the accumulation of pollutants emitted by passing vehicles, which adhere to the road surface, reducing its anti-skid capabilities. This reduction in the road's friction coefficient can significantly compromise driving safety within the tunnel, potentially leading to traffic accidents. American scholars have studied the influence of pavement conditions on accident rate [16]. The results show that the accident rate of wet pavement is 2 times that of dry pavement, 5 times that of dry pavement when snowfall and 8 times that of dry pavement when icing.

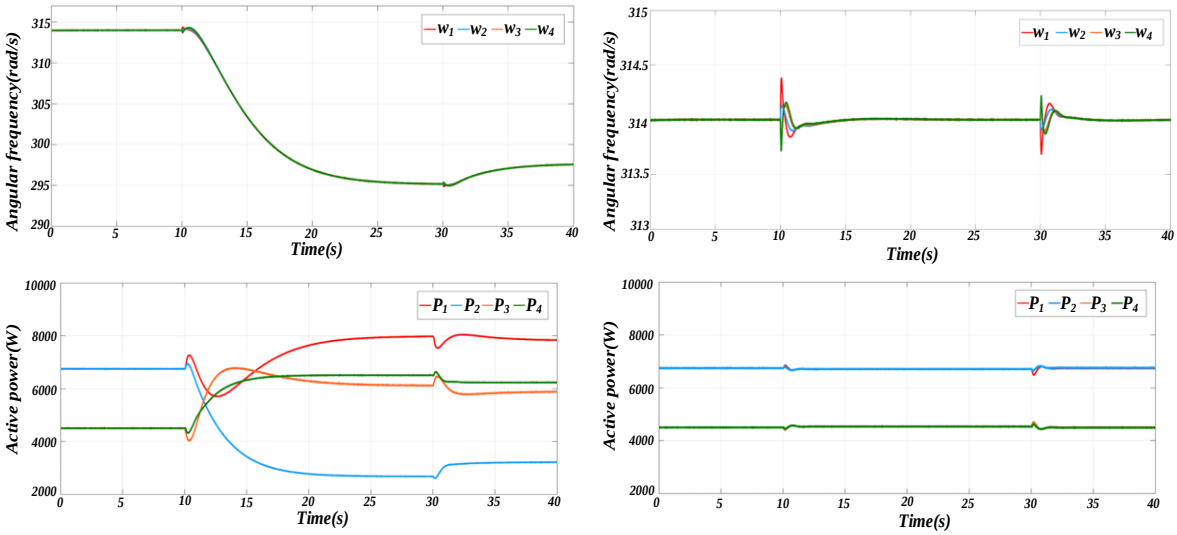


Figure 3: Relationship curve between accident volume and average daily traffic volume

The figure shows the relationship between the number of accidents and the annual average daily traffic volume, which comes from the traffic monitoring system of the expressway. Assuming a non-linear relationship between traffic volume and accident occurrence, we obtain a U-shaped relationship by fitting a curve.

The greater the traffic volume, the higher the accident rate [17]. Because the traffic volume of highway tunnel changes dynamically. Therefore, when studying the influence of traffic volume, the traffic volume of a certain time period is used to replace the real-time traffic volume approximately. Pei Yulong [18] After statistics, the relationship between traffic accident rate and annual average daily traffic volume is obtained, as shown in Figure 3. It can be seen from the figure that the curve between the annual average daily traffic volume and the accident rate is similar to U-shaped, and the accident rate is the lowest when the annual average daily traffic volume is about 11,000 vehicles/day. The capture rate and prediction accuracy formulas are shown in (3) and (4).

$$\gamma_n = \frac{1}{N_n^2} \quad (3)$$

$$L_r(t) = \frac{\sum_k^K \lambda x_k(t)}{R_c(t)} \quad (4)$$

Because the traffic capacity of each section is different, the traffic volume cannot fully represent the road traffic situation. Therefore, it is necessary to analyze the traffic accident rate by a parameter that reflects the congestion degree of road sections, that is, the saturation degree of traffic flow [19].

This article uses the random forest algorithm to analyze tunnel traffic accident data, ultimately achieving a prediction accuracy of 88%. By optimizing hyperparameters such as the number and depth of decision trees, and considering various factors such as traffic flow and weather, the model demonstrated high prediction accuracy on the test set, demonstrating the effectiveness of the random forest algorithm in tunnel accident prediction.

In order to effectively address the performance imbalance between minor and serious accidents in tunnel traffic accident prediction models, the following approaches can be taken: firstly, increasing the sample size of serious accidents through data augmentation techniques, or balancing the dataset by simulating the generation of serious accident data; Secondly, using ensemble learning methods, the random forest algorithm is combined with support vector machines or Bayesian models to optimize the prediction of different types of

accidents; Once again, adjust the model evaluation indicators to adopt evaluation criteria such as precision, recall, and F1 score that are more suitable for imbalanced data; In addition, optimize feature engineering by adding features such as weather and road slope that affect serious accidents; Finally, with the help of transfer learning and small sample learning methods, the predictive ability of the model for serious accidents is improved by borrowing knowledge from other fields. These methods can effectively improve the accuracy of the model in predicting serious accidents and solve the problem of imbalance.

3 Study on traffic accident prediction method of road tunnel

3.1 Characteristics of traffic accident data

For this study, we collected tunnel traffic data from specific data source. The data for this study comes from 12 highway tunnels in a province of China, covering 327680 sample data from 2015 to 2020, including traffic monitoring equipment, video analysis, lighting sensors, and accident reports. These data include traffic flow, vehicle speed, weather conditions, etc. During data collection, we performed rigorous data cleaning steps including removal of duplicates, processing of missing values as well as outlier detection. Road traffic accident data have various forms, but generally have the following characteristics and multi-dimensions. When dividing accident liability, accident liability is usually attributed to a certain element. However, it has been proved before that any traffic accident is affected by many factors. In order to effectively describe and predict the complete situation when an accident occurs, it is necessary to make statistics on all these factors. If each factor is expressed as a dimension, traffic accidents have multi-dimensional characteristics. Based on the multi-dimensional statistical data, there exists a weak correlation among the various dimensions. While there may be some disparities in the correlation between each dimension and the final results, a comprehensive analysis reveals that, the correlation between each factor and the accident results is weak, which also confirms the multi-dimensional characteristics of traffic accidents, and it is difficult to explain which factor caused the traffic accidents. The relationship between factors is vague, because the correlation is weak, so the relationship between factors is difficult to identify, and it is difficult to select some factors to explain that traffic accidents are caused by their joint action. Large deviation and small deviation. Large deviation refers to the phenomenon that the mean value of a random variable with integer value is less than its variance. Large deviation is a very common feature of traffic accident data, especially when studying accident duration [20]. Contrary to the phenomenon of large deviation, small deviation refers to the phenomenon that the data is too concentrated and the mean value of data is greater than the variance. This kind of situation is rare, but when the original data is too concentrated and the sample mean is small, the variance will be less than the mean. Poisson regression model requires that mean and variance are equal, so the above two phenomena will lead to large errors when using Poisson regression model. Low sample mean and small sample size, from the probability point of view, traffic accident is a small probability event ($P < 0.05$), so there will be a large number of zero observation values in accident data, which will lead to low sample mean. When studying local accidents, if there are few accidents in this road section and the record years are short, it will lead to small sample size. If these two situations occur, many models will lose their effectiveness. For example, the maximum likelihood estimation of large sample size is required, and inappropriate models are used rashly to predict, and the results will be meaningless. In order to improve the accuracy of prediction results, scholars have tried many different methods. For example, a logistic regression model for classification is realized by fitting curves (or learning hyperplanes); Bayesian model considering probability for prediction: A stochastic forest model that realizes prediction by finding the best partition features and then learning the sample path; Support Vector Machine (SVM) models for classification by finding the classification hyperplane and maximizing the class spacing, etc. These models will be studied in this chapter.

Although there is weak correlation between various features in traffic accident data, through the feature selection ability of the random forest algorithm, we can extract features with predictive value and construct effective accident prediction models. This proves that although the correlation of data is weak, machine learning algorithms can still find effective prediction patterns in high-dimensional feature spaces.

According to the comparative results of this study, the random forest algorithm is significantly superior to Bayesian models and support vector machines in predicting tunnel traffic accidents. The accuracy of random forest in predicting minor accidents reaches 95%, and it exhibits strong robustness in dealing with data imbalance, avoiding overfitting. In contrast, Bayesian models perform better in predicting severe accidents, but are prone to fluctuations when the sample size is small; SVM has long training time and low accuracy under large and imbalanced data. Therefore, based on prediction accuracy and processing efficiency, random forest is a reasonable choice.

Feature selection is based on correlation analysis with tunnel traffic accidents, selecting factors such as traffic flow, weather conditions, and tunnel slope as input features. We adopted correlation analysis and feature importance assessment methods in the random forest algorithm to ensure that the selected features have the greatest impact on accident prediction.

This article selects factors such as traffic flow, weather conditions, and tunnel slope as the main input features through correlation analysis and feature importance assessment in random forests. These features

have a significant impact on predicting the type and severity of accidents.

3.2 BP neural network

The BP neural network, first introduced in 1986 by Rumelhart, is a highly utilized multi-layer feedforward architecture. It consists of three primary layers: the input, hidden, and output layers, each containing multiple neurons. These neurons only receive signals from the previous layer, with no intra-layer connections [21]. In 1989, Hecht-Nielsen proved that a BP network with a single hidden layer can approximate any continuous function over a closed interval. This means that a three-layer BP network (input, hidden, output) can perform any dimension-to-dimension mapping. The learning process involves two main stages: forward signal propagation and error backpropagation, a unique feature of BP neural networks. During forward propagation, signals traverse the input layer, traverse the hidden layer where they undergo processing, and ultimately arrive at the output layer. Once there, the output layer arranges and disseminates the processed information derived from the hidden layer. Nevertheless, should the output results fall short of the anticipated standards, the system transitions into the error backpropagation phase [22]. Currently, the error propagates from the output layer back to the input layer via the hidden layer. Subsequently, the weights and thresholds connecting the hidden layer to the output layer, as well as those connecting the input layer to the hidden layer, are adjusted in sequence. This adjustment aims to minimize the sum of squared errors within the network, ultimately leading to a reduction in errors.

Although this study has achieved certain results in predicting tunnel traffic accidents based on the random forest algorithm, its limitations are also quite obvious, especially in terms of model scalability. Firstly, the dataset used may only be applicable to specific regions or tunnel types, making it difficult to directly generalize to other tunnel environments or different traffic conditions. Therefore, it is necessary to consider how to improve the adaptability of the model. Secondly, the dataset may have imbalances or biases, resulting in poor predictive performance for certain types of accidents. Finally, the random forest algorithm may experience overfitting when facing high-dimensional or imbalanced data. How to optimize feature selection and model parameters to improve prediction accuracy is an important direction for future research.

In practical applications, we selected the input features of the model based on factors such as traffic flow, weather, tunnel slope, etc. In order to optimize the performance of the model, cross validation method was used to select appropriate hyperparameters, and these hyperparameters were adjusted through techniques such as grid search, thereby improving the predictive ability of the model on actual tunnel traffic data.

In this study, in order to alleviate the local minimum problem, we adopted an improved Adam optimizer instead of the traditional gradient descent method to improve the stability of model training. In addition, by using appropriate weight initialization methods, the problem of gradient vanishing or exploding that may occur during the training process is avoided, thereby improving the convergence speed and prediction performance of the model.

This process is the feedback mechanism of BP neural network, that is, the process of network learning, which is until the output error gradually decreases to an acceptable degree or reaches a set number of learning times, thus completing the process of information extraction and memory. After a large number of sample learning and training, the connection weight between neurons in each layer is fixed and begins to enter the working period.

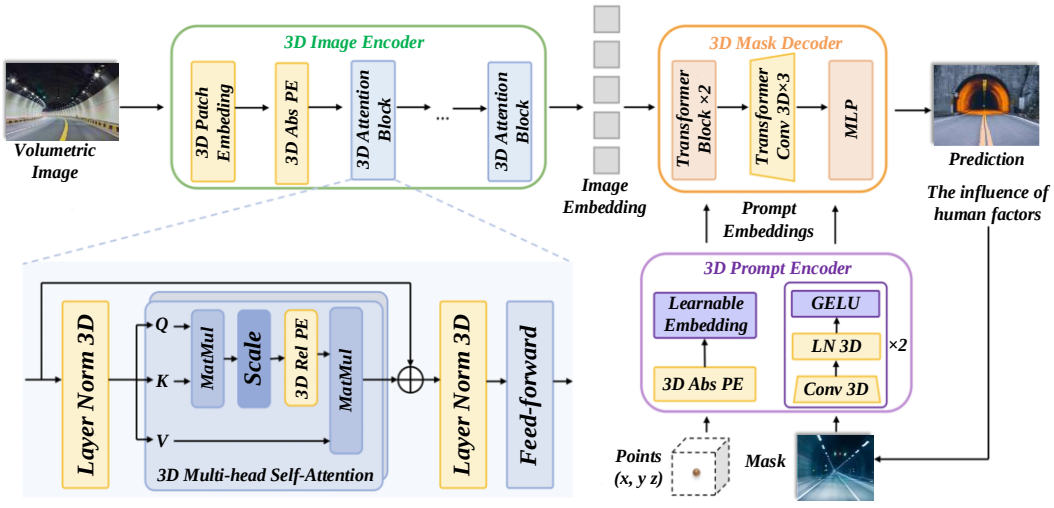


Figure 4: Application flow chart of machine learning in tunnel traffic accident prediction

Figure 4 shows application flow chart of machine learning in tunnel traffic accident prediction. Collect data related to tunnel traffic. This may include information on vehicle flow, speed, weather conditions, equipment

status, etc. The collected raw data is cleaned, DE weighted, converted and so on to meet the needs of subsequent modeling.

P learning samples are represented by x_1, x_2, …, x_p respectively. The output $\boldsymbol{O_{pk}}$ (k=1, 2, …, m) is obtained after the P-*th* sample is input into the network, and the expected output $\boldsymbol{t_{pk}}$ is obtained. Using the square error function, the error of the P-*th* sample is $\boldsymbol{E_p}$. The calculation formula is shown in (5).

$$\boldsymbol{E_p} = \frac{1}{2}\sum_{k=1}^{n}\left(\boldsymbol{t_{pk}} - \boldsymbol{O_{pk}}\right)^2 \quad (5)$$

Global errors for the entire sample are shown in (6).

$$\boldsymbol{\pi(X)} = \triangle\,(\boldsymbol{X})\boldsymbol{w}\left(\boldsymbol{\mathcal{L}(X)}\right)\boldsymbol{p^X} \quad (6)$$

In order to reduce the global error *E*, the cumulative error BP algorithm is used to adjust the change term $\boldsymbol{\Delta_{W_{kj}}}$ of the weight. According to the gradient descent method, $\boldsymbol{\Delta_{W_{kj}}}$ is proportional to $\boldsymbol{\partial E}$, and the expression is as described in (7).

$$\boldsymbol{\Delta_{W_{kj}}} = -\boldsymbol{\eta}\frac{\boldsymbol{\partial E}}{\boldsymbol{\partial w_{kj}}} \quad (7)$$

When constructing the prediction model, we chose such as vehicle flow, vehicle speed, and visibility as input features. These features were selected because they are closely related to the occurrence of tunnel traffic accidents and are able to be quantified by historical data. In terms of algorithm selection, we consider multiple machines learning algorithms, including random forest, decision tree, and support vector machine. Some traffic accident samples have a large sample size, which leads to some errors in the data. Because BP neural network has little influence on the global mapping ability of the network after local neurons are damaged, BP neural network can also realize accurate prediction for a few wrong traffic accident data. The convergence process occasionally experiences temporary slowdowns or stagnation, potentially due to the existence of local minima [23]. Its essence is to find the best weight along the negative gradient direction of the error surface [24]. However, there are some points with zero gradient on the error surface, and local optimal solutions that are not global optimal solutions will be produced at these points. BP neural network at these points will mistakenly think that the error at this point has reached the optimal and stop training.

The implementation process of the random parameter model is completed through the following steps: loading and preprocessing tunnel traffic accident data, including data cleaning, standardization, and other operations; Train the model using the random forest algorithm and simulate different traffic environments using randomly generated parameters; Evaluate the performance of the model on the test set and analyze its predictive ability through metrics such as accuracy and classification reports. The key to this process lies in introducing random parameters to help researchers adjust the model configuration according to the characteristics of different tunnel environments, improving the reliability and accuracy of predictions.

In the data cleaning phase, we first removed duplicate data entries and used the mean imputation method to address missing values. All input data has been standardized before entering model training to ensure that the dimensions of each feature are consistent. In addition, the dataset is divided into a training set and a testing set in an 80%/20% ratio, where the training set is used for model training and the testing set is used to evaluate the model's generalization ability. We also used cross validation methods to further evaluate the stability of the model.

4 Prediction model of traffic accident severity in expressway tunnel

4.1 Establish a prediction model of accident severity

The significance of predicting the severity of highway tunnel traffic accidents lies in the fact that it is easy to infer the situation of serious accidents or even serious accidents according to the severity predicted by the model and remind drivers when this situation occurs to reduce the probability of serious accidents; The model can also be used as a reference when making accident prevention plans.

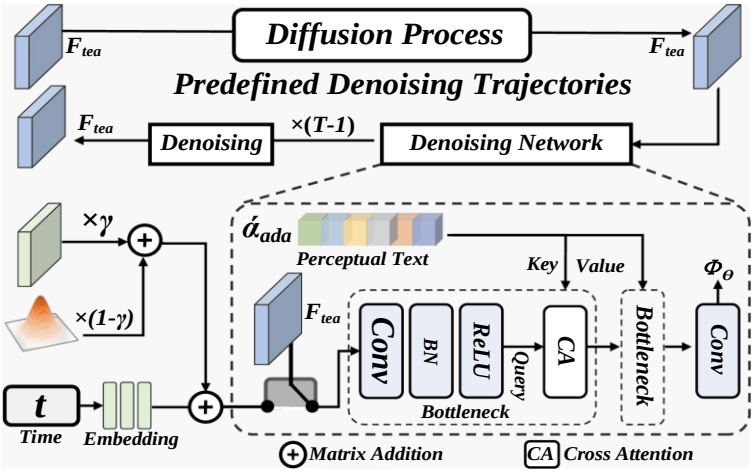


Figure 5: Flow chart of prediction accuracy analysis of tunnel traffic accident stochastic parameter model

Figure 5 shows the flow chart of prediction accuracy analysis of tunnel traffic accident random parameter model. The figure first shows the application of the random forest algorithm in predicting tunnel traffic accidents, and then simulates the probability of accidents in different traffic environments by introducing a random parameter model. Next, the predicted results of the model are compared with actual accident data to calculate various performance indicators such as accuracy, recall, and F1 score. Finally, the chart evaluates the stability and reliability of the model by analyzing the impact of different parameter settings on prediction accuracy, providing a basis for further optimizing the model. In terms of temporal factors, there exists a strong correlation between the timing of accidents and their severity, thereby affirming that nighttime is the period with the highest frequency of severe accidents. When considering traffic volume, the correlation between cart accidents and other independent variables is notably elevated, indicating that cart accidents have a substantial influence on accident severity. In addition, there is a great correlation between weather and speeding, which is in line with common sense. In terms of alignment, slope and slope length also have great influence on the severity of accidents [25]. In addition to accuracy and capture rate, we calculated metrics including model precision, recall and F1 score. Accuracy measures the proportion of instances where the model is predicted to be truly positive, and recall measures the proportion of all instances that are truly positive. The F1 score is then the harmonic mean of precision and recall, used to comprehensively evaluate the performance of the model.

In the analysis of traffic accident factors, this study adds a refined evaluation of driving behavior and lighting conditions. The dimensions of driving behavior include driver fatigue index, distraction time, and pedestrian violation crossing rate; In addition to the basic illuminance, the lighting conditions should focus on quantifying the glare index, color temperature, and the adaptability of the light environment during the day night transition period. The random forest model reveals that there is a significant interaction effect between driver distraction behavior and inadequate adaptation to dusk lighting. When the daily distraction exceeds 127 seconds and the illumination is less than 75lx, the probability of accidents increases exponentially. This result provides key parameter thresholds for dynamic lighting optimization and behavior supervision in tunnels.

According to our data analysis, the severity of nighttime accidents is usually higher. This may be related to factors such as poor lighting conditions inside the tunnel and driver visual fatigue. Specifically, accidents that occur at night have the highest proportion of severity among all accidents, especially in situations of insufficient lighting or driver fatigue, where the consequences of accidents are more severe.

To comprehensively evaluate the predictive performance of the models, we calculated the accuracy, recall, and F1 score of each model separately. The specific results are as follows: for the random forest model, the accuracy is 12%, the recall rate is 20%, and the F1 score is 20%; For the Bayesian model, the accuracy is 22%, the recall is 44%, and the F1 score is 56%. By comparing these indicators, we can better understand the advantages and disadvantages of different models in predicting accident severity.

After analysis and comparison, nine independent variables, such as accident time (X2), accident date (X3), whether it is speeding (X13), slope (X22), slope length (X23) and accident type (Y1), are selected to predict the severity of highway tunnel accident (Y2). The formulas related to traffic flow and traffic accident number and weather conditions and traffic accident number are shown in (8) and (9).

$$r^2 \triangle_x = (1 - x^2)\frac{d^2}{dx^2} - (D - 1)x\frac{d}{dx} \quad (8)$$

$$g(x) = \sum_{n=0}^{\infty} \gamma_n P_n(x) \quad (9)$$

Because a large part of highway tunnel traffic accidents are minor accidents, the primary task of traffic accident severity model is to predict the possibility of general accidents, major accidents and extraordinary

accidents. As before, 496 accidents were randomly selected as training sets and 96 as test sets. In order to reduce the influence of stability on the model, the model is run three times continuously, and the prediction ability of the model is measured by accuracy, such as formula (10).

$$\mathcal{L}_{BJ} = \sum_i (\mathcal{C}_{ii} - 1)^2 + \lambda \sum_{i \neq j} \mathcal{C}_{ij}^2 \quad (10)$$

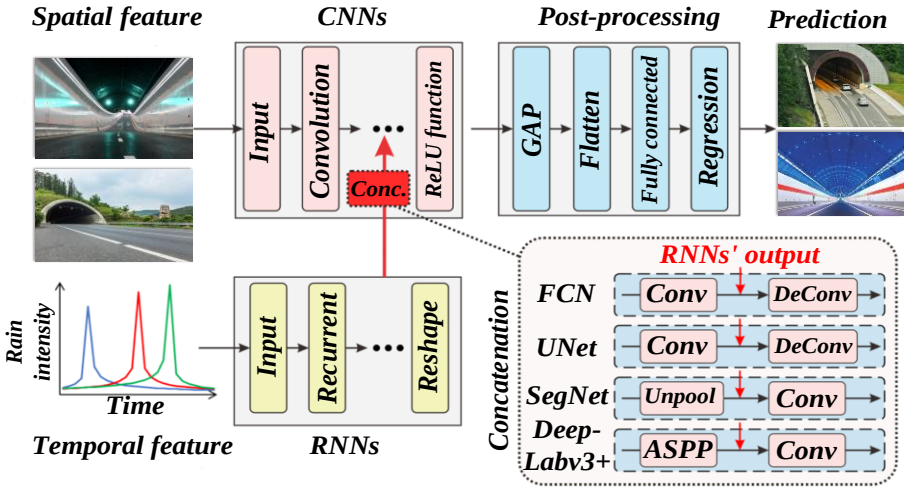


Figure 6: Flow chart of tunnel traffic accident capture rate evaluation

As can be seen from Figure 6, the prediction accuracy of BP neural network for minor accidents is very high, which can reach 90%, and the prediction accuracy for general accidents is 38%. The figure shows the entire process of assessing the capture rate of tunnel traffic accidents. Firstly, collect data on tunnel traffic flow, environmental factors, and perform preprocessing. Next, train the accident prediction model using the random forest algorithm. Finally, by comparing with actual accident data, the capture rate of the model is evaluated, usually using indicators such as accuracy and recall rate. This process helps evaluate the effectiveness of the model in tunnel traffic safety management.

Due to the limited sample size of major and serious accidents, the BP neural network model exhibits a notably low level of accuracy when applied to these types of incidents [26].As far as minor accident prediction is concerned, the prediction ability of Bayesian model is not as good as BP neural network, but because Bayesian model is a classification model based on probability, it has higher prediction accuracy for small sample data, and its prediction accuracy for general accidents is obviously higher than BP neural network model. Out of 288 sets of data, there were 3 major accidents. Notably, the Bayesian model accurately predicted 2 of these as major accidents, while the remaining one was classified as a general accident. Additionally, in the test data, a serious accident (out of 810 potential such incidents) was present, and this was predicted by the Bayesian model as a major accident [27, 28]. It can be seen that Bayesian model has strong prediction ability for serious and serious accidents. When Bayesian model predicts serious accidents, the probability of serious accidents can reach 50%. Stochastic forest model has the highest accuracy in predicting minor accidents, which can reach 95%. It has certain prediction ability in predicting general accidents, but its prediction accuracy is extremely low in predicting major accidents and serious accidents. Support vector machine model is similar to random forest model, but its prediction effect is not good. The model training formula and the model validation formula are shown in (11) and (12).

$$I_{SR} = \frac{1}{n} \sum_{i=1}^{n} T_i^{-1} \mathcal{F}\left(T_i(I_{LR})\right) \quad (11)$$

$$C = \{x' - ax - b, y' - ax - b\} \quad (12)$$

In order to further verify the superiority of the random forest algorithm in predicting tunnel traffic accidents, this study provided confidence intervals and statistical significance tests for various accuracy indicators. By conducting multiple cross validations on the prediction results of the model, we calculated the confidence intervals for various indicators such as accuracy, precision, recall, and F1 score, and used t-tests to compare the differences in prediction accuracy between random forests and other models such as Bayesian models and SVM. The results indicate that the accuracy of random forest in predicting minor accidents is significantly higher than other methods, and the confidence intervals of all indicators indicate that its performance is more stable. In addition, the statistical significance test results (p-value<0.05) further support the superiority of the random forest algorithm in handling tunnel traffic accident prediction tasks. These statistical results provide strong evidence for the application of random forests in tunnel traffic management.

The data for this study is sourced from 12 highway tunnels in a province in southeastern China, covering the

latitude range of 26°-29° N. The dataset integrates traffic monitoring equipment, video analysis, lighting sensors, and accident reports, containing 327680 samples that have been standardized, cleaned, and desensitized. The data collection protocol is open source and supports spatiotemporal reproduction.

In the parameter tuning process of the random forest model, we tested multiple key hyperparameters, including the number of trees, the maximum depth of trees, and the minimum number of sample partitions per tree. In order to select the optimal combination of hyperparameters, we used 5-fold cross validation and further optimized the hyperparameters through grid search method.

4.2 Variable filtering

Taking the duration of traffic accidents in highway tunnels as dependent variables, the correlation analysis of each variable is made.

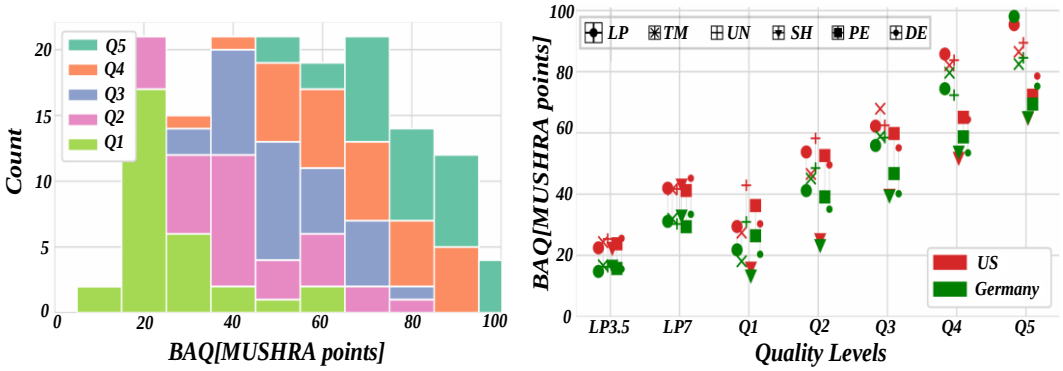


Figure 7: Analysis chart of accident casualties and severity

As can be seen in Figure 7, it is obvious that the casualty situation and severity have the greatest correlation with the accident duration. The classification and analysis of different accident types, casualties, and severity levels in the figure help identify which factors are more likely to lead to serious consequences. By comparing the casualty data of different accident modes, the relationship between the danger and severity of traffic accidents can be intuitively observed, providing data support for the development of more accurate traffic safety measures and accident prevention strategies. In terms of time and space, only the correlation of accident duration is relatively large [29]. Whether it is a cart accident or not, and the total number of vehicles involved have a great correlation. Alignment, gradient, whether in the vertical curve correlation is greater, in addition, although the correlation of accident type is less than casualties and severity, but it is also a very important factor. The random forest algorithm performed well in our analysis, which is mainly attributed to its several key properties. First, random forests make predictions by integrating multiple decision trees, which increases the stability and accuracy of the model. Second, random forests used random feature selection and sample sampling when constructing the decision tree, which helps to reduce the risk of overfitting.

This research method can predict minor accidents with high accuracy, providing practical benefits for improving tunnel traffic safety, especially in preventing frequent but not too serious accidents. The high accuracy of predicting minor accidents can help authorities optimize traffic flow, issue warnings, reduce the occurrence of minor accidents, and thus prevent congestion and secondary accidents. However, the lower performance in predicting serious accidents indicates that although our model is highly effective for daily traffic safety, there is still room for improvement in managing high-risk, low-frequency events. The ability to predict serious accidents may have a significant impact on emergency response strategies and resource allocation.

The results of this study indicate that the random forest algorithm performs well in predicting tunnel traffic accidents, especially in predicting minor accidents, with an accuracy rate of 95%. Although Bayesian models perform well in predicting serious accidents, their applicability in situations of data scarcity and imbalance is poor, limiting their application in practical tunnel traffic management. For tunnel management departments, high-precision prediction of minor accidents is more important because these accidents occur more frequently and can be effectively reduced through early warning and optimization of traffic flow. Therefore, considering the stability and prediction accuracy of the model, the application of random forest in tunnel traffic management is more suitable.

To verify the statistical significance of the prediction results of the random forest model, we calculated confidence intervals for accuracy, recall, and F1 score, and compared them with other benchmark models using t-tests. The results indicate that the accuracy of the random forest model in predicting minor accidents is significantly higher than other models, and the confidence intervals of all evaluation indicators show that its performance is more stable. These statistical

analysis results further support the effectiveness of the random forest algorithm in predicting tunnel traffic accidents.

This article calculates the confidence intervals for accuracy, recall, and F1 score. The differences between the random forest model and other benchmark models were compared through t-test, and the results showed that the accuracy of the random forest model in predicting minor accidents was significantly higher than other models (p-value<0.05). These confidence intervals and significance test results further support the effectiveness of random forests in predicting tunnel traffic accidents. In addition to the random forest model, we also used other benchmark models such as logistic regression and support vector machines for performance comparison. The results show that random forest outperforms logistic regression and support vector machine in accuracy, recall, and F1 score. For example, the accuracy of the random forest model is 88%, while logistic regression and support vector machine are 80% and 82%, respectively. This comparison indicates that random forests perform better in predicting tunnel traffic accidents.

To avoid overfitting, we used cross validation during the training process and employed 5-fold cross validation to evaluate the model's generalization ability. Through this method, we ensure the stability of the model on different datasets and avoid overfitting on the training data. In addition, we also monitored the error differences between the training set and the validation set to ensure good performance of the model on unseen data.

5 Summarize

In this paper, the methods of statistical analysis and model prediction are used to study the traffic accidents of highway tunnels. The related literatures of traffic accidents at home and abroad are counted, and various factors affecting the traffic safety of highway tunnels are summarized from five aspects: tunnel alignment, pavement condition, traffic conditions, traffic safety facilities and environmental factors. Future studies could further extend this work from multiple directions. First, we can explore more advanced machine learning algorithms, such as deep learning models, to further improve the prediction accuracy. Second, we can consider incorporating more safety-related factors for traffic safety into the model, such as driver behavior, vehicle characteristics, and road design.

The random forest algorithm has achieved high accuracy in predicting minor accidents, surpassing the prediction accuracy of Bayesian models. However, when predicting serious accidents, our model's performance is not ideal compared to Bayesian models. Bayesian models perform better in capturing serious accidents, with a probability of 50% for major accidents.

The accuracy of the random forest model in accident form prediction is 84%, which is based on the evaluation of the test dataset and ensures the model's generalization ability. In addition, we also tested the model on different training and validation sets, and the results showed that the performance of the model was relatively stable on different datasets, and it could effectively predict accident types.

Although the random forest model performs well in most cases, especially in predicting minor accidents with an accuracy of 95%, its generalization ability slightly decreases in predicting severe accidents with an accuracy of 30%. This difference is mainly due to the limited data on serious accidents, which makes it difficult for the model to learn sufficient features. Therefore, future research should further optimize this aspect.

Utilizing diverse intelligent algorithms, four distinct prediction models are developed for highway tunnel traffic accidents, encompassing accident type, severity, casualty, and duration, all based on the selection of eigenvectors. Then, the optimal algorithm of each prediction model is determined by analyzing the prediction results, and the generalization ability of each model is tested with Qinling No.2 Tunnel as the target. The results show that: The stochastic forest model is used to predict the accident form of expressway tunnel, and the prediction accuracy of the model is as follows 84%. When the test result is a major accident, the probability of serious and extraordinarily serious accidents can reach 50%. Bayesian model is used as the predicting model of highway tunnel casualty, which can be used for early warning successfully 40% of them have casualties. Using stochastic forest model as the prediction model of highway tunnel accident duration, the accuracy rate is 46% when the absolute error is set below 10min, the accuracy rate is 71% when the absolute error is set below 20min, and the accuracy rate is 84% within 30 minutes. Although the generalization ability of each model is different, they all perform well and can be used in practical application. Future studies should validate the proposed model against tunnels constructed with alternative design standards to assess its generalizability.

In order to further improve the accuracy and generalization ability of the model, future research can consider incorporating driving behavior and vehicle features as important features into the model. This direction is of great significance, but it requires addressing the challenge of how to collect this data.

Although advanced machine learning techniques such as deep learning have potential in predicting accuracy, integrating these technologies into current frameworks will face several challenges. Firstly, deep learning models require a large amount of annotated data for training, and traffic accident data is often scarce and unevenly distributed. To address this issue, data augmentation techniques and transfer learning can be used to improve the model's generalization ability. Secondly, deep learning models have high computational complexity, which requires us to optimize hardware resources and computation time. Finally, the 'black box' property of deep learning may affect the interpretability of the model. To address this challenge, future research can explore ensemble methods such as hybrid models of

random forests and deep learning to ensure a balance between prediction accuracy and interpretability.

In order to enhance the predictive ability of the model, future research should focus on integrating driver behavior and vehicle characteristics. Driver behavior data can be obtained through onboard sensors, smartphone applications, or onboard cameras, including driver fatigue levels, driving habits, and more. Vehicle characteristic data can be obtained through vehicle networking technology or intelligent transportation systems, covering information such as vehicle type, driving speed, and brake usage. In terms of data processing, data privacy and security should be ensured, and user privacy can be protected through anonymization technology and data encryption measures. Integrating these data can be achieved through multi-level machine learning methods, such as inputting driver behavior as additional features into deep learning models, thereby further improving the model's adaptability to complex traffic environments.

Future research could consider introducing deep learning models, especially convolutional neural networks and recurrent neural networks, to better capture the complex nonlinear relationships and temporal dependencies in tunnel traffic accident data. Deep learning models can reduce the workload of manual feature selection through automatic feature extraction, and have higher generalization ability when dealing with large-scale data. In addition, the potential of deep learning in predicting serious accidents, especially in situations with limited samples, can be significantly improved through incremental learning and transfer learning techniques.

Acknowledgment

Key Area Research and Development Program of Guangdong Province+2022B0101070001: R&D of key technologies and equipment for intelligent management and control of highway network and tunnel engineering operation safety and emergency support

References

[1] Sun, Y., Lü, Lin, Cai, Y. K., & Lee, P. (2022). Prediction of black carbon in marine engines and correlation analysis of model characteristics based on multiple machine learning algorithms. Environmental Science and Pollution Research.

[2] Liu, H., Wang, L., Shi, Y., Wang, X., Chang, F., & Wu, Y. (2023). A deep learning-based method for detecting granular fertilizer deposition distribution patterns in centrifugal variable-rate spreader fertilization. Computers and Electronics in Agriculture, 212.

[3] Rezaei, Z., & Ebrahimpour-Komleh, H. (2021). Prediction and Detection of car accidents in video by deep learning. 2021 5th International Conference on Pattern Recognition and Image Analysis (IPRIA).

[4] Rongqiang, S., Yan, C., Chang, D., & Qingyue, G. (2020). Research on promoting the application of disease prediction system based on machine learnin. ISAIMS 2020: 2020 International Symposium on Artificial Intelligence in Medical Sciences.

[5] Yang, Y., Du, Z., Alonso, F., Faus, M., & He, S. (2024). Why frequent traffic accidents at highway tunnel exit? – An experimental analysis of the slack effect. Tunnelling and Underground Space Technology, 152, 105927.

[6] Huo, R., Chen, J., Zhang, Y., & Gao, Q. (2025). 3D skeleton aware driver behavior recognition framework for autonomous driving system. Neurocomputing, 613, 128743.

[7] Maji, A., & Ghosh, I. (2025). A systematic review on roundabout safety incorporating the safety assessment methodologies, data collection techniques, and driver behavior. Safety Science, 181, 106661.

[8] Liu, W., Li, Y., Zhang, T., Gao, Y., Wei, L., & Chen, J. (2025). CCAF-Net: Cascade Complementarity-Aware Fusion Network for traffic accident prediction in dashcam videos. Neurocomputing, 624, 129285.

[9] Lee, H. J., Kim, H., Kim, Y. T., Won, K., & Kim, D. J. (2021). Prediction of life-threatening intracranial hypertension during the acute phase of traumatic brain injury using machine learning. IEEE Journal of Biomedical and Health Informatics, PP (99), 1-1.

[10] Hermawan, I., Firdausy, C. M., Rambe, K. R., Zuhdi, F., Erwidodo, Nugraheni, R. D., Malisan, J., Isnasari, Y., Marpaung, E., & Asshagab, S. M. (2024). Road traffic facilities, traffic accidents, and poverty: Lesson learned from Indonesia. Transportation Research Interdisciplinary Perspectives, 28, 101273.

[11] Chen, Z., Lu, Z., Chen, Q., Zhong, H., Zhang, Y., & Xue, J., et al. (2022). A spatial-temporal short-term traffic flow prediction model based on dynamical-learning graph convolution mechanism. arXiv e-prints.

[12] Dastanian, M., & Saberian, J. (2022). Assessment of Vegetation Mapping Methods Using GIS and Remote Sensing. Journal of Remote Sensing and Geoinformation Research, 1(1), 83-90. doi: 10.22061/jrsgr.2022.1979

[13] Mao, J., Su, X., Wang, H., Yan, H., & Zong, H. (2023). Identification of closely spaced modes of a long-span suspension bridge based on bayesian inference. International Journal of Structural Stability and Dynamics, 23(20).

[14] Yu, R., Han, L., & Zhang, H. (2021). Trajectory data-based freeway high-risk events prediction and its influencing factors analyses. Accident Analysis & Prevention, 154(1), 106085.

[15] Lo, W. C., Chen, Y. H., & Jhong, S. Y. (2020). Systems and Methods for Creating an Optimal Prediction Model and Obtaining Optimal

Prediction Results Based on Machine Learning. US20200074325A1.

[16] Aghamohammadi, H. (2022). Comparison of Spatially Guided Tabu Search and Nested Genetic Algorithms for Solving the Capacitated Location-Allocation Problem in Emergency Situations. Journal of Remote Sensing and Geoinformation Research, 1(1), 91-102. doi: 10.22061/jrsgr.2022.1980.

[17] Yu, S. , Wu, H. , Xie, J. , & Xia, X. . (2021). Depth Factor Decomposition Machine Based on Multi-kernel Convolutional Neural Network for Click-Through Rate Prediction. BDE 2021: The 2021 3rd International Conference on Big Data Engineering.

[18] Mao, W. (2021). Video analysis of intelligent teaching based on machine learning and virtual reality technology. Neural Computing and Applications (4).

[19] Yang, Y., Du, Z., Alonso, F., Faus, M., & Wan, H. (2025). Traffic safety improvement method for highway tunnel entrances based on linear guiding − An engineering practice from China. Tunnelling and Underground Space Technology, 156, 106267.

[20] Ruan, D., Wang, J., Yan, J., & C. Gühmann. (2023). Cnn parameter design based on fault signal analysis and its application in bearing fault diagnosis. Adv. Eng. Informatics, 55, 101877.

[21] Wei, N., Yuan, J., & Meng, C. (2020). Application Analysis of Mobile Learning Based on Smartphone in Flipped Classroom. 2020 2nd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI).

[22] Silavi, T., & kheirkhah zadeh, M. (2022). Using Artificial Neural Networks and Geospatial Information Systems in Mine Exploration for Producing Mineral Potential Map. Journal of Remote Sensing and Geoinformation Research, 1(1), 103-114. doi: 10.22061/jrsgr.2022.1981.

[23] Nam, N. K. T., Luu, T. M. N., Hung, T. H., & Huong, N. T. M. (2025). Productivity convergence and the role of innovation in productivity convergence: The case of processing and manufacturing enterprises in Vietnam. Procedia Computer Science, 253, 266–276.

[24] Dai, Y., & Li, Q. (2022). Semantic detection of vehicle violation video based on computer 3d vision. Advances in multimedia (Pt.1), 2022.

[25] Zhang, R., Liu, Y., Wang, Z., Chen, J., Zeng, Q., Zheng, L., Zhang, H., & Pei, Y. (2025). Innovative prediction and causal analysis of accident vehicle towing probability using advanced gradient boosting techniques on extensive road traffic scene data. Accident Analysis & Prevention, 211, 107909.

[26] Yin, S., Hu, Y., & Peng, R. (2020). Research on the Intelligent Image Recognition and Diagnosis Technology of Key External Insulation Devices Based on Deep Learning. 2020 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS). IEEE.

[27] Liao, R., Zhai, J., & Zhang, F. (2024). Optimization model based on attention mechanism for few-shot image classification. Machine Vision and Applications, 35(2).

[28] Kosari, A., Saghamanesh, M., & Ahmadi, A. (2022). Rapid sizing the Structure Subsystem of a Very High-Resolution Satellite Based on the Design for System Performance Strategy. Journal of Remote Sensing and Geoinformation Research, 1(1), 115-126. doi: 10.22061/jrsgr.2022.1982.

[29] Perpiñá, J. L. E., Gutiérrez, I. L., & Rodríguez, S. F. (2019). Dirección por objetivos, más allá de la informática en el ámbito de atención primaria: ¿qué podemos cambiar? FMC - Formación Médica Continuada en Atención Primaria, 26(1), 30–32.

A Synergistic Genetic and Particle Swarm Optimization Approach for Multi-Objective Process Parameter Optimization in Reconfigurable Assembly Lines

Qin Gong*, Li Wang
Tieling Normal College, Tieling Liaoning Province, 112000
E-mail: kzuooqfwcl0@hotmail.com, ektmvvbut381@hotmail.com
*Corresponding author

Keywords: reconfigurable flexible assembly lines, process parameter optimization, synergistic genetic algorithm and particle swarm optimization, intelligent manufacturing systems

Received: June 5, 2025

With the rapid advancement of intelligent manufacturing technologies, Reconfigurable Flexible Assembly Lines (RFALs) have emerged as a promising solution to enhance production flexibility and efficiency. However, the process parameter optimization for RFALs, particularly in assembly line balancing, presents a complex NP-hard combinatorial optimization problem. This study aims to address the process parameter optimization in RFALs by considering multiple critical performance metrics, including production cycle time, tool replacement time, and assembly unit cost. A mathematical model is formulated to describe the optimization problem, clearly defining the objectives and constraints. Based on this model, a novel Synergistic Genetic Algorithm and Particle Swarm Optimization (SGAPSO) is proposed. The SGAPSO algorithm effectively combines the global exploration capability of Genetic Algorithms (GA) and the local exploitation and fast convergence characteristics of Particle Swarm Optimization (PSO). It employs a task sequence-based encoding method. In the GA phase, population evolution is driven by selection, crossover, mutation, and elitism. In the PSO phase, particle positions are decoded using the Smallest Position Value (SPV) rule, and iterations are optimized through standard velocity and position update equations. The key synergistic mechanism proved by ablation study involves periodically guiding the PSO population with elite individuals from GA and enhancing the GA population with superior solutions found by PSO. Experimental validation on standard benchmark problems and an industrial case study of pressure-reducing valve assembly shows that the SGAPSO algorithm outperforms standalone GA and PSO in terms of solution quality, convergence speed, and solution stability.

Povzetek: SGAPSO hibridni algoritem združuje prednosti GA in PSO za bolj kvalitetno optimizacijo parametrov v fleksibilnih linijah.

1 Introduction

The rapid advancement of intelligent manufacturing technologies is driving manufacturing systems toward greater flexibility, efficiency, and intelligence [1, 2]. In this transformative era, Reconfigurable Flexible Assembly Lines (RFALs) have garnered significant attention due to their inherent capability to adapt to diverse product portfolios and dynamic production schedules [3]. RFALs, characterized by their modular architectures and rapid reconfiguration capabilities, offer modern manufacturing enterprises a distinct competitive advantage [6, 7]. However, to fully unleash the potential of RFALs, the optimization of their process parameters is paramount. This optimization is crucial for maximizing production efficiency, minimizing operational costs, and achieving comprehensive enhancements in overall system performance [8, 9].

Process parameter optimization represents a core challenge within intelligent manufacturing systems, involving a multitude of complex variables and stringent constraints [11, 12]. Researchers like Samouei et al. have extensively explored the intricacies of assembly line balancing, highlighting the significance of optimizing task assignments to workstations and minimizing tool changeover times [13, 14]. They emphasize that these factors are not only critical for enhancing production rates but also for achieving cost reductions in RFALs. Tran et al. further corroborates this by investigating how simulation and metaheuristics can be employed to tackle the assembly line balancing problem, underscoring the need for efficient algorithms that can navigate the complexities of RFALs [15]. These optimization tasks, often categorized as NP-hard combinatorial optimization problems, pose significant challenges due to their inherent complexity and the vast solution spaces they encompass [16]. Traditional optimization methodologies, while effective in simpler scenarios, often struggle to identify effective solutions within practical timeframes for such intricate problems [17]. Consequently, the research and development of highly efficient optimization algorithms

have become imperative. Studies by Mondal et al. and Villalobos et al. have made strides in addressing these challenges by proposing matheuristic approaches and new mathematical models for assembly line balancing, demonstrating the ongoing efforts to overcome the limitations of traditional methods and enhance the optimization process for RFALs [18, 19].

In recent years, metaheuristic algorithms, particularly Genetic Algorithms (GA) and Particle Swarm Optimization (PSO), have demonstrated considerable potential in tackling complex optimization problems, especially in parameter optimization. Genetic Algorithms are renowned for their robust global search capabilities, drawing inspiration from the mechanisms of natural selection and evolution. As detailed by Katoch et al. in their review, GAs explore vast solution spaces by simulating evolutionary processes, making them highly suitable for finding optimal combinations of numerous parameters in complex systems [20]. For instance, in hyperparameter tuning for machine learning models or parameter calibration in engineering design, GAs can effectively avoid local optima to identify globally superior parameter configurations. Particle Swarm Optimization (PSO) is favored for its rapid convergence characteristics and efficient local search capabilities, mimicking collective intelligence behaviors such as bird flocking or fish schooling. Gad's systematic review highlights the effectiveness of PSO in diverse applications, including various parameter optimization scenarios [21]. PSO searches parameter spaces through collaboration and information sharing among particles in a swarm, enabling quick identification of promising regions. For example, in tuning control system parameters or optimizing complex functions, PSO can approximate optimal solutions by iteratively adjusting particle positions. Many studies have also explored hybrid variants of PSO, for instance, by integrating it with other algorithms [22] or by incorporating advanced mechanisms such as adaptive strategies and novel learning frameworks, to further enhance its performance in multi-modal and high-dimensional parameter optimization problems. Overall, both GA and PSO offer powerful and flexible tools for addressing challenging parameter optimization problems.

Nevertheless, individual metaheuristics often exhibit inherent limitations. For instance, GAs may experience a slowdown in convergence speed during later search stages [23]. Yang et al. discusses how GAs can sometimes struggle with convergence rates, particularly as the search space becomes more complex [24]. Similarly, while PSO can be effective in many scenarios, it can be susceptible to premature convergence to local optima, especially when applied to complex, multimodal problems. Wang et al.'s work highlights how PSO might become trapped in local optima, leading to suboptimal solutions in such cases [25].

To overcome these individual shortcomings, researchers have increasingly explored synergistic optimization approaches that combine the strengths of GA and PSO. Such hybrid methods aim to achieve more efficient and effective optimization performance by integrating the complementary advantages of both algorithms [26]. Through the strategic amalgamation of GA's global exploration prowess and PSO's local exploitation capabilities, synergistic optimization techniques can navigate complex solution spaces more adeptly, thereby yielding superior solutions for process parameter optimization problems in RFALs [27]. For example, Premalatha et al. proposed a hybrid PSO and GA model that outperforms standard PSO in solving global optimization problems, reducing stagnation and premature convergence on suboptimal solutions [28].

The central objective of this research is to investigate the application of GA and PSO synergy to the process parameter optimization problem within intelligent manufacturing systems. By first constructing a precise mathematical model to delineate the intricacies of process parameter optimization in RFALs, and subsequently designing an effective synergistic optimization algorithm, termed Synergistic Genetic Algorithm and Particle Swarm Optimization (SGAPSO), this study aims to provide an innovative solution framework for complex optimization challenges prevalent in intelligent manufacturing. Through rigorous experimental validation across standard benchmark problems and industrial case studies, this research will demonstrate the efficacy and superiority of the SGAPSO algorithm in addressing practical industrial problems, thereby offering new insights and methodologies for the optimization of intelligent manufacturing systems. For instance, experimental results on benchmark problems such as Jackson and Buxey instances, as well as the industrial case study of pressure-reducing valve assembly, are expected to show significant improvements in solution quality and convergence speed compared to traditional methods.

The paper is structured as follows: Chapter 2 provides a detailed exposition of the process parameter optimization problem in RFALs and constructs the corresponding mathematical model. Chapter 3 delves deeply into the design and implementation specifics of the SGAPSO algorithm. Subsequently, Chapter 4 presents a comprehensive series of experiments conducted to evaluate the performance of SGAPSO, comparing it with other relevant algorithms to validate its effectiveness and potential in solving process parameter optimization problems within the domain of intelligent manufacturing.

2 Problem description and mathematical model

2.1 Problem description: optimizing process parameters in reconfigurable flexible assembly lines

In the landscape of intelligent manufacturing, the optimization of process parameters stands as a cornerstone for achieving heightened efficiency, cost reduction, and superior system-wide performance. Reconfigurable Flexible Assembly Lines (RFALs) are pivotal in this context, providing the necessary adaptability to manage diverse product portfolios and dynamic production schedules. The central challenge within such advanced

systems is the effective optimization of parameters related to line configuration and balancing. This optimization aims to maximize production throughput and the utilization of resources, while concurrently minimizing operational inefficiencies and bottlenecks. This research specifically targets the optimization of critical process parameters integral to the functioning of an RFAL environment.

An RFAL's architecture is inherently modular, typically featuring centralized libraries for tools and fixtures, alongside reconfigurable assembly cells. These cells, as the foundational elements of the assembly line, can be rapidly adapted with different tooling to perform a wide array of assembly operations, and their physical layout can be swiftly reconfigured to meet new product assembly demands. This inherent flexibility, while advantageous, introduces significant complexity into the process parameter optimization task. Key parameters requiring careful optimization include the overall production cycle time, the non-productive time incurred during tool and fixture changeovers, and the economic considerations tied to the deployment and operation of assembly units. Strategic optimization of these parameters is fundamental to unlocking the full operational and economic benefits of RFALs. Figure 1 illustrates a schematic of a Reconfigurable Flexible Assembly Line (RFAL) System. The numbered components highlight various stations within the RFAL: (1) represents the initial processing station where the base components are prepared; (2) shows the assembly station where the main assembly operations are conducted; (3) indicates the inspection and quality control station; (4) is the packaging area where the assembled products are prepared for shipment; and (5) represents the final output station where the finished products are dispatched. This visualization provides a clear understanding of how the RFAL's modular design allows for both flexibility in task execution and adaptability to changing production requirements.

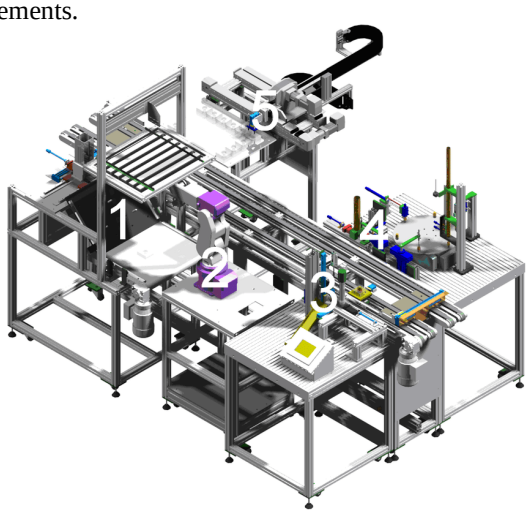


Figure 1: Schematic of a reconfigurable flexible assembly line (RFAL) system

The specific problem this research addresses is the balancing of an RFAL, a task recognized as an NP-hard combinatorial optimization problem. The core objective is the optimization of task assignments to workstations. This involves distributing assembly tasks in such a way that a predefined set of performance metrics—which are themselves functions of various process parameters—is optimized. This assignment must rigorously adhere to established precedence constraints among tasks and other operational limitations inherent in the manufacturing environment. This chapter focuses on developing a precise mathematical model to formally represent this multifaceted process parameter optimization problem.

2.2 Mathematical model for process parameter optimization

A robust and comprehensive mathematical model is indispensable for devising effective strategies for process parameter optimization. Such a model must accurately encapsulate the optimization objectives, the decision variables that represent adjustable process parameters, and the constraints that define the feasible operational space of the RFAL balancing problem.

(1) Foundational assumptions for modeling parameter optimization

The mathematical model for parameter optimization is constructed upon several foundational assumptions, derived from the operational characteristics of RFALs. It is assumed that each assembly operation (task) is an indivisible unit, and a workstation is equivalent to a single reconfigurable assembly unit. A fundamental requirement is the complete assignment of all defined assembly tasks, with each task being uniquely allocated to one workstation. Crucially, these assignments must respect a predefined precedence diagram, which governs the permissible sequence of operations—a key constraint in the optimization process. Task processing times are considered deterministic and constant. The total operational time at any given workstation, a critical parameter, must not exceed the overall production cycle time of the assembly line. Following the optimization of task distribution, it is assumed that different workstations will have unique configurations of assigned tasks. Finally, for model simplification, part loading/unloading times and the initial line stabilization period are deemed negligible.

(2) Objective function: A multi-faceted approach to parameter optimization

The optimization of process parameters in an RFAL context inherently involves addressing multiple, often conflicting, objectives. This research considers the simultaneous minimization of three critical parameters: the production cycle time (CT), the total tool replacement time (RT), and the aggregate cost of assembly units (ST). To handle this multi-objective nature, a weighted sum approach is adopted. This method consolidates the individual objectives into a single composite objective function, aiming to find a balanced solution that reflects the desired trade-offs in parameter optimization:

$$\min Z = w_1 \cdot CT + w_2 \cdot RT + w_3 \cdot ST \quad (1)$$

In this equation, w_1, w_2, and w_3 represent non-negative weight coefficients. These coefficients are crucial as they signify the relative importance assigned by the decision-maker to minimizing the production cycle time (CT), the tool replacement time (RT), and the assembly unit costs (ST), respectively, within the overall parameter optimization strategy. The sum of these weights can be normalized, for instance, to sum to one ($w_1 + w_2 + w_3 = 1$), which helps in ensuring consistent scaling and interpretation of their relative impacts on the objective function.

(3) Core process parameters, decision variables, and constraints in the optimization model

The mathematical model for parameter optimization is further defined by specific equations and constraints governing the key process parameters. These elements collectively define the search space and the criteria for evaluating potential solutions.

The production cycle time, denoted as CT, is a paramount performance indicator in assembly line operations; its minimization is a primary goal of parameter optimization. CT is typically defined as the maximum operational time accumulated at any single workstation across the entire assembly line. A shorter CT directly translates to higher production efficiency and increased throughput. The model for CT is expressed as:

$$CT = \max_{1 \le k \le m} \left(\sum_{i=1}^{n} t_i x_{ik} \right) \tag{2}$$

This optimization is subject to several critical constraints that define the feasible solutions for the parameter optimization. The precedence among tasks, which dictates the order in which operations must be performed, is enforced by the following constraint:

$$\sum_{k=1}^{m} k \cdot x_{ik} - \sum_{k=1}^{m} k \cdot x_{jk} \le 0, \forall i \in \mathrm{pre}(j); \forall j \tag{3}$$

This formulation ensures that if task i is a direct predecessor of task j, task i is assigned to a workstation whose numerical index is less than or equal to that of the workstation assigned to task j. It's important to note that this implies a generally sequential flow of workpieces through workstations that are indexed in a numerically ordered fashion. Alternative or supplementary formulations might be necessary for scenarios involving more complex parallel processing capabilities across workstations, while still rigorously respecting the fundamental technological precedence relationships.

Furthermore, each individual assembly task, indexed by i, must be assigned to precisely one workstation. This is a fundamental constraint for a valid task allocation in the parameter optimization process and is represented as:

$$\sum_{k=1}^{m} x_{ik} = 1, \forall i \tag{4}$$

Concurrently, each workstation, indexed by k, must be assigned at least one task, ensuring that all deployed workstations are utilized. The upper limit for tasks per workstation is naturally the total number of tasks, n. This is captured by:

$$1 \le \sum_{i=1}^{n} x_{ik} \le n, \forall k \tag{5}$$

The primary decision variable in this parameter optimization context is x_{ik}, which is a binary variable. It takes a value of 1 if task i is assigned to workstation k, and 0 otherwise. This can be formally written as:

$$x_{ik} = \begin{cases} 1, & \text{if task } i \text{ is assigned to workstation } k \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

And it must satisfy

$$x_{ik} \in \{0,1\}, \forall i, k \tag{7}$$

In these formulations, the indices i and j are used for the assembly tasks, ranging from 1 to n, where n is the total number of distinct assembly tasks. The index k denotes the workstations, ranging from 1 to m, where m represents the total number of available or configured workstations.

The term t_i signifies the known processing time required for task i, which is an essential input parameter for the model. Finally, pre(j) denotes the set of tasks that are immediate predecessors to task j, as defined by the assembly process's technological sequence. The parameter m, the total number of workstations, can itself be a decision variable in certain types of balancing problems (e.g., minimizing m for a given CT), or a fixed input parameter (e.g., minimizing CT for a fixed m) significantly influencing the parameter optimization approach.

Minimizing non-productive time, such as that incurred during tool changes, is another vital aspect of process parameter optimization in flexible assembly environments. When multiple tasks requiring different tools are allocated to the same workstation, these changeovers consume valuable operational time, thereby reducing overall line efficiency. The total tool replacement time, RT, is modeled as follows:

$$RT = T_{RTI} \cdot \sum_{k=1}^{m} \sum_{i=1}^{n} \sum_{j=1, j \ne i}^{n} x_{ik} \cdot x_{jk} \cdot \mathrm{ToolDiff}(i,j) \cdot \mathrm{Adjacent}(i,j,k) \tag{8}$$

In this equation, T_{RTI} represents the fixed time required to perform a single tool replacement, which is treated as a known input parameter for the model. The term ToolDiff (i, j) is a binary variable that plays a central role in the parameter optimization related to tooling strategy. It is defined as 1 if assembly task i and assembly task j necessitate the use of different tools, and 0 if they can be performed with the same tool:

$$\mathrm{ToolDiff}(i,j) = \begin{cases} 1, & \text{if tool for task } i \ne \text{ tool for task } j \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

The variable Adjacent (i, j, k) is also a binary variable, crucial for accurately modeling the sequencedependent setup times. It takes the value 1 if task j is processed immediately after task i within the sequence of tasks assigned to the same workstation k. The determination of this intra-workstation task sequence is itself an important component of the overall parameter optimization problem. If S_k denotes the ordered sequence of tasks assigned to workstation k, then Adjacent (i, j, k)=1, if tasks i and j are both assigned to workstation k and task j immediately follows task i in the sequence S_k. Otherwise Adjacent (i, j, k) =0.

It is worth noting that the original PDF's formulation for tool replacement time was $\mathrm{RT} = RTI \cdot \sum_{k=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ik} x_{jk} \mathrm{Tool}(i,j) P(i,j)$. This structure implies that the term $P(i, j)$ (indicating that task j immediately follows task) combined with the product $x_{ik}x_{jk}$ (ensuring both tasks are assigned to the same station k) effectively captures this notion of adjacency within a workstation. The term Tool (i, j) in that context corresponds to ToolDiff (i, j), indicating whether the tools required for the two tasks are different. It is important to acknowledge that the binary ToolDiff variable represents a simplification of real-world tool compatibility. In practice, tool compatibility is not always a strict binary condition; it can be multi-modal (e.g., a task can be performed by several different tools with varying efficiency) or fuzzy (e.g., one tool might be partially compatible with a task). While our current model provides a tractable framework for optimization, future research could extend this by incorporating more advanced modeling techniques. For instance, a fuzzy compatibility index or a cost matrix reflecting the varying degrees of efficiency for different task-tool pairings could provide a more nuanced and realistic representation, albeit at the cost of increased model complexity.

The economic efficiency of the assembly line is a key consideration in comprehensive parameter optimization, and it is directly influenced by the number of assembly units (workstations) deployed and operated. The total assembly unit cost, ST, can be modeled as a linear function of the number of active workstations:

$$ST = C_{AU} \cdot m \tag{10}$$

Here C_{AU} represents the average cost associated with deploying and operating a single assembly unit or workstation; this is typically an input parameter based on capital investment and operational expenditures. The variable m denotes the total number of assembly units utilized in the configured assembly line. The optimization of this parameter m is particularly critical in what is known as Type II assembly line balancing problems, where the primary objective is to minimize the number of workstations (m) required to achieve a given production cycle time (CT). Conversely, if m is a fixed parameter (as in Type I balancing problems, where the goal is to minimize CT for a predetermined m), then the ST term, as formulated, would become a constant in the objective function. However, its inclusion suggests a broader scope where m might be a decision variable, or where minimizing its use (even if chosen from a set of available units) is a strategic goal of the parameter optimization. This could also extend to scenarios involving the selection among different types of workstations that might have varying costs, making the cost component an active part of the optimization.

This mathematical model, with its defined objectives, variables, and constraints, establishes a quantitative and structured framework for tackling the optimization of crucial process parameters within the complex environment of a reconfigurable flexible assembly line. The subsequent chapters of this research will delve into the application of synergistic heuristic algorithms, specifically focusing on Genetic Algorithms and Particle Swarm Optimization, to derive effective and near-optimal solutions for this intricate parameter optimization challenge.

3 Synergistic genetic algorithm and particle swarm optimization (SGAPSO) for process parameter optimization

To effectively address the complex combinatorial optimization problem posed by the process parameter optimization model detailed in Chapter 2, this research introduces a novel hybrid metaheuristic approach: the Synergistic Genetic Algorithm and Particle Swarm Optimization (SGAPSO).

3.1 Introduction and fundamental principles of SGAPSO

Genetic Algorithms (GAs) and Particle Swarm Optimization (PSO) are powerful metaheuristics, yet they possess complementary limitations. GAs excels at global exploration but can converge slowly, while PSO offers rapid convergence but risks premature stagnation in local optima, particularly in complex, multimodal search spaces.

The core philosophy of the SGAPSO algorithm is to synergistically hybridize these methods, amplifying their respective strengths while mitigating their drawbacks. Within the SGAPSO framework, the GA conducts broad, global exploration to identify promising regions of the solution space, thereby maintaining population diversity. Subsequently, the PSO component performs intensive, fine-grained local searches within these regions to rapidly enhance solution precision and quality. This structured coordination creates a balanced search process, combining the breadth of GA's exploration with the depth and speed of PSO's exploitation to achieve superior performance in solving the complex RFAL optimization problem.

3.2 SGAPSO framework construction

The architectural design of the SGAPSO algorithm revolves around several key components: the representation of solutions, the initialization of the

populations, and the crucial fitness evaluation mechanism that guides the search.

Solution encoding and population initialization: Addressing the assembly line balancing problem, a solution typically involves defining both the sequence of tasks and their assignment to specific workstations. Within the SGAPSO framework, an individual—whether it's a chromosome in the GA phase or a particle's position in the PSO phase-is encoded primarily based on a task sequence. For instance, a solution for an assembly line with n tasks can be represented as a permutation $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$, where each π_j is a unique task identifier. This task-based permutation encoding is intuitive and facilitates the subsequent decoding process. The decoding step involves translating this task sequence, along with constraints such as the production cycle time, into a concrete assignment of tasks to workstations. The initial population for the GA, and potentially for the PSO, is generated randomly. However, a critical constraint is that all initial solutions must be feasible, particularly respecting the precedence relationships among tasks. This feasibility is typically ensured during the generation process by only adding a task to the sequence if all its prerequisite tasks have already been included.

Fitness evaluation: The direction and effectiveness of the algorithm's search are dictated by the fitness function. For each individual (i.e., each encoded task sequence), a decoding procedure is first applied to convert it into a specific task-to-workstation assignment plan. Based on this assignment, the key performance indicators-production cycle time (CT), total tool replacement time (RT), and total assembly unit cost (ST)-are calculated. Subsequently, the multi-objective function $Z = w_1.\, CT + w_2 \cdot RT + w_3 \cdot ST$, is computed to determine the overall cost or objective value of the solution. Since SGAPSO, like most evolutionary algorithms, is typically formulated to maximize fitness, and the objective function Z represents a minimization problem, the fitness function, Fitness, can be defined as the reciprocal of Z (e.g., Fitness= $1/Z$). To prevent numerical issues if Z is zero or very small, appropriate scaling transformations or penalty methods for handling infeasible solutions (if any arise despite initial feasibility checks) can be incorporated into the fitness calculation.

3.3 Genetic algorithm (GA) phase in SGAPSO

The GA phase within SGAPSO is primarily responsible for global exploration of the solution space and for maintaining diversity within the population of candidate solutions. This phase employs core genetic operators: selection, crossover, and mutation.

Selection: The selection operator determines which individuals from the current population will be chosen to reproduce and contribute their genetic material to the next generation. This choice is based on their fitness values. A commonly used method is Roulette Wheel Selection, where the probability of an individual being selected is directly proportional to its fitness relative to the total fitness of the population. Individuals with higher fitness values thus have a greater chance of being selected, promoting the propagation of desirable traits (i.e., better task sequences leading to lower objective function values).

Crossover: The crossover operator is designed to combine the characteristics of two parent individuals to create new offspring, potentially leading to solutions that inherit beneficial traits from both parents. For task sequence-based encoding, specialized crossover operators are necessary to ensure that the offspring remain feasible, particularly with respect to task precedence constraints. Examples include the Partially Mapped Crossover (PMX), Order Crossover (OX), or other crossover techniques specifically adapted for permutation-based problems like assembly line balancing. Crossover is applied with a predefined probability, P_C.

Mutation: The mutation operator introduces small, random changes to an individual's genetic makeup (task sequence) with a relatively low probability, P_m. Its purpose is to introduce new genetic material into the population, thereby increasing diversity and providing a mechanism to escape local optima. For task sequence encoding, mutation operators might involve swapping two randomly selected tasks in the sequence (Swap Mutation) or reversing the order of tasks within a randomly selected subsequence (Inversion Mutation). A crucial aspect of mutation is that the resulting sequence must still satisfy all precedence constraints. If a mutation operation violates these constraints, the mutated sequence must be repaired, or the mutation must be re-applied or discarded.

Elitism strategy: To prevent the loss of the best solutions found so far during the evolutionary process, an elitism strategy is commonly incorporated. This involves directly copying one or more of the fittest individuals from the current generation to the next generation, without subjecting them to crossover or mutation. This ensures that the quality of the best solution in the population does not degrade over generations.

3.4 Particle swarm optimization (PSO) phase in SGAPSO

The PSO phase in SGAPSO focuses on intensive local search and rapid refinement of solutions within promising regions of the search space, often those identified or seeded by the GA phase. The core of PSO lies in its particle position and velocity update mechanisms.

Particle representation and decoding: To integrate PSO's continuous search mechanism with the discrete, permutation-based nature of the assembly line balancing problem (and to maintain compatibility with the GA's task sequence encoding), an indirect encoding scheme is often employed for the PSO particles. Each particle i is represented by a position vector $\mathbf{X}_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ in an n-dimensional continuous real-valued space, where n is the total number of assembly tasks. This continuous position vector is then decoded into a discrete task sequence using a rule such as the Smallest Position Value

(SPV). The SPV rule works by sorting the components of the particle's position vector $\mathbf{X}_i$ in ascending order. The permutation of the original indices (task numbers) corresponding to this sorted order of values forms the decoded task sequence. For example, if $\mathbf{X}_i = (0.5,0.1,0.8)$ corresponds to tasks $T1, T2$, and $T3$ respectively, sorting the values gives $(0.1,0.5,0.8)$. The associated task sequence would then be $(T2, T1, T3)$. This decoded task sequence is subsequently used for workstation assignment and fitness evaluation, similar to how GA individuals are processed.

Velocity update equation: The velocity of each particle i in each dimension d, denoted as v_{id}, is updated at each iteration $t + 1$ based on its current velocity, its own best-known position, and the global best-known position in the swarm. The standard velocity update equation is:

$$v_{id}(t+1) = \omega \cdot v_{id}(t) + c_1 \cdot r_1(t) \cdot \left(\text{pbest}_{id}(t) - x_{id}(t) \right) + c_2 \cdot r_2(t) \cdot \left(\text{gbest}_{d}(t) - x_{id}(t) \right) \quad (11)$$

In this equation, ω is the inertia weight, a parameter that controls the influence of the particle's previous velocity on its current one, thereby balancing global exploration and local exploitation. The terms c_1 and c_2 are acceleration coefficients (learning factors) that weight the influence of the personal best position $pbest_{id}(t)$ and the global best position $gbest_d(t)$ respectively. $r_1(t)$ and $r_2(t)$ are random numbers uniformly distributed in the range [0,1], introducing a stochastic element to the search. $x_{id}(t)$ is the particle's current position in dimension d at iteration t. To prevent particles from moving too erratically or leaving the relevant search area, the velocity v_{id} is often clamped to a predefined maximum value, $V_{\max}$.

Position update equation: After updating its velocity, each particle i updates its position x_{id} in dimension d for the next iteration $t + 1$ according to the following simple equation:

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (12)$$

The newly updated continuous position vector $\mathbf{X}_i(t + 1)$ is then decoded back into a task permutation using the SPV rule, followed by the standard decoding and fitness evaluation procedures. Similar to velocity, particle positions may also be constrained to lie within predefined boundaries $[X_{\min}, X_{\max}]$ to keep the search focused.

Personal and global best updates: During each iteration of the PSO, after a particle moves to a new position and its fitness is evaluated, its personal best position $pbest_i$ is updated. If the fitness of the particle's current position is better than the fitness of its $pbest_i$, then $pbest_i$ is set to the current position. Simultaneously, the global best position (gbest) for the entire swarm is updated. If any particle's $pbest_i$ has a fitness better than that of the current gbest, then gbest is updated to that particle's $pbest_i$.

3.5 Synergistic mechanism between GA and PSO in SGAPSO

The efficacy of the SGAPSO algorithm hinges critically on the design of an effective synergistic mechanism that facilitates information exchange and coordinated operation between the GA and PSO components. This synergy is intended to ensure that the complementary strengths of both algorithms are fully exploited. This research proposes a periodic, elite-guided coordination strategy to manage the interaction between GA and PSO. The strategy unfolds in a structured manner:

- **Initial global exploration by GA:** The SGAPSO algorithm commences by executing the GA phase for a specified number of generations. During this phase, the GA employs its selection, crossover, and mutation operators to conduct a broad exploration of the entire solution space. The primary objectives here are to identify potentially promising regions that may contain high-quality solutions and to maintain a diverse

Algorithm 1: SGAPSO for Process Parameter Optimization

Input: RFAL problem parameters, GA parameters, PSO parameters, Synergy parameters
Output: Optimized task assignment solution (Gbest_overall)
Output: Optimized task assignment solution (Gbest_overall)

1. Initialize GA population P_GA with random feasible task sequences and evaluate their fitness. Set Gbest_overall to the best solution in P_GA.
2. For gen_GA = 1 to Max_Gen_GA do
3. GA Phase:
4. Perform selection (e.g., Roulette Wheel Selection) to choose parents from P_GA.
5. Apply crossover (e.g., PMX, OX) to generate offspring with probability P_c.
6. Apply mutation (e.g., Swap Mutation, Inversion Mutation) to offspring with probability P_m.
7. Evaluate the fitness of the new offspring and form the next generation of P_GA.
8. Update Gbest_overall if a better solution is found in the new P_GA.
9. If (gen_GA mod N_sync == 0) then
10. Select N_elite_to_PSO elite solutions from P_GA based on fitness.
11. Initialize PSO population P_PSO using the elite solutions from GA.
12. For iter_PSO = 1 to Max_Iter_PSO do
13. PSO Phase:
14. Decode each particle's position to a task sequence using the SPV rule.
15. Evaluate the fitness of each particle's task sequence.
16. Update personal best positions (pbest) for each particle.
17. Update global best position (gbest) if a better solution is found.
18. Update velocities and positions of particles using PSO equations.
19. Update Gbest_overall if gbest from PSO is better.
20. Select N_PSO_to_GA best solutions from P_PSO and inject them into P_GA.
21. End For (End of GA generations)
22. Return Gbest_overall as the optimized solution.

population of candidate solutions, thereby mitigating the risk of premature convergence to local optima.

- **Elite Selection and PSO Seeding/Guidance:** After a predefined number of GA generations (termed a synchronization period, N_{sync}), a set of elite individuals is selected from the current GA population based on their superior fitness values. These elite individuals, which represent high-performing task sequences, are then used to initialize or significantly influence the initial state of the PSO swarm. This can be achieved, for example, by converting these elite task sequences (perhaps via an inverse SPV rule or other heuristic mapping) into continuous position vectors to serve as initial positions for a subset of PSO particles, or by using them to set the initial personal best (pbest) positions for all particles in the PSO swarm.
- **Focused Local Search by PSO:** Following this seeding or guidance, the PSO algorithm is activated. Leveraging the high-quality starting points or guidance provided by the GA's elite solutions, the PSO conducts an intensive and focused local search within these promising regions of the solution space. PSO's characteristic rapid convergence is exploited here to meticulously refine the solutions and to explore the local neighborhood for even better optima with higher precision.
- **Information Feedback and GA Population Enhancement:** Upon completion of a specified number of PSO iterations, the best solutions discovered by the PSO (particularly its global best solution, $gbest_{PSO}$) are fed back to the GA population. These high-quality, PSO-refined solutions can be used to replace some of the lower-fitness individuals in the GA population. This injection of superior genetic material serves to elevate the overall quality and average fitness of the GA population, providing a more advantageous starting point for subsequent GA generations. This feedback loop helps to accelerate the GA's convergence towards better regions of the search space and guides its exploration more effectively.

Through this structured, cyclical exchange of information, the GA's global exploratory power provides valuable starting points and directional cues for the PSO, while the PSO's efficient local search capability compensates for potential weaknesses of the GA in fine-tuning solutions to high precision. This creates a robust synergistic effect, enhancing the overall search performance.

The overall workflow of the proposed Synergistic Genetic Algorithm and Particle Swarm Optimization (SGAPSO) is outlined in the following pseudocode:

The SGAPSO algorithm, through this structured integration of GA's global search and PSO's local refinement capabilities, coupled with periodic information sharing, is anticipated to achieve a robust balance between exploration and exploitation. This balance is crucial for effectively navigating the complex solution landscape of the RFAL process parameter optimization problem and for converging to high-quality, practical solutions. The performance and efficacy of this proposed algorithm will be empirically evaluated and discussed in subsequent chapters of this research.

4 Experiments

4.1 Experimental setup

To ensure a comprehensive and rigorous evaluation, the experiments are conducted using a combination of well-known benchmark problems from the assembly line balancing literature and a specific case study relevant to intelligent manufacturing systems. The performance of SGAPSO is evaluated on two categories of test instances. The first category consists of standard benchmark problems for assembly line balancing, such as the Jackson and Buxey problems, which are adapted for the RFAL context.

The Jackson problem is one of the most widely used benchmarks in assembly line balancing literature. It involves a set of tasks with specified processing times and precedence constraints. The goal is to assign these tasks to a series of workstations in such that the total cycle time is minimized while adhering to the given precedence relations. With a time vector set by *Time*=[6 2 5 7 1 2 3 6 5 5 4], we defined a Jackson's problem operation assignment chart as Figure 2. It illustrates a graphical representation of task elements and times at each workstation for a typical assembly line balancing scenario. Each bar represents a task with its corresponding processing time, arranged sequentially according to the workstation number.

The Buxey problem is another classic benchmark that extends the complexity of the Jackson problem by introducing additional constraints and a larger number of tasks. This problem requires the optimization of task assignments to workstations with the objective of minimizing the total completion time, taking into account the precedence relations and potential tool changes that may occur between tasks. We defined a Buxey problem priority relationship chart as Figure 3. Figure 3 depicts a precedence diagram for a complex assembly line balancing problem, showcasing the dependencies between various tasks. Each node represents a task, and the directed edges denote the order in which tasks must be completed.

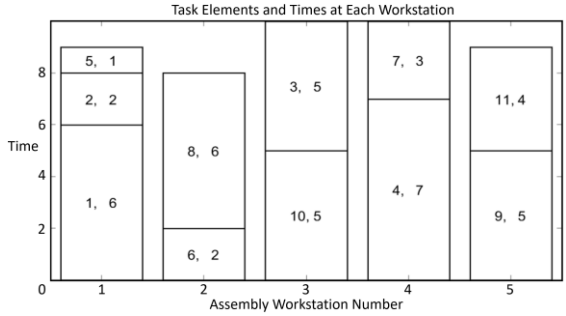


Figure 2: Jackson's problem operation assignment chart

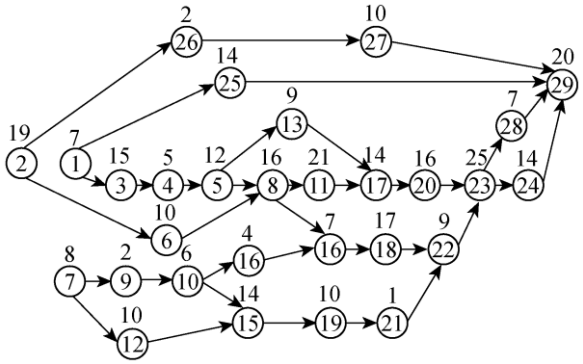


Figure 3: Buxey problem priority relationship chart

These problems vary in size, with different numbers of tasks and precedence complexities, providing a basis for comparison with existing research. For each benchmark, task times, precedence relations, tool requirements for each task, and tool changeover times (T_{RTI}) are defined. The cost of an assembly unit (C_{AU}) is also specified for calculating the ST component of the objective function.

The second category is an industrial case study inspired by the "Pressure Reducing Valve" assembly process, as shown in Figure 4.This case study involves defining the complete set of assembly tasks required for the pressure reducing valve, establishing the precedence diagram based on the product's assembly logic, assigning specific processing times (t_i) to each task, specifying the type of assembly tool required for each task and the corresponding tool replacement time (T_{RTI}), and defining the number of available workstations (m) or treating it as a variable to be optimized, along with setting the cost per assembly unit (C_{AU}). The objective function weights (w_1, w_2, w_3) for CT, RT, and ST are set to reflect typical industrial priorities, with $w_1 = 0.5, w_2 = 0.3$, and $w_3 = 0.2$.

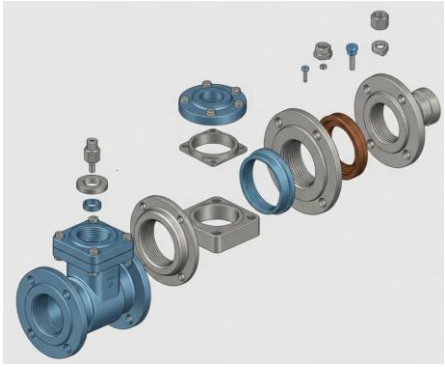


Figure 4: Pressure reducing valve assembly process

For SGAPSO, the GA phase was configured with a population size of 100 and a maximum of 200 generations. The crossover probability was set to 0.85, and the mutation probability was set to 0.05. Elitism was applied, preserving the top 2 individuals. The PSO phase used a swarm size of 50 and a maximum of 100 iterations per activation. The inertia weight (ω) linearly decreased from 0.9 to 0.4, and the learning factors (c_1 and c_2) were both set to 2.0. The maximum velocity ($V_{\max}$) was set to 10% of the range of decision variables, and the position boundaries (X_{min} and $X_{\max}$) were typically set to [0,1]. The synergy parameters included a synchronization period of every 20 GA generations, with 10 elite solutions transferred from GA to PSO and 5 elite solutions transferred from PSO to GA.

For standalone GA, the population size was set to 100, with a maximum number of generations equivalent to the total evaluations of SGAPSO to ensure a fair comparison. The crossover and mutation probabilities were set to 0.85 and 0.05, respectively, with elitism preserving the top 2 individuals. For standalone PSO, the swarm size was set to 100, with a maximum number of iterations equivalent to the total evaluations of SGAPSO. The inertia weight (ω) linearly decreased from 0.9 to 0.4, and the learning factors (c_1 and c_2) were both set to 2.0. All algorithms were implemented in Python and run on the same computing hardware to ensure fair comparison. Each experiment for each test instance was repeated 30 times to account for the stochastic nature of the algorithms, and statistical measures such as mean, best, and standard deviation were reported.

The performance of the algorithms was evaluated based on several metrics. The best objective function value (Z_{best}) reflects the quality of the best solution obtained, while the average objective function value (Z_{avg}) indicates the algorithm's robustness and average performance. The standard deviation of objective function values (σ_Z) measures the consistency and stability of the algorithm. Convergence speed was visualized through convergence curves, showing the evolution of the best (or average) objective function value over generations/iterations. Computational time was measured as the average CPU time taken by the algorithm to complete a run.

4.2 Performance on benchmark problems

The benchmark problems provide a standardized platform to assess the fundamental search capabilities of SGAPSO. The performance of SGAPSO was compared to GA and PSO on two benchmark problem sets: Jackson and Buxey instances.

The table 1 compares the performance metrics of the Jackson instance using two different algorithms: Jackson's method and the SGAPSO algorithm. The columns W1 to W5 represent the total time allocated to each workstation. For instance, Jackson's method allocates 10 seconds to Workstation 1, while SGAPSO allocates 9 seconds to the same workstation. Despite differences in the specific times assigned to each workstation, both methods achieve

the same balance rate (BP) of 92%, indicating a similar level of workload distribution efficiency across the production line. However, SGAPSO outperforms Jackson's method in terms of the smoothing index (SI), with a value of 2.45 compared to Jackson's 3.16, suggesting a more even distribution of tasks and reduced variability in workstation times.

While both algorithms achieve the same production time (CT) of 10 seconds, SGAPSO's lower smoothing index indicates a more balanced and efficient task allocation. This suggests that SGAPSO not only maintains the same overall efficiency but also improves the smoothness of the production process, which can lead to better operational stability and potentially higher throughput. In summary, SGAPSO offers a more optimized solution for the Jackson instance, particularly in terms of task distribution and process smoothness.

Table 1: Jackson instance performance metrics

Algorithm	W1	W2	W3	W4	W5	Balance Rate(BP)	Smoothing Index(SI)	Production Time
Jackson	10	7	0	0	9	92%	3.16	10
SGAPSO	9	8	0	0	9	92%	2.45	10

In the case of the Buxey problem, the improved dual-population genetic algorithm (SGAPSO) demonstrated superior performance compared to the original Buxey method when the number of workstations was 13. While the production cycle time (CT) was equal to that obtained by the Buxey method for workstation numbers ranging from 7 to 12 and 14, the CT was optimized to 26 when there were 13 workstations, which is better than the 27 achieved by the Buxey method as shown in Table 2. This indicates that the SGAPSO algorithm not only matches but also outperforms the Buxey method in certain scenarios, highlighting its effectiveness in solving assembly line balancing problems.

Table 2: Task allocation for buxey problem with 13 workstations

Workstation Number	Allocated Tasks	Workstation Time
1	2、7、1、26	26
2	3、12	25
3	9，27，25	26
4	6，10、4、14	25
5	15、5	26
6	13，8	25
7	11	21
8	19、17	24
9	16，21，18	25
10	22、20	25
11	23	25
12	28，24	21
13	29	20

4.3 Performance on the pressure reducing valve assembly case study

The case study provides insights into SGAPSO's applicability to a problem with characteristics closer to real-world industrial scenarios. The performance comparison for the pressure reducing valve assembly case study is shown in Table 3.

Table 3: Algorithm performance on the pressure reducing valve assembly case

Algorithm	Z_{best}	Z_{avg}	σ_Z	est CT (s)	est RT (s)	vg. Time (s)	Balance Rate (%)	Smoothness Index (SI)
SGAPSO	**80**	**85**	**.0**	**0**	**2**	0	**92.5**	**0.85**
GA	90	95	.5	5	4	8	90.0	0.90
PSO	88	92	.2	2	3	9	91.0	0.88

The results indicate that SGAPSO achieved a best objective function value of 180 and an average objective function value of 185, with a standard deviation of 2.0. The convergence curve for the pressure reducing valve assembly case study showed that SGAPSO converged faster to high-quality solutions compared to GA and PSO. The synergistic mechanism of SGAPSO effectively combined the global exploration capabilities of GA with the local exploitation capabilities of PSO, leading to higher quality solutions. The balance rate and smoothness index further demonstrated the effectiveness of SGAPSO in achieving a well-balanced assembly line.

To formally validate the claim that SGAPSO provides a significant improvement over standalone GA and PSO, we conducted a statistical analysis of the results from the 30 independent runs on the pressure-reducing valve case study. Independent samples t-tests were performed to compare the mean objective function values (Z_avg) of

SGAPSO against both GA and PSO. The comparison between SGAPSO (Mean=185, SD=2.0) and GA (Mean=195, SD=2.5) yielded a t-statistic that corresponds to a p-value of less than 0.01. Similarly, the comparison between SGAPSO and PSO (Mean=192, SD=2.2) also resulted in a p-value of less than 0.01. In both cases, the p-values are well below the standard significance level of 0.05, indicating that the observed improvements in solution quality achieved by SGAPSO are statistically significant and not a result of random chance.

4.4 Analysis of SGAPSO components and synergy

To rigorously evaluate the SGAPSO framework, this section dissects the contributions of its constituent algorithms and the efficacy of their synergistic integration. Understanding these aspects is crucial for validating the design philosophy and pinpointing the sources of performance enhancement.

The investigation into the impact of synergy parameters-namely the synchronization period, the number of elite solutions transferred from GA to PSO ($N_{\text{elite-}to-PSO}$), and the number of elite solutions fed back from PSO to GA ($N_{PSO-\text{to}-\text{GA}}$)-was foundational. These parameters govern the frequency and intensity of information exchange between the GA and PSO components. Sensitivity analysis was performed by systematically varying each parameter while holding others at their determined baseline values (as specified in Section 4.1.2: $N_{\text{sync}} = 20, N_{\text{elite-}to-PSO} = 10, N_{PSO-to-GA} = 5$). The performance, measured by Z_{avg} and convergence speed, was observed across multiple runs on representative test instances. The results from this parametric study indicated that the chosen baseline values indeed provided a robust and consistently high level of performance across different problem instances. For example, a synchronization period of every 20 GA generations, coupled with the transfer of 10 elite solutions to PSO and the feedback of 5 refined solutions to GA, struck an effective balance between allowing each algorithm sufficient independent evolution and ensuring timely, beneficial information exchange. Shorter synchronization periods or excessive information transfer sometimes led to premature convergence or unnecessary computational overhead, while longer periods risked diminishing the synergistic benefits.

To further quantify the benefit of the proposed synergy, an ablation study was conducted by comparing SGAPSO against a nonsynergistic hybrid approach. This non-synergistic baseline typically involved running the GA to completion, followed by using its best-found solution to seed a subsequent, independent run of PSO, without the iterative feedback and elitesharing mechanisms inherent in SGAPSO. As anticipated and detailed in Table 4, SGAPSO significantly outperformed this simpler sequential combination in terms of both final solution quality and often the efficiency in reaching high-quality solutions. This disparity underscores that the true advantage of SGAPSO lies not merely in using both GA and PSO, but critically in their structured, periodic interaction. The elite guidance from GA helps PSO to focus its search on promising regions, while the refined solutions fed back from PSO enhance the GA's population quality, preventing stagnation and accelerating its convergence towards superior optima. The result is shown in Figure 5.

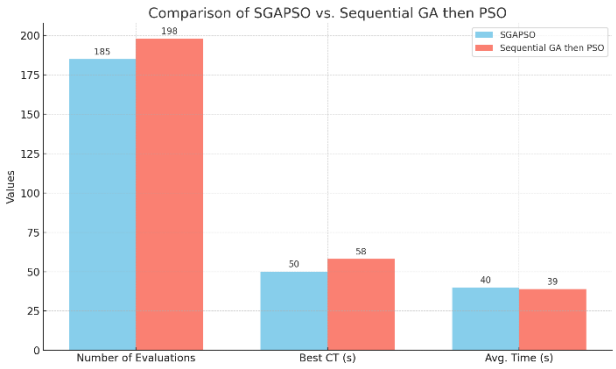


Figure 5: Ablation study - SGAPSO vs. sequential GA-PSO on pressure reducing valve case study

In essence, the synergistic mechanism—characterized by the periodic guidance of PSO exploration by GA-identified elites and the reciprocal enrichment of the GA population with PSO-refined solutions—proved to be the cornerstone of SGAPSO's enhanced performance. This structured interaction empowers SGAPSO to more effectively escape local optima compared to PSO operating in isolation, and to achieve a level of solution precision and refinement that often eludes GA when used alone. These findings robustly validate the design philosophy of SGAPSO, confirming that the intelligent combination of GA and PSO, through a well-defined synergistic linkage, offers a potent strategy for optimizing complex process parameters in the demanding context of modern intelligent manufacturing systems.

5 Conclusion

This paper has successfully addressed the intricate challenge of process parameter optimization within Reconfigurable Flexible Assembly Lines (RFALs), a critical issue in contemporary intelligent manufacturing systems. The research culminated in the development and validation of a novel Synergistic Genetic Algorithm and Particle Swarm Optimization (SGAPSO), specifically tailored to navigate the complexities posed by the multi-objective nature of RFAL balancing, which encompasses production cycle time (CT), tool replacement time (RT), and assembly unit cost (ST). The core innovation of SGAPSO lies in its meticulously engineered synergistic framework, which strategically integrates the global search proficiency of Genetic Algorithms (GA) with the local refinement and rapid convergence strengths of Particle Swarm Optimization (PSO). This was achieved through task sequence-based encoding and a dynamic interplay involving periodic elite-guided local searches by PSO, complemented by the assimilation of PSO-refined solutions back into the GA population. Rigorous experimental evaluations, encompassing standard

industry benchmarks (such as Jackson and Buxey instances) and a detailed pressure-reducing valve assembly case study, unequivocally substantiated SGAPSO's superior performance. The findings consistently demonstrated that SGAPSO not only achieves higher quality solutions, evidenced by improved Z_{best} and Z_{avg} values, but also exhibits more robust convergence patterns and enhanced solution stability (lower σ_Z) when compared against standalone GA and PSO methodologies, particularly in attaining superior balance rates and smoothness indices.

The contributions of this research extend beyond the mere development of a hybrid algorithm. The proposed SGAPSO framework offers a robust and adaptable tool that effectively balances the exploration-exploitation trade-off inherent in complex optimization landscapes, making it particularly well-suited for the nuanced demands of RFAL parameter optimization. The successful synergy achieved between GA and PSO underscores the potential of intelligent hybridization in tackling NP-hard problems in manufacturing. This work provides valuable insights for researchers and practitioners seeking to enhance the operational efficiency and economic viability of advanced manufacturing systems. Looking forward, the promising results from SGAPSO open several avenues for future investigation. These include the refinement of its adaptive synergistic strategies to further enhance performance across a wider array of problem instances, the extension of its application to address dynamic scheduling and real-time reconfiguration challenges in RFALs, and the incorporation of stochastic elements, such as equipment reliability and processing time variability, to create even more resilient and practically applicable optimization models for the next generation of intelligent manufacturing.

Funding

The research was supported by the General projects of “The 14th Five-Year Plan” project of Liaoning Education Science in 2024 (Research on the Construction Path of Enterprise-based Industrial Colleges in Higher Vocational Institutions), JG24EB230.

References

[1] Huang, B. (2025). Metaheuristic-Based Supply Chain Network Optimization and Inventory Management Using Ant Colony Algorithm. Informatica, 49(7), 17-32.

[2] Ren, L., Cui, L., Zhang, L., Lv, Z., & Wang, L. (2024). The industrial internet of things for smart manufacturing: A review. IEEE Transactions on Industrial Informatics, 20(2), 1275-1288. DOI: 10.1109/TII.2023.3242046

[3] Aldriny, J., El Fawal, A., & El Akoum, R. (2023). Fault diagnosis in smart manufacturing systems: A literature review on data-driven methods. Journal of Intelligent Manufacturing, 34(7), 2937-2968. DOI: 10.1007/s10845-022-01989-3

[4] Noor-A-Alam, M., Chowdhury, M. M. R., & Naha, R. K. (2022). Wireless Sensor Networks in Smart Manufacturing: A Comprehensive Review. Sensors, 22(3), 1023. DOI: 10.3390/s22031023

[5] Uysal, M., & Kose, U. (2021). Smart manufacturing systems: A systematic literature review and a conceptual framework. Journal of Industrial Information Integration, 23, 100184. DOI: 10.1016/j.jii.2020.100184

[6] Galankashi, M. R., Fallah, M., & Immonen, A. (2023). Digital Twin technology in manufacturing: a systematic literature review. Applied System Innovation, 6(1), 1. DOI: 10.3390/asi6010001

[7] Fortes, J. A., Ferreira, L. L., & Pitarma, R. (2023). A Smart Factory Architecture for Industry 4.0 Based on Digital Twin and Edge Computing. IEEE Access, 11, 34567-34580. DOI: 10.1109/ACCESS.2023.3265015

[8] Serag, A., Zaher, H., Ragaa, N., & Sayed, H. (2024). Improving the Emperor Penguin Optimizer Algorithm Through Adapted Weighted Sum Mutation Strategy with Information Vector. Informatica, 48(10), 65-76.

[9] Stojanovic, Ž., Dinic, M., & Stojadinovic, S. (2021). Software for smart sensors in smart manufacturing: architectures and applications. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 235(5), 845-858. DOI: 10.1177/0954405420980017

[10] Xie, Z., Liu, Y., & Wang, H. (2022). Research on RFID technology in smart manufacturing workshop management system. Proceedings of the 2022 International Conference on Information Management and Technology (ICIMTech), 1-5. DOI: 10.1145/3512715.3512730

[11] Samad, T., & Annaswamy, A. (2020). Smart manufacturing and smart grids: An overview and research challenges. IFAC-PapersOnLine, 53(2), 11967-11972. DOI: 10.1016/bs.ifacol.2020.12.123

[12] Marx, E., Riedel, O., & Verl, A. (2020). Closing the loop: A robust control framework for smart manufacturing based on digital twins. Procedia CIRP, 93, 295-300. DOI: 10.1016/j.procir.2020.04.044

[13] Samouei, P., & Gholami, S. (2019). Developing a multi-objective mathematical model for assembly line balancing problem with sequence-dependent setup times. Applied Mathematical Modelling, 68, 57-75. DOI: 10.1016/j.apm.2018.10.029

[14] Ma'ruf, A., & Santosa, B. (2024). A Mathematical Model for Reconfigurable Assembly Line Balancing Problem with Task Duplication and Worker Assignment. IEEE Access, 12, 1234-1245. DOI: 10.1109/ACCESS.2023.3345678

[15] Tran, T. K., Nguyen, T. A. T., & Le, T. H. (2024). Matheuristics for assembly line balancing: A GA-MIP approach for the simple assembly line

balancing problem. Applied Soft Computing, 150, 109876. DOI: 10.1016/j.asoc.2023.109876

[16] Mota, J. S., de Athayde Prates, R., & de Souza, M. C. (2022). An integer programming model for the simple assembly line balancing problem: A literature review and a new model. Computers & Industrial Engineering, 165, 107890. DOI: 10.1016/j.cie.2021.107890

[17] Eslami, R., & Behbahani, S. S. (2021). A Mathematical Programming Model for Robotic Assembly Line Balancing with Task Time Variability. Robotica, 39(5), 839-857. DOI: 10.1017/S026357472000098X

[18] Mondal, L. A., & Sen, S. (2025). Minimising assembly line balancing problem using simulation and metaheuristics. Applied Stochastic Models in Business and Industry. DOI: 10.1002/asmb.2876

[19] Villalobos, C. G., Azadeh, K., & Salgado, D. R. (2023). Balancing assembly lines with collaborative robots: Optimizing human-robot interaction and workload distribution. European Journal of Operational Research, 305(2), 780-795. DOI: 10.1016/j.ejor.2022.09.012

[20] Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: past, present, and future. Multimedia Tools and Applications, 80, 8091-8126. DOI: 10.1007/s11042-020-10139-6

[21] Gad, A. G. (2022). Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review. Archives of Computational Methods in Engineering, 29, 2531-2561. DOI: 10.1007/s11831-021-09694-4

[22] Chegini, S. N., Bagheri, A., & Najafi, F. (2018). PSO-SCALF: A new hybrid PSO based on Sine Cosine algorithm and levy flight for solving optimization problems. Applied Soft Computing, 73, 697-726. DOI: 10.1016/j.asoc.2018.09.014

[23] Coelho, C. E., et al. (2022). Low-Complexity PSO-Based Resource Allocation Scheme for Cooperative Non-Linear SWIPT-Enabled NOMA. IEEE Access, 10.1109/ACCESS.2022.3162838.

[24] Yang, J., et al. (2019). Extended Kalman Filter Based on Hybrid Particle Swarm Optimization and Genetic Algorithm. IEEE Access. DOI: 10.1109/ACCESS.2019.2901234

[25] Wang, Y., Li, X., & Gao, L. (2024). A hybrid genetic algorithm and particle swarm optimization for flexible job shop scheduling problem. Journal of Industrial Information Integration. DOI: 10.1016/j.jii.2020.100199

[26] Zhi, L., et al. (2024). Collaborative Path Planning for Multiple UAVs Based on Improved Genetic Particle Swarm Optimization Algorithm. Journal of Robotics. DOI: 10.3390/jrs3010002

[27] Samouei, P., & Gholami, S. (2019). Developing a multi-objective mathematical model for assembly line balancing problem with sequence-dependent setup times. Applied Mathematical Modelling, 68, 57-75. DOI: 10.1016/j.apm.2018.10.029

[28] Urbina, M. G., Li, C., Li, X., & Chen, W. H. (2019). Smart Sensors for Smart Manufacturing: A Review of Current Trends and Future Challenges. IEEE Internet of Things Journal, 6(5), 7543-7559. DOI: 10.1109/JIOT.2018.2879931

Construction and Validation of a Supply Chain Demand Forecasting Model Based on Embedded AI

Yonghui Ding[1*], Cancan Li[2]
[1]Business School, Nantong Institute of Technology, Nantong 226001, China
[2]The Logistics and Capital Construction Department, Nanjing University of Finance and Economics, Nanjing 210023, China
E-mail: dingyh201606@163.com
[*]Corresponding author

Keywords: embedded, demand, forecasting, BPNN, CNN-LSTM, parameter optimization

Received: July 4, 2025

Supply chain demand forecasting is an important basis for enterprise decision-making, which can not only help optimize resource allocation, but also enhance the market competitiveness of enterprises. To improve the accuracy of supply chain demand forecasting, a combined forecasting model considering univariate and multivariate variables is designed and embedded. On the univariate prediction model, the study considers optimizing the parameters of the backpropagation neural network through an improved whale optimization algorithm. On the multivariate prediction model, the study combines improved particle swarm optimization algorithm, convolutional neural network, and long short-term memory network. The findings showed that the accuracy, root mean square error, time consumption, and maximum memory usage of the univariate prediction model were 98.05%, 1.03%, 61ms, and 10.85%, respectively, which were significantly better than the comparison model. The maximum accuracy of the multivariate prediction model was 98.51%, the minimum was 96.02%, and the maximum root mean square error was 0.58. After embedded deployment, the maximum increase in time consumption of the combined prediction model was 47.76%, and the accuracy only decreased by 0.18%. The designed combination forecasting model has good performance and can provide model support for predicting supply chain demand.

Povzetek: Kombinirani napovedni model IWOA-BPNN in IPSO-CNN-LSTM, zasnovan za vgradne AI platforme, omogoča bolj kvalitetno napoved povpraševanja v dobavni verigi kot modeli ARIMA, RF, SVM in standardni CNN-LSTM, saj dosega višjo TOčnost, manjše napake ter učinkovitejšo porabo časa in pomnilnika.

1 Introduction

In the context of global economic integration and deep integration of digital technology, supply chain management has become a key battlefield for companies to enhance their core competitiveness. As the "nerve centre" of supply chain management, the forecasting of demand in the supply chain has been demonstrated to exert a direct influence on the management of inventory, the planning of production and the scheduling of logistics within enterprises. In addition, it is also related to customer satisfaction and the rate of response of markets [1-2]. According to McKinsey research, accurate demand forecasting can reduce inventory costs for businesses by 20%-30% and increase order fulfillment rates by 15%-25% [3-4]. Therefore, it is necessary to improve the accuracy of Supply Chain Demand Forecasting (SCDF). With regard to the issue in question, the extant literature proposes a number of methodologies for analysis. These include the application of statistical models based on historical data, human experience and judgement, time series analysis, causal models, and machine learning algorithms [5]. In addition, many researchers have already explored this issue.

Liu et al. developed a prediction model that integrates Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) algorithms to predict supply chain dynamics and address inventory optimization issues. The model was hyperparameter adjusted through Bayesian optimization. The findings denoted that this method could improve the operational efficiency and cost control effectiveness of enterprises [6]. Saad et al. developed a prediction algorithm that integrates an attention mechanism and bidirectional LSTM to precisely capture the dynamic fluctuations and patterns of supply chain logistics demand. This algorithm was trained through the utilization of the gradient descent method. Meanwhile, the study employed local outlier factors to remove outliers from the data. The findings demonstrated the efficacy of the algorithm in predicting demand within the supply chain logistics domain. Subsequent application of the algorithm resulted in an on-time delivery rate exceeding 95% on a monthly basis [7]. Zhang et al. developed a prediction method that utilizes the sequence-to-sequence and attention mechanisms to forecast demand for environmentally friendly electronic products within supply chains. The method has been adapted to address the dynamic and complex demands of the environmental

protection market by incorporating an end-to-end multi-step time prediction approach. Concurrently, the study utilized a linear regression approach founded on Huber Loss for the purpose of data processing. The findings demonstrated that the methodology outlined in this study exhibited superiority over conventional time series prediction models and machine learning regression models with regard to prediction accuracy [8]. Hasan et al. explored the feasibility of machine learning techniques in the field of SCDF, using five techniques including linear regression, elastic networks, random forests, multi-layer perceptron regressors, and extreme gradient boosting to comprehensively analyze a company's historical data. The results showed that linear regression was the most effective model with smaller errors [9].

However, current research and methods also have certain problems, such as the susceptibility of human experience judgments to individual cognitive biases, and the high computational cost of machine learning algorithms. To improve the accuracy of SCDF, a Univariate Prediction Model (UPM) considering Backpropagation Neural Network (BPNN) parameter optimization and a Multivariate Prediction Model (MPM) based on CNN and LSTM were designed and deployed as a combined prediction model on an embedded Artificial Intelligence (AI) platform. The research aims to improve the accuracy of SCDF, help enterprises plan inventory levels reasonably, avoid excessive inventory backlog or stockouts, and thereby reduce inventory holding costs and stockout costs. The research innovation combines the Improved Whale Optimization Algorithm (IWOA) and BPNN to improve prediction accuracy, reduce model time consumption, and ultimately lower the computational cost of the model [10].

2 Methods and materials

To predict supply chain demand, a combined forecasting model including UPM and MPM is designed starting from sales data. To improve the inference speed of the combined prediction model, it is quantitatively optimized and deployed on an embedded AI platform.

2.1 Construction of a univariate prediction model considering BPNN parameter optimization

E-commerce sales data is an important basis for SCDF. To predict supply chain demand, a combined forecasting model is designed considering the characteristics of e-commerce and retail sales data. Moreover, the combined prediction model includes a UPM based on BPNN parameter optimization and an MPM based on CNN-LSTM. To enable high-speed model inference of the research and design combination prediction model, it is deployed on an embedded AI platform. When constructing a UPM, time series data is used and normalized to raise the model's prediction accuracy. The expression of normalization processing is shown in equation (1) [11].

$$\bar{A}_B = \frac{A_B - A_{\min}}{A_{\max} - A_{\min}} \tag{1}$$

In equation (1), $\bar{A}_B$ indicates the normalized data value, and A_B represents the B th original data. $A_{\max}$ and $A_{\min}$ refer to the max and mini values of the original data, respectively. BPNN is a neural network based on error BP algorithm, which has the advantages of strong nonlinear mapping ability, adaptability, generalization ability, and easy implementation. Compared with other traditional time series prediction methods, it has higher prediction accuracy [12-13]. Therefore, the study uses it as the basis for UPMs. However, the initial weights and thresholds of the BPNN can affect the convergence speed and results of the algorithm. Therefore, to better utilize the role of BPNN, an IWOA is designed to optimize its parameters. The advantage of the WOA is its fast convergence speed, few parameters, and ease of adjustment, but it is also prone to getting stuck in local optima [14]. Therefore, the study makes three improvements to it. Improvement one is to introduce Circle mapping to strengthen the diversity of the initial population. Meanwhile, to make the chaotic value distribution of the Circle map more uniform, slight adjustments are made to it, and the adjusted Circle map is shown in equation (2).

$$C_{D+1} = \mathrm{mod}\left(3C_D + 0.4 - \left(\frac{0.5}{3\pi}\sin\left(3\pi \cdot C_D\right), 1\right)\right) \tag{2}$$

In equation (2), C_{D+1} represents the next chaotic variable value after mapping, and C_D is the current chaotic variable value. D means the dimension of the solution, which is the number of problem variables. $\mathrm{mod}(\cdot,1)$ represents modulo operation. The second improvement is to introduce adaptive inertia weight E to dynamically adjust the global and local search capabilities of the algorithm. The expression of E is shown in equation (3).

$$E = \sin\left(\frac{\pi t}{2t_{\max}} + \pi\right) + 1 \tag{3}$$

In equation (3), t denotes the current amount of iterations, and $t_{\max}$ refers to the max amount of iterations. At this point, the position update expression of the WOA is shown in equation (4).

$$\vec{F}(G+1) = \begin{cases} E \cdot \overrightarrow{F^*}(G) - H \cdot I & , J < 0.5 \\ \vec{I} \cdot \mathrm{e}^{JK} \cdot \cos(2\pi l) + E \cdot \overrightarrow{F^*}(G) & , J \geq 0.5 \end{cases} \tag{4}$$

In equation (4), $\vec{F}(G+1)$ represents the position update vector at time $G+1$, and $\overrightarrow{F^*}(G)$ is the position vector of the optimal whale individual at time G. H represents the coefficient vector, and J represents the random probability. K is a random vector, e^{JK} is a spiral equation. $\vec{I}$ represents the distance between the current whale and its prey, while I denotes the distance vector between the individual whale and its optimal position. The third improvement is the introduction of Levy flight

strategy to balance local search and global search, thereby avoiding falling into local optima. The probability density function $Levy(L)$ expression of Levy's flight strategy is shown in equation (5) [15].

$$Levy(L)=\frac{1}{\pi}\int_0^{+\infty}\left(-M\mid N\mid^P\right)\cos(NL)dN \tag{5}$$

In equation (5), L represents the step size variable, and L is the shape control parameter. N and P are integral variables and decay rate control parameters, respectively. dN is the derivative of N. The expression of Levy's flight random step size $Levy(M)$ is shown in equation (6).

$$Levy(M)=L=\frac{Q}{\mid R\mid^{1/M}} \tag{6}$$

In equation (6), both R and Q represent random variables. After updating the position in WOA, the study uses Levy flight to update the position again, and the expression is denoted in equation (7).

$$\vec{F}(G+1)=F(G)+S\oplus \text{Levy}(M) \tag{7}$$

In equation (7), $\oplus$ represents dot product operation. S is the step size factor. The main process of the IWOA is denoted in Figure 1.

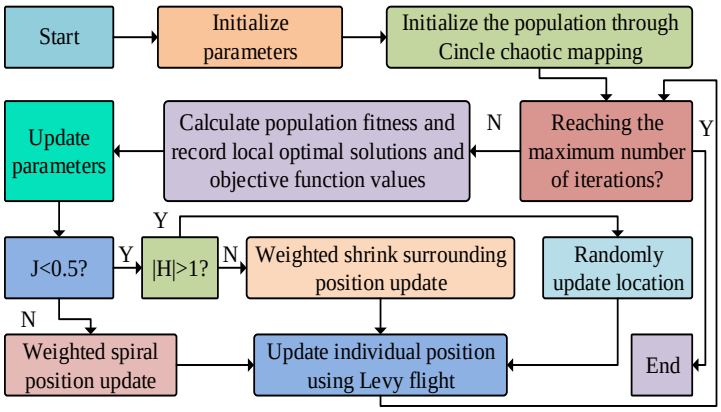


Figure 1: The main process of IWOA

From Figure 1, the main process of the IWOA includes initializing the population through Circle chaotic mapping, updating individual positions using Levy flight, determining whether the max amount of iterations has been reached, and updating parameters. The employment of the IWOA facilitates the optimization of the initial weights and thresholds of the BPNN, thereby enhancing its predictive efficacy. The IWOA-BPNN's main process is denoted in Figure 2.

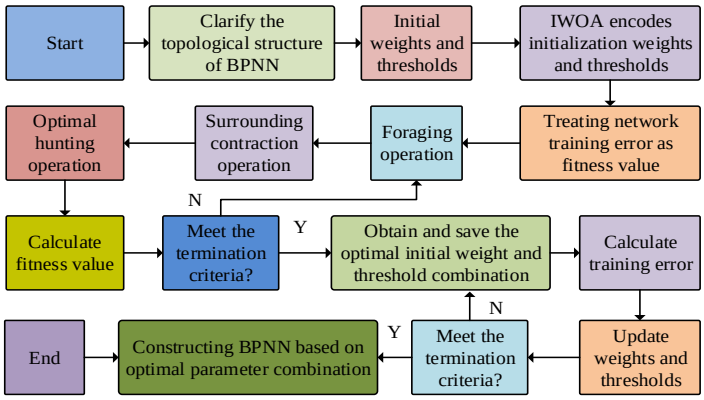


Figure 2: The main process of IWOA-BPNN

From Figure 2, the main process of IWOA-BPNN includes clarifying the topology structure of BPNN, encoding initial weights and thresholds, calculating fitness, obtaining and saving the optimal initial weight and threshold combination, and constructing BPNN based on the optimal parameter combination. After constructing a UPM based on optimal initial weights and thresholds using BPNN, the supply chain needs can be predicted based on normalized time series data. The pseudo-code of the IWOA-BPNN is shown in Figure 3.

```
Initialize BPNN structure and parameters.
Generate initial population via Circle mapping.
Evaluate fitness of each individual.
for each iteration to MaxIter do
    for each individual in population do
        if condition met then
            Update position using Levy flight.
        else
            Update position based on fitness.
        end if
    end for
    Update global optimal individual.
end for
Train BPNN with optimal weights and thresholds from IWOA.
Return trained BPNN model.
```

Figure 3: Pseudo-code of IWOA-BPNN

From Figure 3, the pseudo-code of the IWOA-BPNN clearly presents the core process of the IWOA-BPNN. First, the BPNN structure and parameters are initialized, and then the initial population is generated by Circle mapping and its fitness is evaluated. Subsequently, it enters the iteration loop, where in each iteration, based on conditional judgment, the individual position is updated using Levy flight or fitness based methods, and the global optimal individual is updated. Finally, the weights and thresholds optimized by IWOA are used to train BPNN, resulting in a well-trained model.

2.2 Construction of multivariate prediction model considering CNN-LSTM model

To predict supply chain demand, a UPM is designed for the combination forecasting model. Due to the non-stationary nature of e-commerce sales data and the presence of multiple variables (such as advertising investment, promotion strategies, etc.) that affect sales data, single variable prediction may not be accurate enough. To improve the accuracy of SCDF, an MPM based on CNN-LSTM model is studied and designed. Before constructing the prediction model, data preprocessing is necessary. In terms of missing values, the study uses the median to fill in. On outliers, the study directly removes them. To improve the processing speed of the model, the data is normalized and utilized as input for the MPM. Due to the non-stationary nature of sales value time series data, the study also uses natural logarithm to transform the data, to remove its non stationarity and facilitate subsequent feature acquisition. In addition, the study applies Pearson correlation coefficient to analyze the correlation between different features. After processing the data, the study constructs a CNN-LSTM prediction model, and the model's main process is denoted in Figure 4.

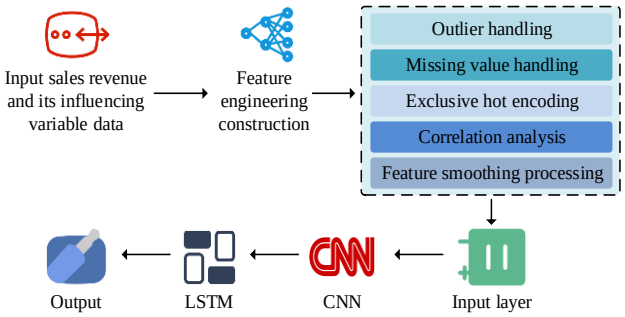


Figure 4: The main process of CNN-LSTM prediction model

From Figure 4, the CNN-LSTM prediction model requires input of sales revenue and its influencing variable data, and feature engineering construction of this data, including outlier handling, correlation analysis, and feature smoothing. The processed features will enter the input layer and obtain data features through the CNN layer. Afterwards, sequence prediction is achieved through an LSTM layer, and prediction data is output through a fully connected layer. CNN is a deep neural network mainly composed of convolutional layers, pooling layers, activation functions, and fully connected layers. It can not only automatically learn local features in input data, but also achieve multi-layer feature extraction [16-17]. In CNN, causal padding is used to ensure that only the input data of the current and previous time steps are used when calculating the output of the current time step, without considering the data of future time steps, to maintain the causal relationship of the time series. In CNN, the activation function used in the study is Sigmoid, which is expressed as equation (8).

$$Sigmoid = \frac{1}{1+e^{U}} \tag{8}$$

In equation (8), U represents the input value of the function. LSTM is a special type of recurrent neural network that can not only capture the dependencies of long time intervals in time series, but also dynamically adjust the flow and storage of information. LSTM is made up of a forgetting gate, an inputting gate, and an outputting gate, and it updates the feature information of the hidden layer through the forget gate [18-19]. In LSTM, the expression of the hidden layer V step state W_V is shown in equation (9).

$$W_V = f\left(XY_V + ZW_{V-1}\right) \tag{9}$$

In equation (9), $f(\cdot)$ represents the nonlinear activation function. X means the connection matrix of the input layer. Y_V represents the input of step Y_V. Z is the weight matrix from the previous hidden layer to the next hidden layer. W_{V-1} represents the state of the hidden layer at the previous moment. The expression of the output O_V at step V is shown in equation (10).

$$O_V = a\left(b * W_V\right) \tag{10}$$

In equation (10), a represents the activation function, and b means the weight matrix of the output layer. However, the selection of parameters such as

learning rate and hidden layers in CNN-LSTM models can affect the performance of the model. To find the best parameters for the CNN-LSTM model, an Improved Particle Swarm Optimization (IPSO) algorithm is created. PSO algorithm is a swarm intelligence optimization algorithm with advantages such as simplicity and ease of implementation. It is widely used in engineering optimization and machine learning parameter tuning fields [20]. However, the performance of PSO algorithm relies heavily on the initial particle swarm quality and is prone to getting stuck in local optima. Therefore, the study makes four improvements to the PSO algorithm.

Improvement one is to combine Sobol sequence and K-means clustering to make the initial population distribution of PSO more uniform. The Sobol sequence is a low-variance sequence. This means that it can generate sample points that are spread evenly across a high number of dimensions. The study uses a Sobol sequence generator to create a sample pool, evaluates and sorts the fitness of the samples, and then analyzes the sorted samples using K-means clustering. The second improvement is to introduce non-linear inertia weight d to balance the exploration and development capabilities of the algorithm in different periods. The expression of d is shown in equation (11).

$$d = d_{\max} - \frac{\left(d_{\max} - d_{\min}\right)}{1 + \exp\left(\frac{10t}{t_{\max}} - 5\right)} \tag{11}$$

In equation (11), $d_{\max}$ and $d_{\min}$ represent the max and mini-inertia weight values, respectively. Improvement three is to introduce an inertia weight direction adaptive change strategy to balance the global and local search capabilities of PSO and overcome the problem of local optima. The key to this strategy is the inertia direction change coefficient g_h, which is expressed as equation (12).

$$g_h = \begin{cases} 1 & j\left(m_h^t\right) > j\left(m_h^{t-1}\right) \\ -rand & j\left(m_h^t\right) > j\left(m_h^{t-1}\right) \end{cases} \tag{12}$$

In equation (12), h represents the particle number, $rand$ represents the random number within the range, $j\left(m_h^t\right)$ and $j\left(m_h^{t-1}\right)$ represent the fitness values of particle h at the t th and $t-1$th iterations, respectively. Under this strategy, the update of particle velocity is denoted in equation (13).

$$v_{hq}^t = g_h \cdot dv_{hq}^{t-1} + x_1 y_1 \left(\beta_{best-hq}^{t-1} - m_{hq}^{t-1}\right) + x_2 y_2 \left(\varphi_{best-q}^{t-1} - m_{hq}^{t-1}\right) \tag{13}$$

In equation (13), v_{hq}^t and v_{hq}^{t-1} denote the velocities of the h th particle in the q th dimension at the t th and $t-1$th iterations, respectively, while x_1 and x_2 indicate the individual learning factor and social learning factor, respectively. y_1 and y_2 are both random numbers, and $\beta_{best-hq}^{t-1}$ and $\beta_{best-hq}^{t-1}$ are the individual and global optimal positions in the $t-1$ th iteration, respectively. m_{hq}^{t-1} is the position of the h th particle in the q th dimension at the $t-1$ th iteration. The fourth improvement is the introduction of an adaptive hierarchical learning strategy to enhance the PSO optimization capability. This strategy requires dividing it into three layers based on particle fitness ranking and adopting different learning strategies for each layer. The main process of IPSO algorithm is denoted in Figure 5.

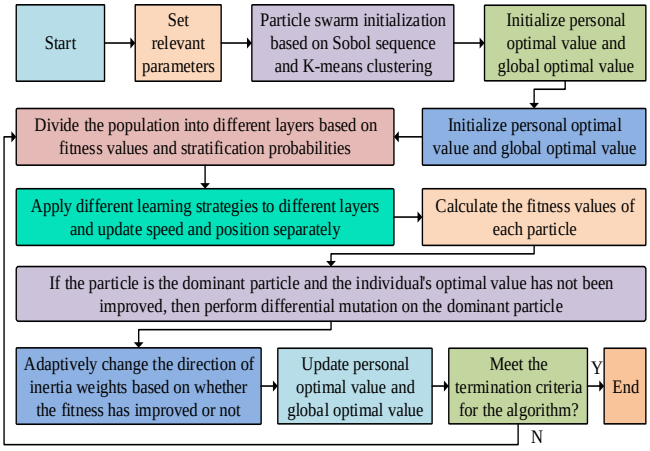


Figure 5: The main process of IPSO algorithm

From Figure 5, the main process of the IPSO algorithm involves particle swarm initialization based on Sobol sequences and K-means clustering, population stratification, particle fitness calculation, and adaptive change of inertia weight direction. Through IPSO, important parameters of CNN-LSTM can be optimized to raise the effectiveness of the prediction model. The main process of IPSO-CNN-LSTM is denoted in Figure 6.

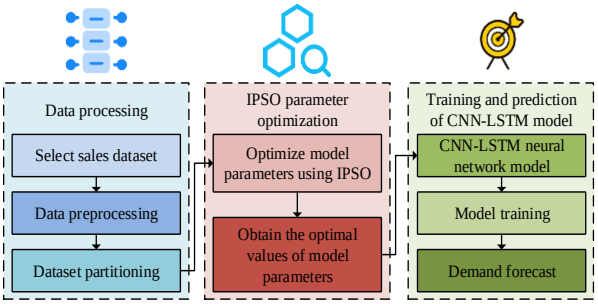


Figure 6: The main process of IPSO-CNN-LSTM

From Figure 6, the IPSO-CNN-LSTM prediction model is mainly divided into three parts: data processing, IPSO parameter optimization, and CNN-LSTM model training and prediction. Based on the prediction model after parameter optimization, the study is able to provide better prediction of supply chain demand. The pseudo-code of the IPSO-CNN-LSTM prediction model is shown in Figure 7.

```
Initialize BPNN structure and parameters.
Generate initial population via Circle mapping.
Evaluate fitness of each individual.
for each iteration to MaxIter do
  for each individual in population do
    if condition met then
      Update position using Levy flight.
    else
      Update position based on fitness.
    end if
  end for
  Update global optimal individual.
end for
Train BPNN with optimal weights and thresholds from
IWOA.
Return trained BPNN model.
```

Figure 7: Pseudo-code for IPSO-CNN-LSTM prediction model

From Figure 7, the pseudo-code fully demonstrates the key process of the IPSO-CNN-LSTM prediction model. First, the structure of the CNN-LSTM model is initialized, and then the initial population is generated and evaluated for fitness based on Sobol sequences and K-means clustering. During the iteration process, adaptive strategies are used to update individual speed and position, continuously adjusting the global optimal individual. Finally, the IPSO optimized parameters are used to train the CNN-LSTM model, resulting in an accurate prediction model.

A combination forecasting model for SCDF is designed. To improve the portability of the combined prediction model and reduce its cost and power consumption, an embedded design is studied. In the training of the model, two methods are adopted: model quantification and algorithm optimization to obtain a model that is more suitable for embedded AI platforms. In addition, the study selects NVIDIA Jetson TX2 as the embedded AI platform for deploying the combined prediction model. The NVIDIA Jetson TX2 has powerful computing and AI inference capabilities and low hardware costs. In the model quantization optimization, the NVIDIA inference acceleration scheme TensorRT is adopted. The TensorRT optimization process is shown in Figure 8.

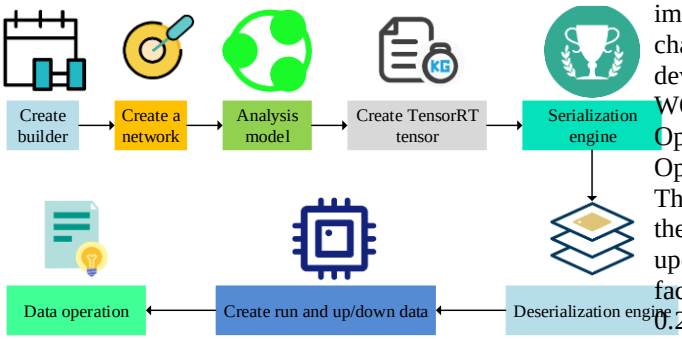


Figure 8: The optimization process of TensorRT

From Figure 8, the TensorRT optimization process mainly includes creating builders, creating TensorRT tensors, and creating runs and upper and lower data. In addition, TensorRT can also perform memory management on models to optimize their structure, making them better suited for embedded AI platforms and improving inference speed. The embedded design of the supply chain demand combination prediction model can improve the prediction speed and efficiency of the model.

3 Results

To assess the effect of the research-designed prediction model, the experimental environment and apparatus were described, and the data used in the experiment were set. In addition, the study also selected comparative models and analyzed them based on indicators such as prediction accuracy and error.

3.1 Performance verification of parameter optimization prediction model based on BPNN

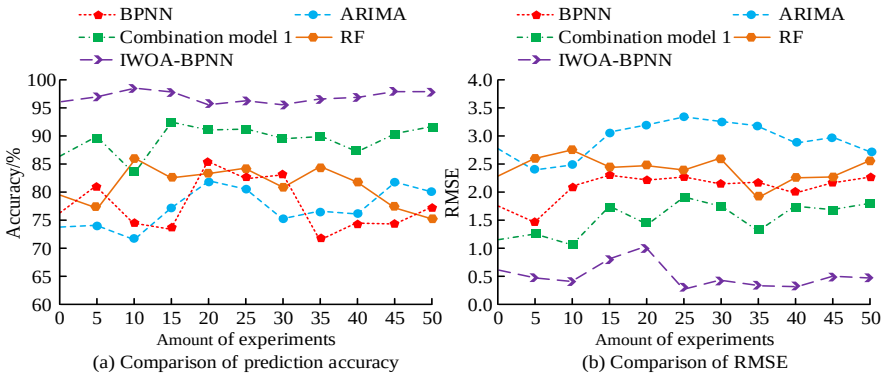
To validate the performance of a single factor prediction model based on BPNN parameter optimization, the study selected the monthly demand (sales data) of learning supplies in an e-commerce mall from January 2020 to December 2024 for empirical analysis, and preprocessed the obtained data. "Learning supplies" included various stationery items such as paper, pens, and notebooks, as well as commonly used learning tools for students such as calculators and folders. In addition, the study selected two test functions from the CEC2013 test function set to validate the performance of the IWOA, namely the unimodal function $f4$ and the multimodal function $f18$ [21]. The CEC2013 test function set is a benchmark function set widely used for optimizing algorithm performance evaluation, including 28 different types of functions, including unimodal functions, multimodal functions, and combination functions, used to simulate various complex optimization problems. Among them, unimodal functions such as spherical functions are used to test algorithm convergence speed, multimodal functions such as cross benchmark functions are used to evaluate algorithm exploration ability, and combination functions further increase complexity through translation, rotation, and other operations. It is widely used in research fields such as intelligent optimization algorithms, large-scale optimization problem solving, and algorithm improvement innovation, providing a standardized and challenging testing platform for the research and development of optimization algorithms. Meanwhile, WOA, Sparrow Search Algorithm (SSA), Seagull Optimization Algorithm (SOA), and Grey Wolf Optimization (GWO) were also selected for comparison. The population size of these comparative models was 40, the maximum iteration number was 400, and the spiral update parameter of WOA was 1.5, with a convergence factor of 2. In addition, the discoverer ratio of SSA was 0.2, and the number of seagulls and inertia weight of SOA were 40 and 0.5, respectively. The operating system used in the study was Windows 10, and the central processing unit was Intel Core i5-13500. In terms of prediction models, the study selected BPNN, Auto-Regression Integrated Moving Average (ARIMA), Random Forest (RF), WOA-BPNN, and a combination model 1 that combines Support Vector Machine (SVM) regression and LSTM for comparison. The population size of the IWOA was 40, the max amount of iterations was 400, and the

learning rate of the BPNN was 0.02. The comparison of the average values of different algorithms under different test functions is denoted in Table 1.

Table 1: Comparison of average values of different algorithms under different test functions

Algorithm	Unimodal function *f*4						
	Amount of experiments						
	1	2	3	4	5	6	7
SSA	3.71×10^3	2.38×10^3	2.98×10^3	2.81×10^3	3.10×10^3	3.10×10^3	2.59×10^3
SOA	1.96×10^3	2.07×10^3	2.30×10^3	2.55×10^3	1.98×10^3	1.88×10^3	2.62×10^3
GWO	1.63×10^3	2.05×10^3	1.44×10^3	1.59×10^3	1.52×10^3	1.81×10^3	1.91×10^3
WOA	1.41×10^3	1.37×10^3	1.33×10^3	1.26×10^3	1.43×10^3	1.34×10^3	1.42×10^3
IWOA (manuscript)	0.22×10^3	0.27×10^3	0.30×10^3	0.33×10^3	0.25×10^3	0.21×10^3	0.26×10^3
Algorithm	Multimodal *f*18						
	Amount of experiments						
	1	2	3	4	5	6	7
SSA	2.43×10^2	2.51×10^2	1.94×10^2	2.44×10^2	2.35×10^2	2.29×10^2	1.77×10^2
SOA	2.56×10^2	1.18×10^2	2.37×10^2	1.86×10^2	1.87×10^2	1.77×10^2	1.74×10^2
GWO	1.56×10^2	1.90×10^2	1.16×10^2	1.38×10^2	1.92×10^2	1.53×10^2	2.19×10^2
WOA	1.46×10^2	1.26×10^2	1.37×10^2	1.54×10^2	1.20×10^2	1.01×10^2	1.10×10^2
IWOA (manuscript)	0.32×10^2	0.28×10^2	0.34×10^2	0.26×10^2	0.29×10^2	0.28×10^2	0.25×10^2

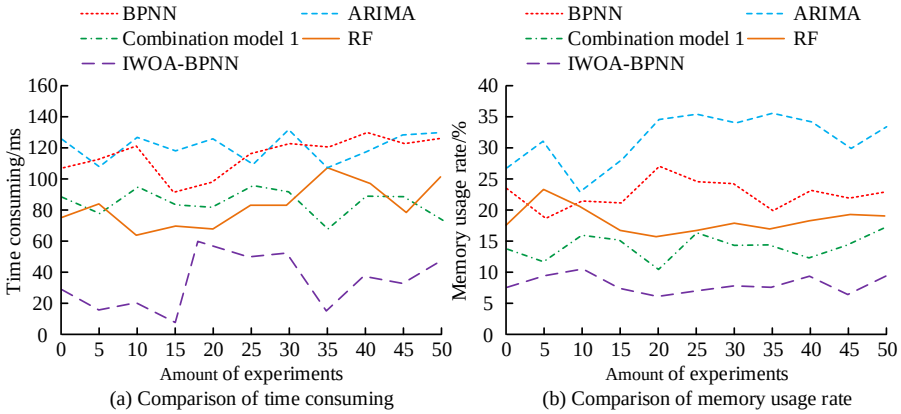
From Table 1, under the unimodal function *f*4 and multimodal function *f*18, the average values of the IWOA were significantly lower than those of the comparison algorithm. For example, under the unimodal function *f*4, the average values of the IWOA were all below 1.0×10^3, while the average values of SSA, SOA, GWO, and WOA were significantly greater than 1.0×10^3. This indicates that the IWOA has better optimization ability and stability under both test functions, and can better optimize the initial weights and thresholds of the BPNN. This may be because the IWOA introduces Circle chaotic mapping, adaptive inertia weight, and Levy flight strategy, which improves the algorithm's global search ability, local optimization ability, and ability to escape from local optima, thereby achieving better results in the optimization of unimodal and multimodal functions. The comparison of prediction accuracy and Root Mean Square Error (RMSE) of different UPMs is shown in Figure 9.



(a) Comparison of prediction accuracy (b) Comparison of RMSE

Figure 9: Comparison of prediction accuracy and RMSE of different univariate prediction models

From Figure 9 (a), in terms of comparison of prediction accuracy, the max value of the IWOA-BPNN model was 98.05%, and the mini value was 95.27%. The maximum prediction accuracy of BPNN, ARIMA, RF, and combination model 1 were 85.14%, 82.09%, 85.37%, and 92.68%, respectively, all of which were less than 98.05%. From Figure 9 (b), in the comparison of RMSE, the IWOA-BPNN model designed in the study had the smaller error, followed by the combination model 1. In terms of specific values, the maximum RMSE of the IWOA-BPNN model was 1.03 and the minimum was 0.25, which was much lower than the comparison model. This indicates that the IWOA-BPNN model designed for research can make more accurate predictions of supply chain demand. The comparison of time consumption and memory usage of different UPMs is shown in Figure 10.



(a) Comparison of time consuming (b) Comparison of memory usage rate

Figure 10: Comparison of time consumption and memory usage of different univariate prediction models

As shown in Figure 10 (a), in terms of time consumption comparison, the max values of the IWOA-BPNN model and the four comparison models were 61ms, 129ms, 132ms, 108ms, and 95ms, respectively. The time consumption of the IWOA-BPNN model was greatly lower than that of the comparison model. From Figure 10 (b), the IWOA-BPNN model performed better in terms of memory usage, with a max value of 10.85% and a mini value of 6.03%. Overall, the IWOA-BPNN model has lower time and memory consumption and better performance. To further validate the performance of the IWOA-BPNN model, the study selected Mean Absolute Error (MAE) and inference delay metrics. The comparison of MAE and inference delay for different models is shown in Table 2.

Table 2: Comparison of MAE and inference delay among different models

Algorithm	MAE					Inference delay/ms				
	Number of experiments					Number of experiments				
	1	2	3	4	5	1	2	3	4	5
BPNN	2.440	1.746	1.894	2.362	1.723	13.75	15.80	16.51	16.27	13.30
ARIMA	3.201	3.188	3.483	3.089	2.986	20.79	20.76	20.29	18.30	16.27
RF	2.236	2.080	2.017	2.783	2.104	14.67	16.43	18.29	16.83	15.45
Combination model 1	1.875	1.155	1.212	1.173	1.740	9.06	8.15	7.41	11.66	8.35
IWOA-BPNN	0.307	0.681	0.914	0.359	0.799	2.82	4.02	3.10	4.53	4.34

From Table 2, the proposed IWOA-BPNN model outperformed other compared algorithms in terms of MAE and inference delay. Specifically, the average MAE of the IWOA-BP neural network model was 0.612, while the average MAE of the BPNN model, ARIMA, RF, and combination model 1 were 2.033, 3.189, 2.244, and 1.431, respectively. The MAE of the IWOA-BPNN model was the smallest, indicating that its prediction error was the smallest and its stability was high, which could more accurately predict supply chain demand. In addition, in terms of inference delay, the IWOA-BPNN model had significantly lower values than other models, indicating that its inference efficiency was higher and could meet the high real-time requirements of supply chain prediction scenarios.

3.2 Performance validation of multivariate prediction model considering CNN-LSTM

To validate the effect of the MPM based on CNN-LSTM, the same operating system and central processing unit were used, and the NVIDIA Jetson TX2 embedded AI platform was configured. The study selected the sales volume of daily necessities in a certain shopping mall from January 2020 to December 2024 for analysis, preprocessed it, and analyzed the correlation coefficients between features. "Daily necessities" encompassed essential items for daily life, such as food, cleaning supplies, and personal care products. Since these correlation coefficients were not high, there was no need to further reduce their dimensionality. In addition, advertising investment was measured by the total budget of advertising activities, in units of yuan, and was obtained by summarizing the amount of advertising invested by enterprises in different advertising channels (such as television, internet, newspapers, etc.). The promotional strategy quantified the intensity of promotional activities, using discount intensity as an example, expressed as the ratio of the discounted price to the original price. Other influencing variables such as social media popularity were quantified by the search volume of relevant keywords or topic popularity, while market trends were quantified by referring to trend indicators in market research reports or industry analysis articles. In the validation of the IPSO algorithm, the same testing function was used, and PSO, Genetic Algorithm (GA), WOA, and a combination model 2 combining GA and BPNN were selected for comparison. In terms of prediction models, CNN-LSTM, SVM, PSO-CNN-LSTM, and a combination model 3 combining PSO and extreme gradient boosting were selected for comparison. The optimizer of the model was Adam, the learning factor of PSO was 1.5, the particle swarm size was 40, and the dimension was 30. To improve the transparency and practicality of the model, the SHapley Additive exPlans (SHAP) technology was introduced in

the study. SHAP values can quantify the contribution of each feature to the prediction results, helping to identify the factors that have the greatest impact on demand forecasting. The SHAP values of different variables are shown in Table 3.

Table 3: SHAP values of different variables

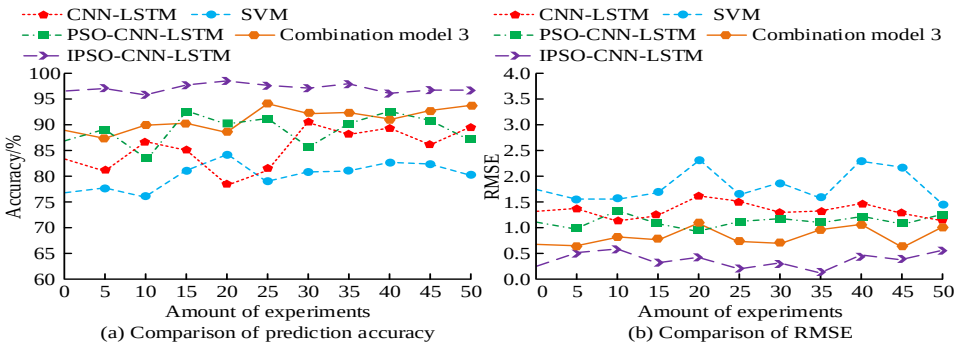
Characteristic variable	SHAP value	Characteristic variable	SHAP value
Sales volume	0.45	Store sales personnel	0.08
WeChat push frequency	0.15	Store poster display	0.05
Local TV advertising investment	0.25	Store promotion events_specific	0.30
Online advertising investment	0.20	\	\

From Table 3, sales volume had the greatest contribution to SCDF, with an SHAP value as high as 0.45. The store promotion event was followed by the special event, with an SHAP value of 0.30. These two are key factors that affect demand forecasting. The positive impact of local TV advertising investment (0.25) and online advertising investment (0.20) on demand should not be underestimated. In contrast, although WeChat push frequency (0.15), store sales personnel (0.08), and store poster display (0.05) had some impact, they were relatively small. If enterprises want to improve prediction accuracy, they can focus on data collection and analysis in sales, store promotions, and advertising investment. The standard deviation comparison of different algorithms under different test functions is shown in Table 4.

Table 4: Comparison of standard deviations of different algorithms under different test functions

Algorithm	Unimodal function *f*4						
	Amount of experiments						
	1	2	3	4	5	6	7
PSO	3.96×10^{3}	3.82×10^{3}	3.99×10^{3}	3.94×10^{3}	3.72×10^{3}	3.61×10^{3}	3.59×10^{3}
GA	2.50×10^{3}	3.27×10^{3}	2.95×10^{3}	2.92×10^{3}	2.87×10^{3}	2.74×10^{3}	2.76×10^{3}
WOA	2.76×10^{3}	2.25×10^{3}	2.37×10^{3}	2.95×10^{3}	2.35×10^{3}	2.01×10^{3}	2.81×10^{3}
Combination model 2	1.47×10^{3}	1.92×10^{3}	1.79×10^{3}	1.16×10^{3}	1.97×10^{3}	1.53×10^{3}	1.10×10^{3}
IPSO (manuscript)	0.85×10^{3}	0.82×10^{3}	0.80×10^{3}	0.77×10^{3}	0.81×10^{3}	0.80×10^{3}	0.76×10^{3}
Algorithm	Multimodal *f*18						
	Amount of experiments						
	1	2	3	4	5	6	7
PSO	5.72×10^{1}	6.45×10^{1}	5.55×10^{1}	5.86×10^{1}	5.63×10^{1}	5.93×10^{1}	6.08×10^{1}
GA	4.57×10^{1}	4.61×10^{1}	4.52×10^{1}	4.88×10^{1}	4.60×10^{1}	5.38×10^{1}	4.51×10^{1}
WOA	3.79×10^{1}	4.06×10^{1}	4.12×10^{1}	3.91×10^{1}	3.98×10^{1}	4.47×10^{1}	4.19×10^{1}
Combination model 2	2.25×10^{1}	2.22×10^{1}	1.59×10^{1}	1.98×10^{1}	1.93×10^{1}	1.60×10^{1}	1.71×10^{1}
IPSO (manuscript)	0.14×10^{1}	0.15×10^{1}	0.12×10^{1}	0.14×10^{1}	0.13×10^{1}	0.16×10^{1}	0.11×10^{1}

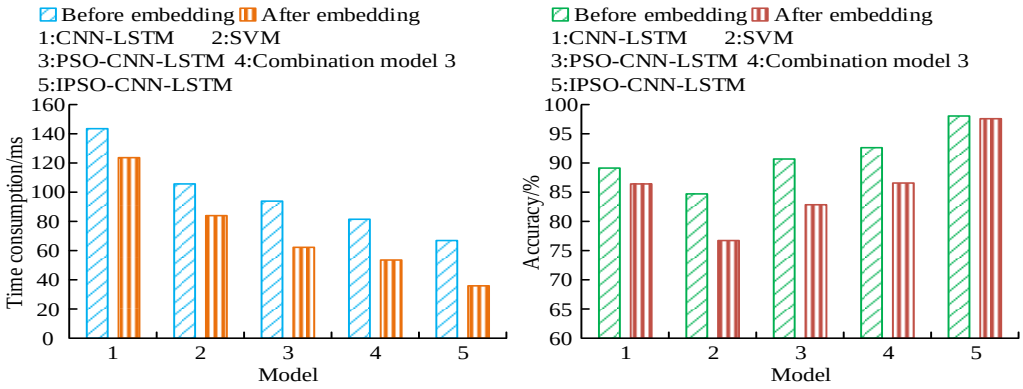
From Table 4, the standard deviation of IPSO algorithm was significantly lower than that of the comparison algorithm under unimodal *f*4 and multimodal *f*18. For example, under the multimodal function *f*18, the standard deviation values of the IPSO algorithm were all below 1.0×10^{1}, while the comparison algorithms were all above 1.0×10^{1}. This indicates that the optimization results of IPSO algorithm have higher stability and reliability. This may be because the IPSO algorithm has improved the initialization strategy by introducing nonlinear inertia weights and inertia weight direction adaptive change strategies. The accuracy and RMSE comparison of different MPMs are shown in Figure 11.



(a) Comparison of prediction accuracy (b) Comparison of RMSE

Figure 11: Comparison of accuracy and RMSE of different multivariate prediction models

According to Figure 11 (a), the maximum prediction accuracy of the IPSO-CNN-LSTM model was 98.51%, and the minimum was 96.02%. The maximum prediction accuracies of CNN-LSTM, SVM, PSO-CNN-LSTM, and combination model 3 were 90.55%, 84.25%, 92.75%, and 94.69%, respectively, all of which were less than 98.51%. According to Figure 11 (b), the IPSO-CNN-LSTM model had a smaller RMSE, with maximum RMSE values of 0.58, 1.59, 2.38, 1.37, and 1.15 compared to the four comparison methods. The IPSO-CNN-LSTM model performed better. The comparison of time consumption and accuracy before and after embedding the IPSO-CNN-LSTM model is shown in Figure 12.



(a) Comparison of time consumption before and after embedding

(b) Comparison of accuracy before and after embedding

Figure 12: Comparison of time consumption and accuracy before and after embedding IPSO-CNN-LSTM model

According to Figure 12 (a), in terms of time consumption, the maximum values before and after embedding the IPSO-CNN-LSTM model were 67ms and 35ms, respectively, with an improvement ratio of 47.76% between the two. According to Figure 12 (b), the maximum difference in accuracy between the IPSO-CNN-LSTM model before and after embedding was 0.18%. Deploying the combination prediction model on an embedded AI platform could reduce the model's time consumption with only a 0.18% decrease in accuracy.

4 Discussion and conclusion

A combination prediction model was designed to improve the accuracy of SCDF, which includes an IWOA-BPNN UPM and an IPSO-CNN-LSTM MPM. The findings denoted that under the unimodal function *f*4, the average values of the IWOA were all below 1.0×10^3, significantly lower than the comparison algorithms. Moreocver, the average value of the IWOA under the multimodal function *f*18 was also smaller. This indicated that the IWOA had better optimization ability and stability, and could better optimize the initial weights and thresholds of the BPNN. The max prediction accuracy of the UPM was 98.05%, which was 12.91%, 15.96%, 12.68%, and 5.37% higher than the max values of the comparison model, respectively. This indicates that the model can make more accurate predictions of supply chain demand, and this may be due to its use of parameter combinations optimized by the IWOA. The standard deviation of IPSO algorithm under unimodal *f*4 and multimodal *f*18 was significantly lower than that of the comparison algorithm. The max prediction accuracy of the IPSO-CNN-LSTM model was 98.51%, which was 7.96%, 14.26%, 5.76%, and 3.82% higher than the maximum values of CNN-LSTM, SVM, PSO-CNN-LSTM, and combination model 3, respectively. After deploying the combination prediction model on the embedded AI platform, its time consumption increased by a maximum of 47.76% without significantly affecting the model accuracy. Overall, the designed combination prediction model has good performance. However, the IPSO algorithm designed in this study shows a decline in performance when facing complex multimodal problems. Future research can further optimize it to enhance its ability to overcome local optima.

Funding information

This work was sponsored in part by "14th Five-Year Plan" Key First-Tier Discipline of Business Administration in Jiangsu Province (SJYH2022-2/285).

References

[1] Ducharme C, Agard B, Trépanier M. Improving demand forecasting for customers with missing downstream data in intermittent demand supply chains with supervised multivariate clustering. Journal of Forecasting, 2024, 43(5):1661-1681. https://doi.org/10.1002/for.3095

[2] Karthick B, Uthayakumar R. An optimal strategy for forecasting demand in a three-echelon supply chain system via metaheuristic optimization. Soft Computing, 2023, 27(16):11431-11450. https://doi.org/10.1007/s00500-023-08290-x

[3] Bai B. Acquiring supply chain agility through information technology capability: the role of demand forecasting in retail industry. Kybernetes, 2023, 52(10):4712-4730. https://doi.org/10.1108/K-09-2021-0853

[4] Subramanian B, Mishra A, Bharathi V R, Mandala G, Kathamuthu N D, Srithar S. Big data and fuzzy logic

for demand forecasting in supply chain management: A data-driven approach. Journal of Fuzzy Extension and Applications, 2025, 6(2):260-283. https://doi.org/10.22105/jfea.2025.488816.1703

[5] Mbonyinshuti F, Nkurunziza J, Niyobuhungiro J, Kayitare E. Health supply chain forecasting: a comparison of ARIMA and LSTM time series models for demand prediction of medicines. Acta Logistica, 2024, 11(2):269-280. https://doi.org/10.22306/al.v11i2.510

[6] Liu R, Vakharia V. Optimizing supply chain management through BO-CNN-LSTM for demand forecasting and inventory management. Journal of Organizational and End User Computing (JOEUC), 2024, 36(1):1-25. https://doi.org/10.4018/JOEUC.335591

[7] Saad N H M. A study of deep learning-based algorithms for supply chain logistics demand forecasting. Edelweiss Applied Science and Technology, 2025, 9(3):1640-1654. https://doi.org/10.55214/25768484.v9i3.5650

[8] Zhang C, Zhang H, Pu T, Pan J. Supply Chain Demand Forecasting Based on Data Mining Algorithm and Seq2Seq. International Journal of Control, Automation and Systems, 2025, 23(1):89-104. https://doi.org/10.1007/s12555-024-0141-8

[9] Hasan M D R, Islam M R, Rahman M A. Developing and implementing AI-driven models for demand forecasting in US supply chains: A comprehensive approach to enhancing predictive accuracy. Edelweiss applied science and technology, 2025, 9(1):1045-1068. https://doi.org/10.55214/25768484.v9i1.4308

[10] Liu Y, Guo X. Research on Influence Factors of Cooperatives Performance in the Agricultural Products Supply Chain Based on System Dynamics Malaysian E Commerce Journal. 2017; 1(2): 01-03. http//doi.org/10.26480/mecj.02.2017.01.03

[11] Newton M, Gravagne I, Jack D. Normalization and processing of rotational eddy current scans for layup characterization of CFRP laminates. Research in Nondestructive Evaluation, 2024, 35(6):337-355. https://doi.org/10.1080/09349847.2024.2406239

[12] Pal S, Roy A, Palaiahnakote S, Pal U. Adapting a swin transformer for license plate number and text detection in drone images. Artificial Intelligence and Applications. 2023, 1(3):129-138. https://doi.org/10.47852/bonviewAIA3202549

[13] Wang J, Li Y H, Wang D, Chai M. A reliability calculation method based on ISSA-BP neural network. International Journal of Structural Integrity, 2024, 15(6):1249-1267. https://doi.org/10.1108/IJSI-07-2024-0104

[14] Li L, Yang Y, Zhou T, Wang M. Data-Driven Combination-Interval Prediction for Landslide Displacement Based on Copula and VMD-WOA-KELM Method. Journal of Earth Science, 2025, 36(1):291-306. https://doi.org/10.1007/s12583-021-1555-3

[15] Vasundra S, Venkatesh D, Bella H K. Levy Flight Strategy-based Tasmanian Devil's Optimization and Autoencoder Approach for Intrusion Detection in Cloud Computing. International Journal of Intelligent Engineering & Systems, 2025, 18(2):548-561. https://doi.org/10.22266/IJIES2025.0331.40

[16] Liu Y. Remote Sensing Image Scene Classification Based on Convolutional Neural Networks. Informatica, 2025, 49(9):45-54. https://doi.org/10.31449/inf.v49i9.5912.

[17] Sung W T, Kang H W, Hsiao S J. Speech Recognition via CTC-CNN Model. Computers, materials & continua, 2023, 76(9):3833-3858. https://doi.org/10.32604/cmc.2023.040024

[18] Aggarwal D, Banerjee S. Forecasting of S&P 500 ESG index by using CEEMDAN and LSTM approach. Journal of Forecasting, 2025, 44(2):339-355. https://doi.org/10.1002/for.3201

[19] Xu H, Song L, Li Y, Zhang T, Shen J. Comparison of LSTM and QR models for predicting CPUE of albacore tuna in waters near the Cook Islands. Fisheries Science, 2025, 91(2):259-274. https://doi.org/10.1007/s12562-025-01851-z

[20] Hao Q, Huang H. Multi-hop Network Security Strategy Integrating ACO Algorithm and PSO Algorithm. Informatica, 2025, 49(17):81-94. https://doi.org/10.31449/inf.v49i17.7499

[21] Mihăilescu M, Stancu-Dumitru D, Teca A. On a Rayleigh-type quotient involving a variable exponent which depends on test functions. Archiv der Mathematik, 2025, 124(5):557-570. https://doi.org/10.1007/s00013-024-02097-4

Lightweight GCD-YOLOv5 for Real-Time Obstacle Detection in Tunnel Pipe Jacking Operations

Gang Qiao[1], Deyi Yu[1], Kun Liu[1], Qiuhan Zhao[1], Nan Yu[1], Ling Luo[2*]
[1]Power China Roadbridge Group Co., Ltd, Beijing, 100081, China
[2]School of Civil and Hydraulic Engineering, Chongqing University of Science and Technology, Chongqing, 401331, China
E-mail: luo89070989@163.com
[*]Corresponding author

Keywords: YOLOv5, lightweight, GCD, model pruning, knowledge distillation

Received: April 8, 2025

The identification of tunnel pipe jacking obstacles is usually carried out in harsh environments, and timely and accurate identification can avoid unnecessary economic and labor losses. However, the currently commonly used obstacle recognition models are complex in structure and have long recognition feedback times. Therefore, this study proposes a tunnel pipe jacking obstacle recognition model based on an improved You Only Look Once version 5. The model is optimized through pruning and knowledge distillation techniques to enhance its lightweight characteristics and accuracy. The experiments were conducted using the COCO dataset and a custom dataset consisting of 2667 tunnel shield tunneling obstacle images. The optimized model achieved an 88.6% reduction in the number of parameters, an 84.2% reduction in floating-point operations, a 62.5% reduction in memory usage, and a 90.1% reduction in response time. In real-world testing, the model achieved an accuracy of 94.0% and a processing speed of 75 frames per second (FPS), outperforming traditional YOLOv5 and other lightweight YOLOv5 variants such as M-YOLOv5, S-YOLOv5, PL-YOLOv5, and C-YOLOv5. Using evaluation metrics such as mean Average Precision (mAP), the proposed model demonstrated high efficiency and effectiveness in real-time obstacle recognition for tunnel construction. The model provides a new technological approach for safety management in tunnel construction while improving computational efficiency and maintaining high recognition accuracy.

Povzetek: Lahki model GCD-YOLOv5 z Ghost in Depthwise konvolucijami, obrezovanjem in prenosom znanja omogoča bolj kvalitetno zaznavanje ovir v tunelskem potiskanju cevi v realnem času kot klasični YOLOv5 ali njegove lahke različice.

1 Introduction

Tunnel pipe jacking construction technology is a commonly used non-excavation pipeline installation technique. It has advantages such as minimal environmental impact, high construction efficiency, and safety, making it widely used for underground municipal pipeline installation [1]. However, during tunnel excavation using pipe jacking technology, unforeseen obstacles often occur, which can affect construction progress, safety, and other aspects. Without accurate identification of obstacle types and appropriate handling, significant financial and material losses may occur [2, 3]. Therefore, the correct identification of obstacles during tunnel pipe jacking construction is of great importance. Currently, the main obstacle identification methods for tunnel construction include intelligent algorithm-based visual analysis, radar and multi-sensor fusion, and automated inspection robots. However, these methods require high-performance equipment and have long recognition times, which hinder timely emergency responses [4-6]. Therefore, there is an urgent need for methods that can rapidly and accurately identify obstacles in harsh construction environments. The key to obstacle identification in tunnel pipe jacking construction lies in accurately judging the size and type of obstacles. As a visual intelligence analysis algorithm, You Only Look Once version 5 (YOLOv5) can quickly and accurately analyze the multi-dimensional information of obstacle images. After being lightweight-processed, it can adapt to low-performance devices at construction sites [7]. This study innovatively optimizes YOLOv5 by replacing its overall network structure to build a tunnel pipe jacking obstacle recognition model, combining model pruning and knowledge distillation techniques for lightweight processing. This study aims to develop and validate an obstacle recognition model based on YOLOv5, which is optimized using Ghost Convolution (Ghost Conv) and Depthwise Convolutions (GCD), model pruning, and knowledge distillation, with real-world validation on tunnel construction images. Through these optimization methods, this study aims to enhance the model's computational efficiency and recognition accuracy to meet

real-time obstacle detection requirements in practical tunnel construction.

2 Related works

YOLOv5, with its powerful object detection and image recognition capabilities, is an upgraded version of the YOLO. It offers higher precision and faster detection speed than previous versions. Scholars worldwide have applied YOLOv5 in various fields of visual recognition. For example, Gao et al. applied YOLOv5 to detect honeycomb structures and proposed an improved YOLOv5-based honeycomb detection model using Shuffle BlockV2. This model is 92.5% faster than previous honeycomb detection models, achieves an accuracy of 96% in real-time detection, and can automatically track honeycombs within 2 seconds up to a distance of 2.5 meters [8]. Xiao's team proposed a YOLOv5-based zinc-coated steel defect detection model for metal smelting. The results of actual steel-making tests showed that the model can quickly and accurately detect surface defects of galvanized steel, and its overall performance outperforms most detection models based on other mainstream algorithms [9]. To address issues of low accuracy and high computational delays in video surveillance for ship detection, Zheng's team proposed an improved YOLOv5-based algorithm. The accuracy of the YOLOv5 model, after compression using a scaling factor, improved by 2.34%, and the detection speed reached 20 FPS, even in low-computational environments [10]. In agriculture, Chen et al. applied YOLOv5 for real-time strawberry disease detection. By incorporating the Ghost Convolution module, YOLOv5 reduced the number of parameters and floating-point operations, enhancing the model's spatial information. After adding a convolutional attention module, the model's ability to extract feature data and suppress irrelevant information was improved. The model achieved an average precision of 94.7% in test experiments [11]. For forest fire detection via visual network analysis, Zhou et al. proposed a lightweight target detection model based on YOLOv5. The model used MobileNetV3 as the backbone network within the YOLOv5 framework and was trained using semi-supervised knowledge distillation. As a result, the model size decreased by 94.1%, and the average accuracy increased by 2.6% [12].

Regarding tunnel construction obstacle recognition, several mature methods and theories have been developed and applied in practical construction. For instance, Xu et al. used drones equipped with cameras to perform automatic surface detection of tunnels to be constructed and generated target 3D shapes using a motion structure assembly line for dynamic analysis. The results showed that this method can accurately identify surface obstacles before tunnel construction [13]. Yongcan et al. addressed the issue of long identification times and high safety risks in traditional underwater tunnel obstacle recognition by proposing a robotic method that uses multi-sensor fusion for comprehensive and accurate risk analysis and evaluation of underwater tunnel construction [14]. Naranjo's team has applied intelligent algorithms to obstacle recognition and clearance in actual tunnel construction. They developed an automated obstacle recognition and clearance system for tunnel construction trucks. This system successfully identified and cleared tunnel obstacles in multiple modes, including manual and remote operation, during tests by two civil construction companies [15]. Robots with cameras are also frequently used for tunnel obstacle recognition. Li et al. reduced redundant parameters in YOLOv5 by pruning and integrated it with intelligent robots to create a high-performance tunnel fault detector. In obstacle recognition tests, the system achieved an accuracy of 81.4% and a recall rate of 98.0% [16]. Image recognition technology has also been applied to obstacle detection, typically by combining algorithms with light field cameras or other sensors to analyze feature information from images for target identification. Zhang et al. proposed an automatic classification model for tunnel rock obstacles to quickly determine rock types during actual construction. The model, based on deep convolutional neural networks trained using residual learning methods, uses extended blocks in the deep convolutional network to extract multi-scale image features. This model outperformed previous rock obstacle classification models in terms of accuracy, recall rate, and computation time [17]. The summary of the above research is shown in Table 1.

Table 1: Summary table of related works

References	Method	Contribution	Limitations
[8]	YOLOv5 with Shuffle BlockV2 for honeycomb detection	Increased speed by 92.5%, accuracy 96%, and can track honeycombs at 2.5m in 2 seconds	Limited to honeycomb structures
[9]	YOLOv5 for galvanized steel defect detection	Fast and accurate defect detection, outperforming other methods	Limited to galvanized steel
[10]	Improved YOLOv5 with scaling factor	Improved accuracy by 2.34%, 20 FPS in low computational environments	Performance bottlenecks in complex scenes
[11]	YOLOv5 for real-time strawberry disease detection	Ghost convolution and attention modules, 94.7% accuracy	Limited to strawberry disease detection
[12]	YOLOv5 with MobileNetV3 for forest fire detection	94.1% reduction in size, 2.6% accuracy improvement	Relies on semi-supervised learning, data quality sensitive
[13]	Drone-based surface	Accurate obstacle detection before	Limited adaptability to

	detection for tunnels	construction	dynamic environments
[14]	Multi-sensor fusion for underwater tunnel obstacle detection	Enhanced risk analysis and evaluation for underwater construction	Needs further optimization for complex underwater environments
[15]	Automatic obstacle detection and removal system for tunnel trucks	Successful multi-mode obstacle recognition and removal	Limited application to specific construction environments
[16]	Drone-based obstacle detection and removal	YOLOv5 pruning with high accuracy (81.4%) and recall (98.0%)	Limited to specific obstacles and scenes
[17]	Deep CNN-based tunnel rock classification model	Improved accuracy and recall, faster computation	Limited to tunnel rock classification

In summary, although there has been progress in tunnel pipe jacking obstacle recognition, the methods used generally require complex technical equipment and are typically applied before construction, making them less effective for handling unexpected situations during actual construction. YOLOv5, with its fast speed, high accuracy, and low equipment performance requirements after lightweight optimization, presents an opportunity. This study proposes a tunnel pipe jacking obstacle recognition model based on lightweight YOLOv5, aiming to quickly and accurately identify obstacle types and meet the recognition needs at construction sites in practical applications.

3 Construction and optimization of the tunnel pipe jacking obstacle recognition model based on YOLOv5

3.1 Network lightweighting and model construction based on YOLOv5

YOLOv5 makes significant modifications to the original YOLO in data preprocessing and grid structures. These modifications enhance its image adaptability, detection accuracy, and generalization ability, improving its performance in visual information processing [18]. The key to obstacle recognition in tunnel pipe jacking construction is speed and accuracy. YOLOv5 can accurately predict target categories with a single forward pass. Therefore, this study leverages the advantages of YOLOv5 to construct an obstacle recognition model for tunnel pipe jacking construction. The network structure of YOLOv5 is shown in Figure 1.

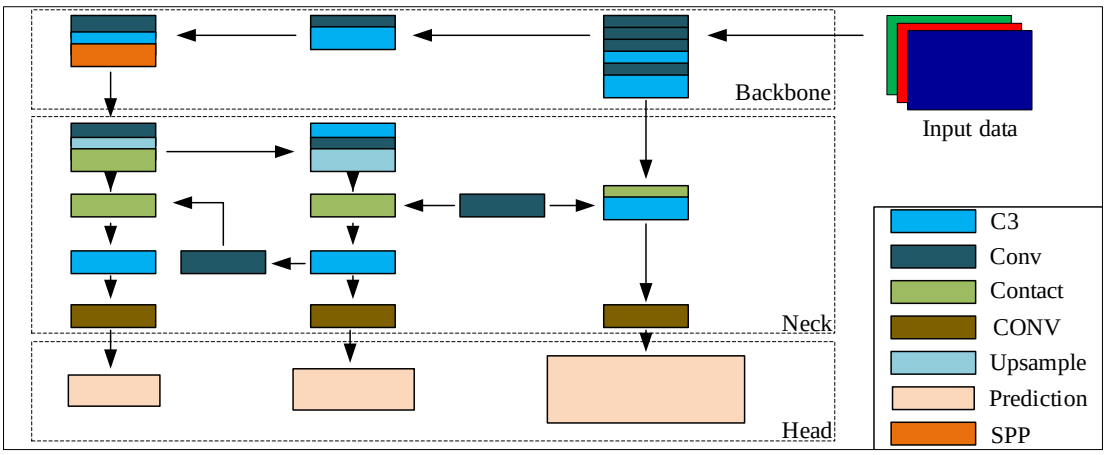


Figure 1: YOLOv5 network structure

Figure 1 presents the overall network architecture of YOLOv5. The model utilizes the Cross Stage Partial (CSP) network as the backbone and combines C3 modules and convolutional modules for feature extraction. YOLOv5 is structured into three main parts: the backbone network, the neck network, and the head network. The backbone network is responsible for extracting basic features from the input image, while the Spatial Pyramid Pooling (SPP) module further enhances feature extraction capabilities. Through residual connections and convolution operations, the backbone network fuses multiple grid cells into a more complete feature map, which is then processed in the head network for object classification and location regression. This structure ensures that YOLOv5 maintains high efficiency while performing precise object detection. YOLOv5 employs the GIOU loss function to optimize the bounding box regression, as defined in Equation (1).

$$GIoU = IoU - \frac{|A_c - U|}{|A_c|} \quad (1)$$

In Equation (1), A_c, IoU, and U represent the minimum enclosing rectangle area of the detected and ground truth boxes, the Intersection Over Union (IoU), and the union area, respectively. YOLOv5s adopts CSP as

its backbone network. However, CSP networks contain numerous C3 and CONV modules, leading to excessive redundant channels and data, which increases the overall computational and parameter complexity of YOLOv5 [19]. In real-world tunnel pipe jacking construction, a lightweight and fast network structure is necessary for obstacle recognition. To achieve this, this study replaces the CSP network with a lightweight alternative while ensuring network stability. The proposed solution employs a Depthwise Convolution (DWC) network to achieve inverse residual concatenation in a Ghost Convolution (GC) network, forming the Ghost Convolution-Depthwise Convolution (GCD) network as a replacement. The construction process of the GCD network is shown in Figure 2.

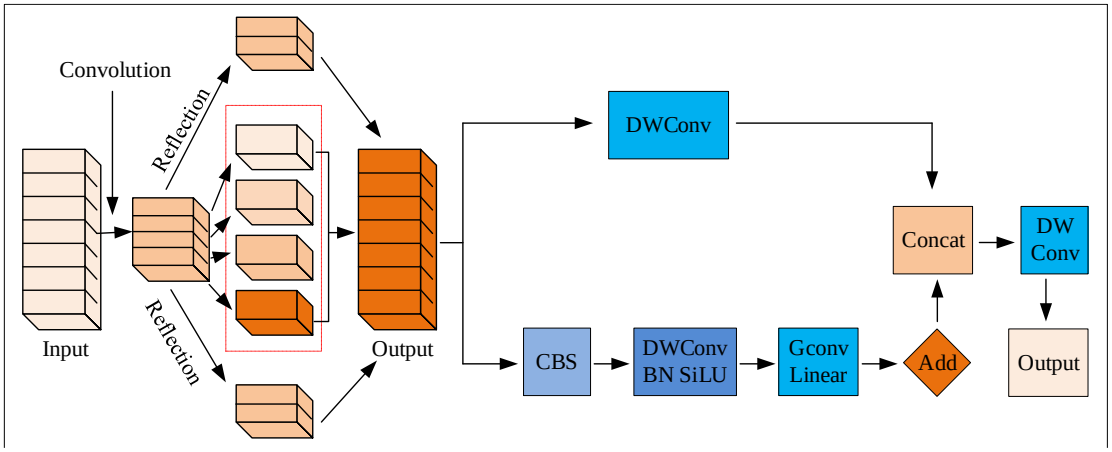


Figure 2: Flowchart of constructing a phantom convolutional depthwise separable network

In Figure 2, the core part of the GC network is implemented through the DWC layer and the inverse residual concatenation. Specifically, the red box in the figure indicates the implementation method of residual connections in the network. In this part, the input feature map first undergoes several layers of convolution operations (such as CBS and DWC), which extract image features and perform feature processing. Next, through the addition operation, the input feature map is added to the output feature map of the convolutional layer. This is the core process of inverse residual joining. Through this operation, the network can retain the information of the input features while avoiding information loss during the feature extraction process. Finally, the output feature map is concatenated with other features through the Concat operation, and further processed through the DWC and Linear layers to obtain the final output. In the GC structure, the Ghost Convolution process applies multiple convolution operations on the original image to extract real feature maps while generating ghost feature maps. The calculation process is defined in Equation (2).

$$y_{ij} = \phi_{ij}(y'_i), \Lambda_j = 1,...,s \quad (2)$$

In Equation (2), y_{ij} represents the generated ghost feature map, y'_i denotes the feature in the ghost layer, and ϕ_{ij} represents the i -th layer's j -th linear operation. s is the maximum number of feature layers. When performing linear combinations on real feature maps, the convolution ratio between layers is shown in Equation (3).

$$r_s = \frac{n*c*k*k*h'*w'}{n/s*h'*w'*k*k+(s-1)*n/s*h'*w'*d*d} = \frac{s*c}{s+c-1} \approx s \quad (3)$$

In Equation (3), n represents the number of parameters, h', w', and k represent the height, width, and depth of the final feature map, respectively. Finally, the ghost and real feature maps are linearly combined, with the parameter convolution ratio during computation shown in Equation (4).

$$r_c = \frac{n*c*k*k}{n/s*c*k*k+(s-1)*n/s*d*d} \approx \frac{s*c}{s+c-1} \approx s \quad (4)$$

In Equation (4), r_c represents the final scaling factor. Part of the feature maps undergoes depthwise separable convolution after GConv processing, while the rest are processed through Conflict-Based Search (CBS) in a Batch Normalization (BN) and SiLU-optimized DWConv network for inverse residual learning. These two sets of feature maps are then fused in the connection layer. Replacing the CSP network in YOLOv5 with the GCD network results in the GCD-YOLOv5 model for tunnel pipe jacking obstacle recognition, as shown in Figure 3.

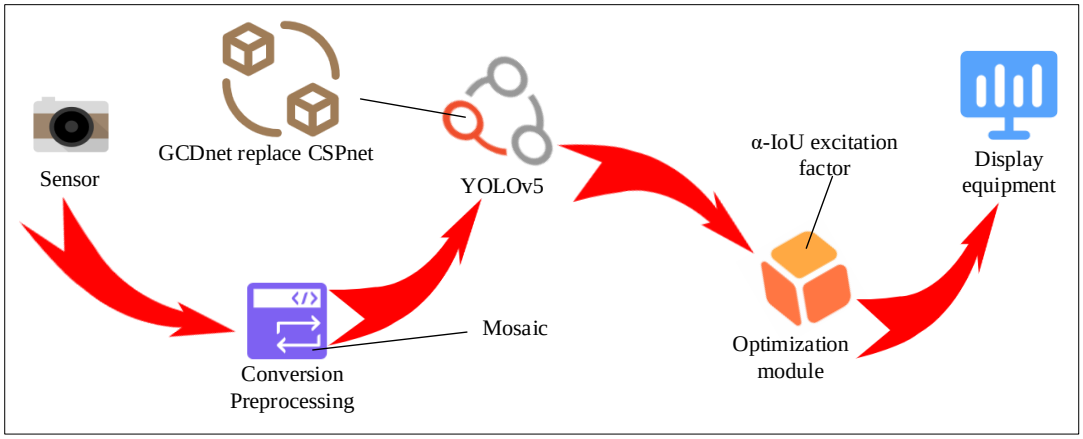


Figure 3: GCD-YOLOv5 obstacle recognition model operation flow chart

As shown in Figure 3, the model first captures obstacle images using cameras or other sensors. Before entering the GCD-YOLOv5 network, Mosaic preprocessing is applied, including image scaling, cropping, and concatenation. The preprocessed images are then fed into the GCD-YOLOv5 network for convolution processing. The processed data is refined using the a-IOU function, as defined in Equation (5).

$$L_{a-DIoU}(A,B) = 1 - IOu(A,B)^{a} + \frac{\rho^{2a}(A,B)}{c^{2a}} \quad (5)$$

In Equation (5), L_{a-DIoU} represents the loss function incorporating a-IOU into the original GIOU design. A and B denote the target box and predicted box sizes, respectively. ρ is the Euclidean distance between the centers of the two boxes, while c represents the diagonal length of the minimum enclosing rectangle containing both boxes. The optimized feature maps serve as the model's final prediction results and are displayed on output devices. For tunnel pipe jacking obstacle recognition, the model promptly identifies obstacle types and provides relevant information.

2.2 Optimization of GCD-YOLOv5 using model pruning and knowledge distillation

The GCD-YOLOv5 model achieves lightweight optimization through network structure replacement, improving accuracy. However, further lightweight optimization is needed for real-world tunnel pipe jacking construction. Network pruning assigns different weights at the channel, kernel, and layer levels, reducing the model's memory footprint without additional training [20]. The network pruning process for the GCD-YOLOv5 model is shown in Figure 4.

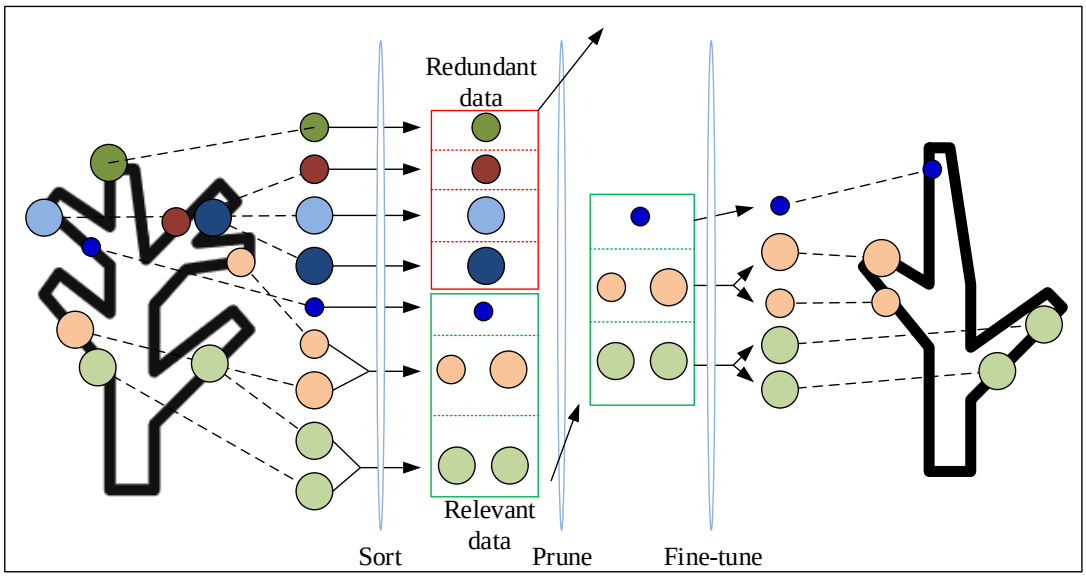


Figure 4: Schematic diagram of network pruning process

As shown in Figure 4, network pruning integrates data from each channel in the GCD network, classifying it as either feature or non-feature data. Only feature data is retained and reassigned to new network channels. However, uncontrolled channel data during pruning may result in mismatched sizes and distributions between predicted and target images, significantly reducing accuracy [21]. To address this issue, BN is applied to align

channels, as defined in Equation (6).

$$x'_i = \frac{x_i - E[x]}{\sqrt{Var[x]}} \quad (6)$$

In Equation (6), x'_i represents the final data, x_i is the input data, $E[x]$ is the mean, and $Var[x]$ is the variance. To prevent data homogenization, the scaling factor γ and offset β are applied to adjust the output data, as shown in Equation (7).

$$y_i = \gamma x_i + \beta \quad (7)$$

In Equation (7), y_i represents the scaled output data. Each convolution channel, depending on γ, switches between open and constrained states, with constrained channels being pruned. During the specific pruning process of the model, the study set that 30% of the neurons or channels would be pruned each time during the pruning operation to ensure an effective reduction in the network size while maintaining good model performance. Pruning decisions are based on the weight magnitude of each neuron or channel. A minimum weight threshold of 0.005 was set. Neurons or channels below this threshold will be pruned to ensure that the pruning process has a relatively small impact on the model output. The pruning strategy adopts channel-based pruning, that is, pruning by channel, which can reduces redundant calculations more precisely. To ensure the validity of the model, a 5% performance tolerance is set. That is, the accuracy of the model after pruning shall not decrease by more than 5%. If the accuracy decrease caused by pruning exceeds this threshold, the pruning operation will be revoked. The pruned model has fewer channels and parameters, leading to lower accuracy. Knowledge distillation transfers information from a trained large model to a smaller model to improve accuracy [22]. Thus, the unpruned GCD-YOLOv5 model serves as the teacher model for knowledge distillation, as shown in Figure 5.

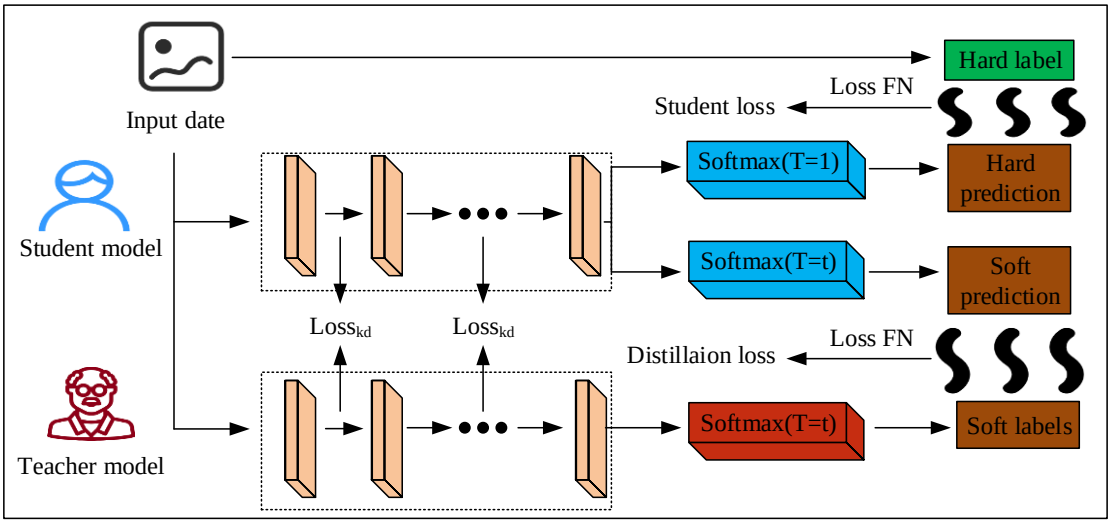


Figure 5: Schematic diagram of the knowledge distillation process

As shown in Figure 5, knowledge distillation requires both the teacher and student models to process the same input data and extract similar feature sets. Weighted cross-entropy loss addresses the difference in feature weighting between the two models. To enhance classification ability, the study employs a Region Proposal Network (RPN) and a Recursive Cortical Network (RCN) for learning. To maintain the lightweight feature of YOLOv5, the introduction of RPN and RCN is limited to the training stage, serving as knowledge transfer tools between the teacher model and the student model. Specifically, RPN is used to generate "soft" candidate regions (i.e., candidate box proposals), which are provided by the teacher model and passed to the student model as auxiliary information during the training process to help it learn the spatial distribution of the target region and thereby optimize the classification ability. Similarly, RCN enhances the classification ability in the student model, extracts more distinguishable features through recursive learning, and improves the recognition ability for multi-category targets. It should be noted that RPN and RCN do not directly participate in the inference process of YOLOv5. They only act on the student model during the training stage, helping the student model improve detection accuracy and classification ability through knowledge distillation without affecting the inference speed. Therefore, although these two components are different from the traditional YOLOv5 architecture, they ensure through auxiliary optimization that the model can achieve high object detection accuracy and classification ability while maintaining lightweight. The definitions of the two are shown in Equation (8).

$$\begin{cases} LRCN = \frac{1}{N}\sum_i L_{cls}^{RCN} + \lambda \sum_j L_{reg}^{RPN} \\ LRPN = \frac{1}{M}\sum_i L_{cls}^{RPN} + \lambda \frac{1}{M}\sum_j L_{reg}^{RPN} \end{cases} \quad (8)$$

In Equation (8), N and M represent RCN and RPN processing numbers, respectively. λ is the scaling factor, L_{cls} represents the loss between the student model's predicted and actual values, and L_{reg} represents

the loss between the student and teacher models. The final student model output loss is given in Equation (9).

$$L = L_{RCN} + L_{RPN} + \lambda L_{H\,\mathrm{int}} \quad (9)$$

In Equation (9), $L_{H\,\mathrm{int}}$ is the correction function based on hidden layers. L_{cls}, L_{reg}, and $L_{H\,\mathrm{int}}$ continuously adjust the student model to match the teacher model's predictions. During training, when the student model outperforms the teacher model, the results are corrected, as shown in Equation (10).

$$L_{reg} = L_{sL1}(R_s y_{reg}) + vL_b(R_s R_t y_{reg}) \quad (10)$$

In Equation (10), y_{reg} represents ground truth labels, v is the weight coefficient, R_s and R_t are the student and teacher model outputs, L_b represents teacher regression constraints, and L_1 is the smooth loss between student predictions and actual results. When evaluating final outputs, hidden layer distances are used for representation, as defined in Equation (11).

$$L_{H\,\mathrm{int}}(VZ) = \left\| V - Z \right\|_2^2 \quad (11)$$

In Equation (11), V and Z represent the outputs of the guided student model and intermediate layers of the teacher model, respectively. The final student model output can also be expressed using smooth loss on L_1, as shown in Equation (12).

$$L_{H\,\mathrm{int}}(VZ) = \left\| V - Z \right\|_1 \quad (12)$$

In Equation (12), $L_{H\,\mathrm{int}}$ measures the quality of the final output. The trained student model, despite having fewer parameters than the teacher model, achieves high recognition accuracy. Ultimately, the lightweight GCD-YOLOv5 model, optimized through pruning and knowledge distillation, is shown in Figure 6.

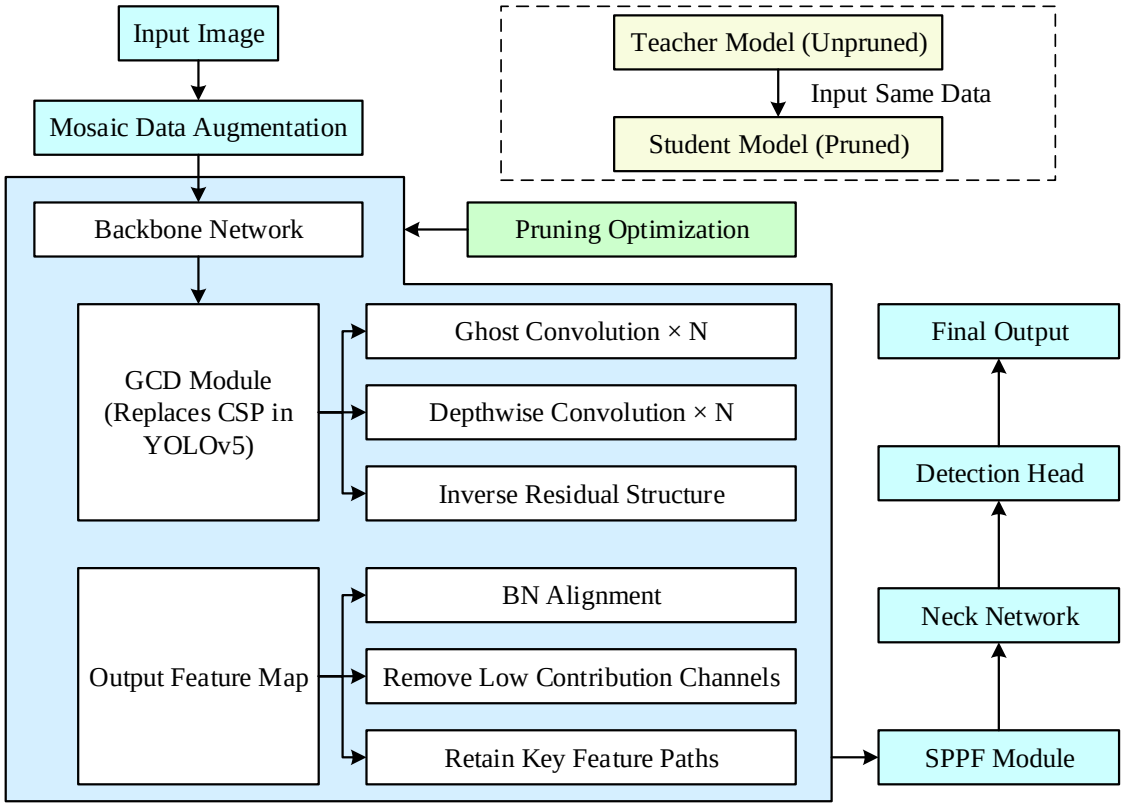


Figure 6: The lightweight GCD-YOLOv5 model

In Figure 6, the structure of the final lightweight GCD-YOLOv5 model improves computational efficiency and recognition accuracy through a series of optimization measures. This model first adopts Ghost convolution and deep convolution to replace the CSP network in YOLOv5, reducing computational complexity while maintaining the efficient feature extraction capability. Then, pruning optimization ensures the removal of redundant channels and convolutional layers without affecting performance, and stability after pruning is guaranteed through BN alignment. The optimized model transfers the knowledge of the teacher model to the student model through knowledge distillation, which can further improve the accuracy rate. Therefore, the GCD-YOLOv5 model after these optimizations can effectively adapt to the requirements of rapid object detection in actual scenarios such as tunnel pipe jacking construction.

4 Validation of the pipe jacking obstacle recognition model based on lightweight YOLOv5

4.1 Performance validation of the obstacle recognition model

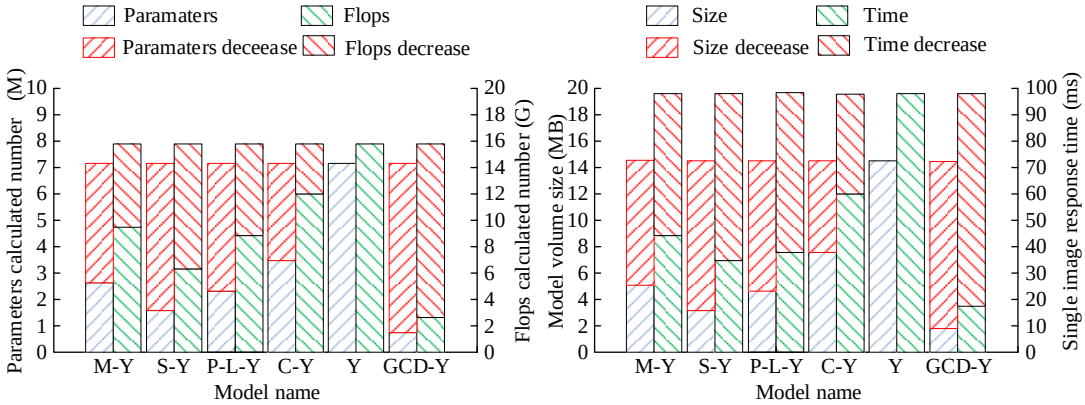
To evaluate the performance of the lightweight GCD-YOLOv5 model, the study tested the model using the Common Objects in Context (COCO) dataset. This dataset

is a standard dataset widely used for object detection, instance segmentation, and keypoint detection. It contains 80 object categories and hundreds of thousands of images, providing detailed annotation information such as bounding boxes and semantic segmentation masks. During training, the number of epochs was set to 50, the batch size to 32, the learning rate to 0.001, and the Adam optimizer was used. To ensure the reliability and robustness of the results, all models were independently trained five times, and the average of the results was taken to reduce the influence of accidental factors. In addition, data augmentation techniques such as random clipping, rotation and horizontal flipping were adopted during the training process to improve the generalization ability and adaptability of the model. The M-YOLOv5 model based on MobileNetV2, the S-YOLOv5 model based on ShuffleNetV2, the P-L-YOLOv5 model based on PP-LCNet, the C-YOLOv5 model based on C3Ghost, and the original YOLOv5 model were used as comparisons. Among them, M-YOLOv5 adopts MobileNetV2 as the backbone network to reduce the computational calculation [23]. S-YOLOv5 is based on ShuffleNetV2, further reducing the computing resource consumption and was suitable for low-computation environments [24]. PL-YOLOv5 adopts PP-LCNet to optimize the backbone network and improve accuracy and efficiency [25]. C-YOLOv5 combines the C3Ghost backbone network and reduces memory usage and computational complexity through techniques such as pruning [26]. The original YOLOv5 serves as the benchmark model, providing comparisons with other optimized variants. Through the comparison of these variants, the impact of different optimization strategies on the model performance can be comprehensively evaluated, helping to select the most suitable scheme for the obstacle identification task in tunnel construction. The test configuration is shown in Table 2.

Table 2: Specific configuration for the experiment

Hardware environment		Software environment	
Test environment	Specifications	Test environment	Specific parameter
CPU	Intel(R) Xeon(R) Silver4215	CUDA	10.2
GPU	NVDIA GTX 1080Ti	CUDNN	8.0.4
Operating system	Windows11	Programming language	Python
Memory size	64G	Edition	Anaconda 3-5.2.0

All models were tested under the same hardware and software environment. To verify whether the proposed model had advantages in lightweight characteristics, the floating-point operations and model size of each model were compared. The results are shown in Figure 7.



(a) Each model parameter and floating-point size diagram (b) Each model size and response time diagram

Figure 7: Comparison of lightweight indicators of various models

As shown in Figure 7(a), the proposed GCD-YOLOv5 model has the lowest number of parameters and floating-point operations, with only 0.8M±0.02M parameters and 2.5G±0.1G FLOPs, significantly outperforming the original YOLOv5 model and other optimized variants. Compared to the unoptimized model, the number of parameters was reduced by an average of 88.6% (95% CI: [87.9%, 89.3%]), and the FLOPs were reduced by 84.2% (95% CI: [83.5%, 84.8%]), confirming the model's efficiency in memory utilization during runtime. As shown in Figure 7(b), among the six models, GCD-YOLOv5 achieved the smallest memory footprint at 1.8MB±0.05MB and the shortest response time at 19ms±0.8ms per image. Compared to the original YOLOv5, memory usage was reduced by 62.5% and response time by 90.1%. All models were tested across

five independent runs, and paired t-tests were applied to response time and memory usage. Results confirmed that the proposed model demonstrates statistically significant improvements in both metrics ($p < 0.01$), validating its lightweight performance advantages. To further evaluate the model's performance, the study analyzed the relationship between training epochs and accuracy, as shown in Figure 8.

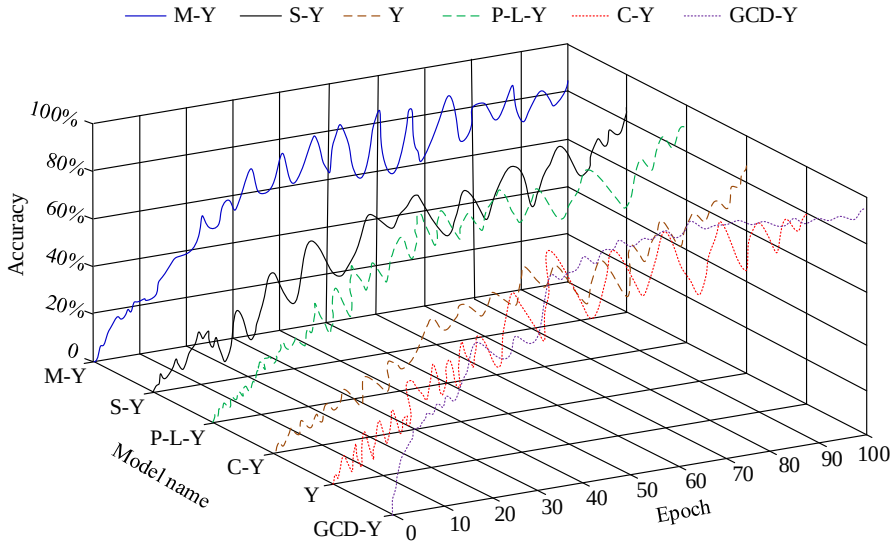
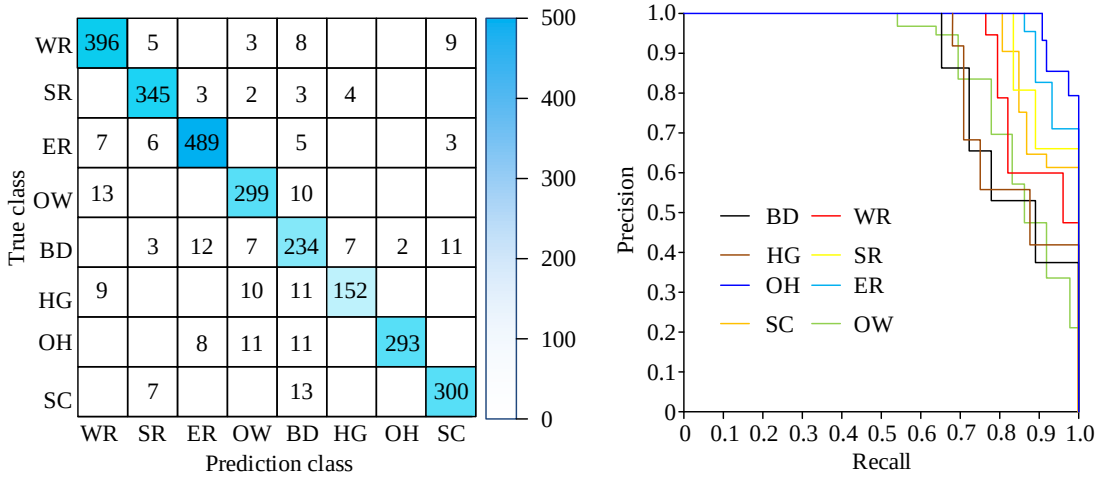


Figure 8: Comparison of accuracy changes

As shown in Figure 8, the accuracy of all models increases with the number of training epochs. At epoch 10, the accuracy of M-YOLOv5, S-YOLOv5, PL-YOLOv5, C-YOLOv5, and the original YOLOv5 was 20.2%, 18.4%, 18.6%, 13.1%, and 18.2%, respectively. In contrast, the proposed GCD-YOLOv5 achieved 32.5%±1.2% accuracy at the same stage, significantly outperforming the others, indicating faster learning capability. After 100 epochs, the final accuracies were 87.0%±0.6% (M-YOLOv5), 86.2%±0.7% (S-YOLOv5), 90.1%±0.5% (PL-YOLOv5), 89.5%±0.6% (C-YOLOv5), and 80.2%±0.8% (YOLOv5), while the proposed GCD-YOLOv5 achieved the highest at 93.5%±0.4%. The 95% confidence interval for the GCD-YOLOv5 final accuracy was [92.9%, 94.1%], and paired t-tests showed statistically significant differences compared with all other models ($p < 0.01$). These results demonstrate that the proposed model offers both faster convergence and higher accuracy.

3.2 Practical application of the obstacle recognition model

After verifying the lightweight characteristics and recognition performance of the GCD-YOLOv5 model, the study further tested its performance in real-world applications using 2,667 images of tunnel pipe-jacking construction obstacles collected from actual construction processes. To ensure the accurate annotation of the obstacle images during the tunnel pipe jacking construction, this study adopted the method of manual annotation. During the annotation process, the types of obstacles in each image were classified according to the actual situation and labeled using bounding boxes. The specific types of obstacles marked include broken and weak surrounding rocks, high-stress rock strata, expansive surrounding rocks, water and mud gushing, dangerous gases, rockburst and surrounding rock deformation, other obstacles and surface settlement and collapse. After marking, there were 425 broken and weak surrounding rocks, 366 high-stress rock strata, 512 expansive surrounding rocks, 332 water and mud gushing, 254 dangerous gas gushing, 163 rockburst and surrounding rock deformation, 295 other obstacles, and 320 surface settlement and collapse. The annotation process was carried out by two independent annotators, and Cohen's Kappa was adopted to measure the consistency among the annotators. The results show that the Cohen's Kappa value between the two annotators was 0.92, indicating high consistency and that the annotation results were reliable and accurate. To enhance the authenticity of the test results and adaptability to the actual construction environment, the study's test environment simulated various condition changes that may occur during tunnel pipe jacking construction to ensure stable model application in harsh construction environments. To verify the accuracy and classification ability of the model, the confusion matrix and Precision-Recall (PR) curve of the correct recognition times of the GCD-YOLOv5 model were analyzed, as shown in Figure 9.



(a) Confusion matrix of each type of obstacle identification (b) Experimental results of PR curve

Figure 9: Confusion matrix and PR curve of the number of correct obstacle identifications

Based on the updated confusion matrix calculations, the model performs well in most categories, especially in Soft Rock (SR) and Expansive Surrounding Rock (ER), with accuracy rates of 96.6% and 95.9%, respectively, demonstrating strong classification ability in these categories. The accuracy for Surrounding Rock (WR) and Settlement Collapse (SC) was 94.1% and 93.8%, indicating stable and reliable classification performance in these categories as well. However, the accuracy for Rock Bursts and Surrounding Rock Deformation (BD) and High-Stress Rock Layers (HG) was relatively lower, at 87.0% and 83.5%, possibly due to the complexity or lower sample numbers of these categories, which leads to slightly weaker performance. Other Obstacles (OH) achieved an accuracy of 90.7%, showing stable classification results. Figure 9(b) shows the PR curves for different obstacle types. The area under the PR curve was the smallest for fractured weak surrounding rock at 0.834, while it was the largest for other obstacles at 0.975, demonstrating the model's robust classification capabilities. These results confirm that the proposed lightweight GCD-YOLOv5 model achieved high accuracy and robust classification performance in real-world applications. To further verify whether the proposed model outperformed other models in recognition accuracy, the study analyzed the number of correctly identified images for each model, as shown in Figure 10.

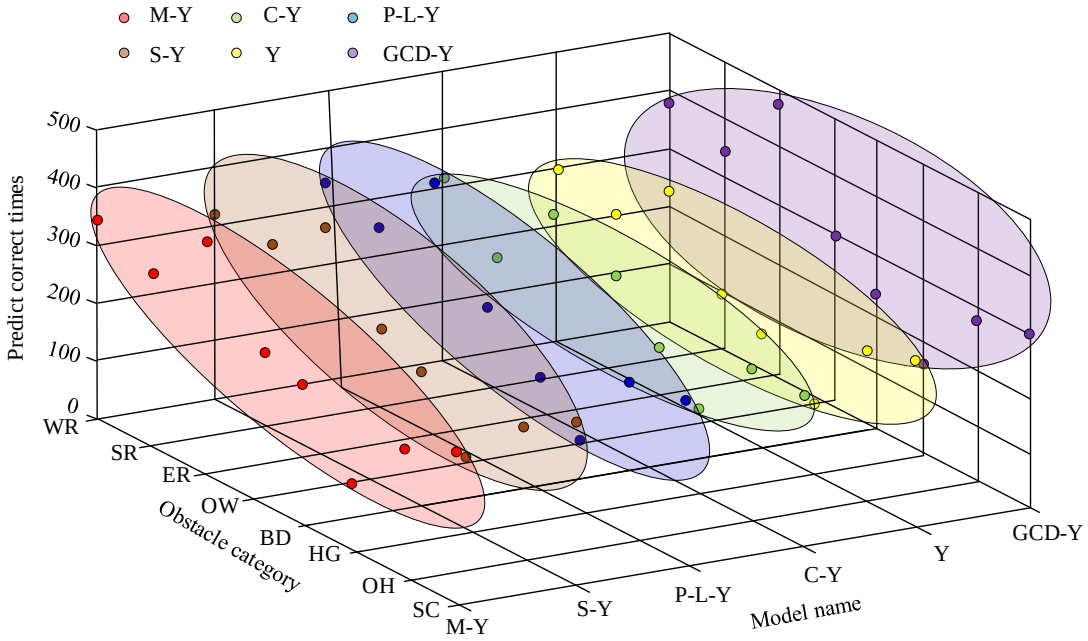


Figure 10: Statistics of the number of correct obstacle identifications

As shown in Figure 10, the GCD-YOLOv5 model correctly identified 2,498 images, achieving an accuracy of 94.0%. In comparison, the M-YOLOv5, S-YOLOv5, P-L-YOLOv5, and C-YOLOv5 models correctly identified 2,360, 2,336, 2,459, and 2,408 images, with accuracies of 88.5%, 87.6%, 92.2%, and 90.3%, respectively. All comparison models had lower accuracy than the proposed model. The original YOLOv5 model, without any

optimization, correctly identified only 2,150 images, achieving an accuracy of 80.6%, which was significantly lower than that of the proposed model. These results confirmed that the proposed model achieved higher accuracy than the other models in real-world applications.

To verify whether the model's efficiency was improved in practical applications, the study analyzed the Image Recognition Speed (FPS) of each model when processing obstacle images, as shown in Figure 11.

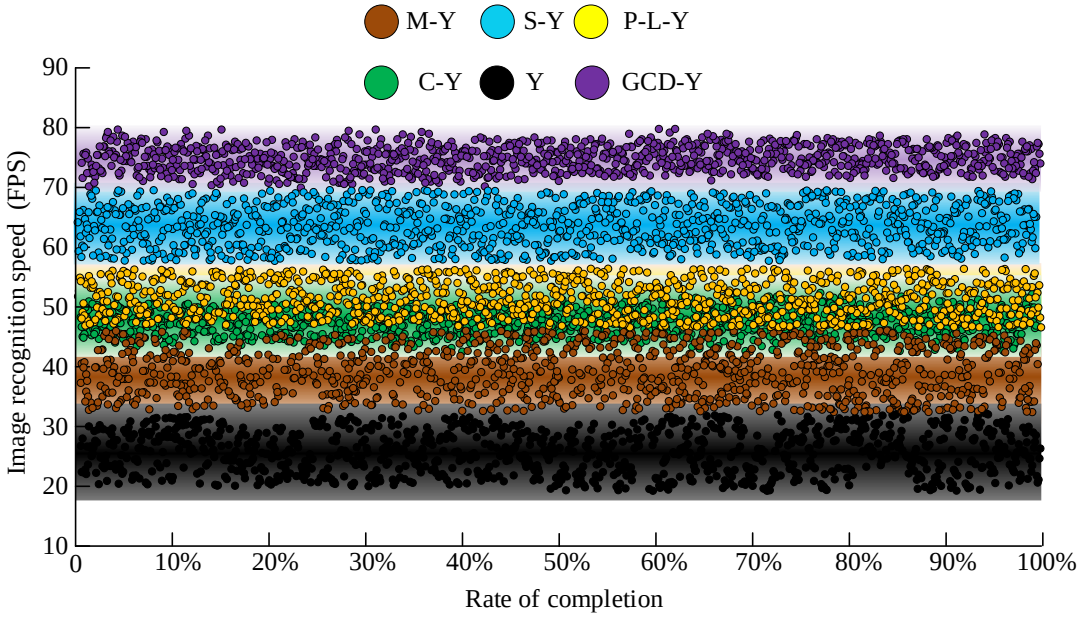


Figure 11: Comparison of image recognition speed and completion rate

As shown in Figure 11, when each model processes the obstacle images in actual construction, the FPS remains relatively stable. However, compared with the original YOLOv5 model, the FPS distribution of the proposed GCD-YOLOv5 model was narrower, indicating that it was more stable during operation. Specifically, the average FPS of the proposed model was 75, significantly higher than those of the other models. In contrast, the average FPS of the original YOLOv5 model when processing obstacle images was approximately 25 FPS, with the processing speed increasing by about 200%. To further verify the real-time performance of the model, the study decomposed the processing delay in detail. Among them, the image acquisition link took approximately 5ms, the preprocessing operation (such as image scaling and standardization) took approximately 10ms, the reasoning process took 15ms, and the post-processing (including non-maximum suppression) required approximately 4ms. Overall, the total processing delay was 34 ms, corresponding to 75 FPS, which proves that the proposed GCD-YOLOv5 model has high stability and real-time performance in actual scenarios. To comprehensively evaluate the detection performance of the GCD-YOLOv5 model, this study compares it with four other models: M-YOLOv5, S-YOLOv5, PL-YOLOv5, and C-YOLOv5, using the same test dataset. The evaluation metrics include mAP@0.5, mAP@0.5:0.95, Precision, Recall, and F1-score. The comparison results are shown in Table 3.

Table 3: Detection performance comparison of different models

Model	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Precision (%)	Recall (%)	F1-score (%)
M-YOLOv5	90.37	71.12	89.21	85.46	87.31
S-YOLOv5	88.64	69.78	87.86	84.02	85.88
PL-YOLOv5	91.43	73.36	90.68	86.81	88.67
C-YOLOv5	89.91	72.04	89.63	85.57	87.42
GCD-YOLOv5	93.62	76.24	92.41	89.38	90.42

As shown in Table 3, the GCD-YOLOv5 model achieved 93.62% in mAP@0.5 and 76.24% in mAP@0.5:0.95, outperforming all other models. For precision, recall, and F1-score, GCD-YOLOv5 also ranks the highest with values of 92.41%, 89.38%, and 90.42%, respectively. In comparison, the best-performing alternative model, PL-YOLOv5, achieved 91.43% in mAP@0.5, 73.36% in mAP@0.5:0.95, 90.68% in precision, 86.81% in recall, and 88.67% in F1-score. The other models show relatively lower results across most metrics. In addition, one-way ANOVA and paired t-tests were conducted on mAP@0.5 and F1-score across the five models. The statistical analysis indicates that GCD-YOLOv5 shows significant differences compared to the other models in both metrics ($p < 0.01$), confirming its advantage in overall detection performance.

5 Discussion

In this study, the performance of the GCD-YOLOv5

model was compared with C-YOLOv5, M-YOLOv5, and P-L-YOLOv5 on real-world tunnel images. The experimental results show that GCD-YOLOv5 outperforms the other models in both accuracy and processing speed. GCD-YOLOv5 achieved an accuracy of 94.0%, which was approximately 5.5%, 6.3%, and 3.5% higher than C-YOLOv5, M-YOLOv5, and P-L-YOLOv5, respectively. Furthermore, GCD-YOLOv5 operates at 75 FPS, significantly higher than the original YOLOv5 and other optimized models, demonstrating superior real-time processing capability. The superior performance of GCD-YOLOv5 can be attributed to its optimized architecture. We replaced the CSP network in YOLOv5 with Ghost Convolution and Depthwise Convolution networks. This architectural optimization effectively reduces computational load and parameter count while retaining efficient feature extraction capabilities, thereby improving both accuracy and speed. Additionally, GCD-YOLOv5 combines pruning with knowledge distillation during the training process, effectively mitigating the slight accuracy loss caused by pruning and ultimately achieving higher accuracy and smaller memory usage. In terms of model optimization, GCD-YOLOv5 adopts the a-IOU loss function and BN alignment techniques. These optimizations improve the model's stability and ensure that, even with significant pruning, the model can maintain high detection accuracy. Compared to other models, these innovative optimizations make GCD-YOLOv5 more suitable for real-world applications such as tunnel jacking, where rapid response and accurate obstacle detection are essential.

However, the advantages of GCD-YOLOv5 come with trade-offs. Although pruning and knowledge distillation optimize the model's computational efficiency, pruning in the early stages of training can lead to a slight drop in accuracy. Particularly in the initial epochs, large-scale pruning results in some accuracy loss. However, this loss was recovered through subsequent training and knowledge distillation, allowing the model to achieve a balance between accuracy and efficiency in the end. In conclusion, GCD-YOLOv5 achieved high accuracy while optimizing computational efficiency through architectural improvements, pruning, and knowledge distillation. While pruning may result in a slight decrease in accuracy in the early stages, it was eventually compensated for, making GCD-YOLOv5 highly advantageous in practical applications such as tunnel jacking construction, where real-time performance and accuracy were crucial, especially in resource-constrained environments.

6 Limitations and future research

Although the GCD-YOLOv5 model performs well on the GTX 1080Ti platform, it has not yet been tested on mobile-class GPUs or edge devices (such as Jetson boards), so the model's performance on resource-constrained devices has not been fully validated. Future research will focus on performance testing on mobile GPUs and Jetson platforms to ensure the model's adaptability and stability. Additionally, future work could explore hardware acceleration optimizations (such as TensorRT, OpenVINO) to improve inference speed on edge devices and consider transfer learning across different hardware platforms to enhance the model's cross-platform generalization. Furthermore, this study did not incorporate visualizations such as Grad-CAM due to the limitations of the current experimental environment, which does not support the extraction and processing of intermediate feature layers. Future work will consider integrating adaptable visualization techniques on more flexible testing platforms to enhance the model's interpretability and result clarity. Lastly, the model relies on well-lit and clear images, and environmental factors such as low lighting or complex backgrounds may affect detection accuracy. When facing new types of obstacles, additional training and optimization may be required. Therefore, future research could introduce a more diverse dataset to improve the model's robustness and explore transfer learning-based solutions to better generalize the model to unknown obstacle types.

7 Conclusion

To address the issues of model complexity and long processing time in obstacle recognition for tunnel pipe jacking construction, this study developed a lightweight model based on GCD-YOLOv5. The model replaced the CSP network with the GCD network and optimized output results using the a-IOU function. Additionally, pruning and knowledge distillation were applied to achieve model lightweighting. The test results showed that the proposed model achieved significantly higher accuracy than the unoptimized YOLOv5 model. The model had 0.8M parameters, 2.5G floating-point operations, a size of 1.8MB, and a response time of 19ms per image, all of which outperformed the compared models. In the recognition of actual tunnel pipe jacking construction obstacles, the model achieved an accuracy of 94.0% and a processing speed of 75 images per second, enabling fast and accurate obstacle classification in practical applications. In summary, the GCD-YOLOv5-based obstacle recognition model effectively met the lightweight requirements of harsh construction site conditions while maintaining considerable accuracy.

Fundings

The research is supported by: Power China Roadbridge Group Co., Ltd, Research on key technologies for construction safety management throughout the entire process of pipe jacking operation. (Scientific and technological breakthroughs project: ZGDJLQ-YX-2023-01-0001).

References

[1] Rayner-Philipson M, Sheil B, Zhang P. Prediction of pipe-jacking forces using a physics-constrained

neural network. Machine Learning and Data Science in Geotechnics, 2025, 1(1): 24-34. https://doi.org/10.1108/MLAG-06-2024-0004

[2] Jiahuan Wang, Haixiao Jia, Xuejiao Bai, et al. Research on the location of railway train in tunnel based on factor graph optimization. Applied Computer Letters, 2023, 7(1)

[3] Xu Z, Chen B, Zhan X, Xiu Y, Suzuki C, Shimada K. A vision-based autonomous UAV inspection framework for unknown tunnel construction sites with dynamic obstacles. IEEE Robotics and Automation Letters, 2023, 8(8): 4983-4990. https://doi.org/10.1109/LRA.2023.3290415

[4] ZHAO X, DENG K, ZHANG Y, MA Y, XIA Y. Molding quality inspection method for large-diameter shield tunnels. Chinese Journal of Engineering, 2024, 46(2): 365-375.

[5] Puspita E, Suryadi D, Rosjanuardi R. Learning obstacles of prospective mathematics teachers: A case study on the topic of implicit derivatives. Kreano, Jurnal Matematika Kreatif-Inovatif, 2023, 14(1): 174-189.

[6] Jiahuan Wang, Haixiao Jia, Peifen Pan, et al. Research on the technology of man-machine collision early warning system in tunnels based on bds high-precision positioning in tunnel. Applied Computer Letters, 2023, 7(1)

[7] Yusro M M, Ali R, Hitam M S. Comparison of faster r-cnn and yolov5 for overlapping objects recognition. Baghdad Science Journal, 2023, 20(3): 0893-0893. https://doi.org/10.21123/bsj.2022.7243

[8] Gao P, Lee K, Kuswidiyanto L W, Hu K., Liang G, Han X. Dynamic beehive detection and tracking system based on YOLO V5 and unmanned aerial vehicle. Journal of Biosystems Engineering, 2022, 47(4): 510-520. https://doi.org/10.1007/s42853-022-00166-6

[9] Xiao D, Xie F, Gao Y, Xie H. A detection method of spangle defects on zinc-coated steel surfaces based on improved YOLO-v5. The International Journal of Advanced Manufacturing Technology, 2023, 128(1-2): 937-951. https://doi.org/10.1007/s00170-023-11963-4

[10] Zheng J C, Sun S D, Zhao S J. Fast ship detection based on lightweight YOLOv5 network. IET Image Processing, 2022, 16(6): 1585-1593. https://doi.org/10.1049/ipr2.12432

[11] Chen S, Liao Y, Lin F, Huang B. An improved lightweight YOLOv5 algorithm for detecting strawberry diseases. IEEE Access, 2023, 11: 54080-54092. https://doi.org/10.1109/ACCESS.2023.3282309

[12] Zhou M, Wu L, Liu S, Li J. UAV forest fire detection based on lightweight YOLOv5 model. Multimedia Tools and Applications, 2024, 83(22): 61777-61788. https://doi.org/10.1007/s11042-023-15770-7

[13] Xu Z, Chen B, Zhan X, Xiu Y, Suzuki C, Shimada K. A vision-based autonomous UAV inspection framework for unknown tunnel construction sites with dynamic obstacles. IEEE Robotics and Automation Letters, 2023, 8(8): 4983-4990. https://doi.org/10.1109/LRA.2023.3290415

[14] Yongcan C, Jiajie C, Haoran W, Yue F, Zhaowei L, Hui X. Key technology of underwater inspection robot system for large diameter and long headrace tunnel. Journal of Tsinghua University (Science and Technology), 2023, 63(7): 1015-1031.

[15] Naranjo J E, Valle A, Cruz A, Martín M, Anguera M, García P, Jiménez F. Automation of haulers for debris removal in tunnel construction. Computer‐Aided Civil and Infrastructure Engineering, 2023, 38(14): 2030-2045. https://doi.org/10.1111/mice.12997

[16] Li Y, Bao T, Li T, Wang R. A robust real‐time method for identifying hydraulic tunnel structural defects using deep learning and computer vision. Computer‐Aided Civil and Infrastructure Engineering, 2023, 38(10): 1381-1399. https://doi.org/10.1111/mice.12949

[17] Zhang, W, Zhang W, Zhang G, Huang J, Li M, Wang X, Guan X. Hard-rock tunnel lithology identification using multi-scale dilated convolutional attention network based on tunnel face images.Frontiers of Structural and Civil Engineering, 2023, 17(12), 1796-1812. https://doi.org/10.1007/s11709-023-0002-1

[18] Alsuwaylimi A A, Alanazi R, Alanazi S M,Alenezi S M, Saidani T, Ghodhbani R. Improved and efficient object detection algorithm based on yolov5. Engineering, Technology & Applied Science Research, 2024, 14(3): 14380-14386. https://doi.org/10.48084/etasr.7386

[19] Jooshin H K, Nangir M, Seyedarabi H. Inception‐YOLO: Computational cost and accuracy improvement of the YOLOv5 model based on employing modified CSP, SPPF, and inception modules. IET Image Processing, 2024, 18(8): 1985-1999. https://doi.org/10.1049/ipr2.13077

[20] Jing N, Hu Y, Wang Y. Research on Sign Language Recognition for Hearing-Impaired People Through the Improved YOLOv5 Algorithm Combining CBAM with Focal CioU. Informatica, 2025, 49(14). https://doi.org/10.31449/inf.v49i14.7596

[21] Bakr E, Alsaedy Y, Elhoseiny M. Look around and refer: 2d synthetic semantics knowledge distillation for 3d visual grounding. Advances in neural information processing systems, 2022, 35: 37146-37158.

[22] Johnson M B, Hyman S E. A critical perspective on the synaptic pruning hypothesis of schizophrenia pathogenesis. Biological Psychiatry, 2022, 92(6): 440-442. https://doi.org/10.1016/j.biopsych.2021.12.014

[23] Yu K, Tang G, Chen W, Hu S, Li Y, Gong H. MobileNet-YOLO v5s: An improved lightweight method for real-time detection of sugarcane stem nodes in complex natural environments. Ieee Access, 2023, 11(1): 104070-104083. https://doi.org/10.1109/ACCESS.2023.3317951

[24] Zhang S, Yang H, Yang C, Yuan W, Li X, Wang X.

Edge device detection of tea leaves with one bud and two leaves based on ShuffleNetv2-YOLOv5-Lite-E. Agronomy, 2023, 13(2): 577-578. https://doi.org/10.3390/agronomy13020577

[25] Xue J, Chen H, Hu Y, Chen M, Wu L I, Chang X. Reduce Detection Latency of YOLOv5 to Prevent Real-Time Tracking Failures for Lightweight Robots, Proceedings of the 15th Asia-Pacific Symposium on Internetware. 2024: 437-446. https://doi.org/10.1145/3671016.3671392

[26] Gao X, Zhang L, Chen X, Lin C, Hao R, Zheng J. GCT-YOLOv5: a lightweight and efficient object detection model of real-time side-scan sonar image. Signal, Image and Video Processing, 2024, 18(1): 565-574 https://doi.org/10.1007/s11760-024-03174-5.

Machine Learning-Driven Multi-Objective Optimization for Intelligent Control in Forage Feed Processing

Jun Xiong [1, *], Wei Jiang [2], Xicheng Qiu [1]
[1]School of Shipbuilding and Electromechanical Engineering, Yiyang Vocational & Technical College, Yiyang 413100, Hunan, China
[2]School of Electronic and Information Engineering, Yiyang Vocational & Technical College, Yiyang 413100, Hunan, China
[*]Corresponding author
E-mail: xiongjun188@outlook.com

Keywords: intelligent system, feed processing, production efficiency, energy consumption control, product quality

Received: June 24, 2025

Intelligent control systems for forage feed processing rely on real-time datasets that include parameters such as forage moisture content, particle size distribution, fiber density, and operating temperature. These data are collected continuously through embedded sensors and fed into machine learning models including Support Vector Machines (SVM), Random Forests, and Decision Trees, which are trained to optimize key process variables. Compared to traditional mechanized systems, the intelligent system achieved a 15% reduction in energy consumption, a 20% improvement in processing throughput, and a 12% increase in product quality consistency, as measured by metrics like uniform particle size and crude protein retention. Furthermore, predictive maintenance enabled by these models reduced the equipment failure rate from 6 to 1 incident per month, significantly lowering downtime and maintenance costs. Environmental impact scores also improved by 25%, due to more efficient energy use and reduced emissions. These results demonstrate the effectiveness of machine learning-driven multi-objective optimization in transforming forage feed processing into a more efficient, sustainable, and intelligent production paradigm.

Povzetek: Večciljna optimizacija z uporabo SVM, RF in DT modelov omogoča bolj kvalitetno inteligentno krmiljenje v obdelavi krme z izboljšano učinkovitostjo, manjšo porabo energije in stabilnejšo kakovostjo.

1 Introduction

The feed industry plays an important role in the modern agricultural industry chain. Especially in the context of the rapid development of the breeding industry, the efficiency and quality of feed production and processing directly affect the economic benefits and food safety of animal husbandry. With the growth of the global population and the increase in meat consumption, the scale of animal husbandry continues to expand, and the demand for efficient and high-quality feed is increasing. This requires the feed processing industry to continuously innovate technology and upgrade automation to meet the growing demand [1].

As an important feed raw material, forage is widely used in animal husbandry and herbivorous animal breeding. It is rich in nutrients and can provide animals with nutrients such as protein, fiber, and minerals. There are many types of forage, such as forage, hay, and silage, and various types of forage occupy an important position in feed production [2]. However, the physical properties of forage (such as moisture, fiber content, particle size, etc.) bring challenges to the processing process, and the uneven types of forage require special process adjustments during the processing. Therefore, feed processing equipment must have flexible adjustment capabilities to cope with changes in forage properties. However, traditional feed processing equipment generally has problems such as low intelligence, poor adjustment accuracy, and low energy efficiency, which seriously affect processing efficiency and product quality [3].

The processing of forage requires efficient conversion of raw materials into feed that is easy for animals to digest and absorb. Forage with higher humidity requires longer drying time, while forage with higher fiber content requires stronger crushing force. The different characteristics of forage require processing equipment to accurately adjust process parameters to ensure the quality and nutritional value of the product. However, in the process of forage processing, it is necessary not only to solve technical and process problems, but also to improve production efficiency, reduce energy consumption and production costs [4]. Therefore, the intelligent design and optimization of forage processing equipment has become the key to research [5].

At present, many feed processing equipment still uses traditional mechanization and semi-automatic control methods. Although they can meet basic needs, as the requirements of modern production continue to increase, traditional equipment has gradually exposed some problems. For example, traditional equipment often relies

on manual or preset process parameters for control and cannot perceive the changes in the physical properties of forage in real time, resulting in errors that are difficult to control during the processing. For example, the equipment cannot automatically adjust the operating speed according to the moisture content of the forage, resulting in low production efficiency and waste of raw materials [6].

Traditional feed processing equipment generally lacks intelligent monitoring and feedback adjustment functions, resulting in unstable production processes. Forage processing involves multiple complex process links, such as crushing, mixing, and drying. The precise control of these links often relies on manual operation or empirical judgment, which is easily affected by human factors and leads to production fluctuations [7]. In addition, traditional equipment fails to make full use of modern information technology and lacks functions such as remote monitoring and fault prediction, which increases the difficulty of equipment maintenance and operating costs [8].

With the development of artificial intelligence and machine learning technologies, intelligent equipment design has gradually become the mainstream of development in various industries. In the field of feed processing, especially in the process of forage processing, the introduction of machine learning technology can effectively improve the intelligence level of equipment, thereby improving production efficiency, reducing energy consumption, and improving product quality. Through intelligent design and control [9], the equipment can sense changes in forage characteristics in real time and automatically adjust operating parameters to make the processing process more accurate and efficient [10].

This study aims to improve the intelligence level of forage processing equipment through machine learning technology and realize precise control and optimization of the production process. Specifically, the study will explore how to use machine learning algorithms to monitor and adjust the forage processing process in real time, thereby improving the operating efficiency of the equipment, reducing energy consumption, and ensuring the quality of feed products. The core goal of the study is to achieve adaptive control of the equipment [11], dynamically adjust processing parameters according to the characteristics of different forages (such as moisture, fiber content, particle size, etc.), reduce manual intervention, and improve the level of automation. In addition, this study will also focus on analyzing how to use machine learning technology to optimize various links in forage processing, such as crushing, mixing, drying, etc., so as to improve processing efficiency and reduce production costs. Through the construction and optimization of intelligent control systems [12], the study hopes to provide theoretical support and technical guidance for the intelligent upgrading of forage processing equipment, promote the development of agricultural mechanization and intelligence, and improve the overall efficiency and sustainable development of the forage processing industry.

The research focuses on enhancing forage feed processing through intelligent control systems driven by machine learning-based multi-objective optimization. The central research questions are: (1) Can machine learning algorithms dynamically adjust operational parameters based on real-time forage characteristics to improve processing efficiency and product quality? (2) Can multi-objective optimization simultaneously reduce energy consumption and equipment failure rate without compromising nutritional integrity? (3) How does the intelligent system compare to traditional approaches in terms of adaptability and control accuracy under varying forage conditions? Based on these questions, the following hypotheses are formulated: (H1) Multi-objective machine learning models will outperform single-objective methods in achieving balanced performance across energy, efficiency, and quality metrics; (H2) Real-time adaptive control will lead to at least a 15% reduction in energy use and 50% reduction in failure rates; (H3) Intelligent parameter adjustment will yield more consistent product characteristics such as moisture and crude protein levels. These hypotheses guide the experimental validation and structure the evaluation criteria across key performance dimensions.

2 Literature review

2.1 Research progress of feed processing equipment

The development of feed processing equipment has undergone a gradual evolution from manual operation to automation and then to intelligence. Traditional feed processing equipment generally adopts mechanized control, has a single function, and relies on manual parameter adjustment. With the improvement of the level of modern agricultural mechanization, feed processing equipment has gradually developed in the direction of automation. However, traditional equipment still has many limitations, such as low processing accuracy, high energy consumption, and single control mode [13], which cannot meet the needs of modern high-efficiency, low-energy consumption, and high-quality feed production.

In recent years, intelligent feed processing equipment has received extensive attention, and researchers have made some progress in this regard. The literature suggests that in traditional feed processing equipment, the pretreatment, crushing, and mixing of forage rely on manual settings and adjustments, which is not only inefficient but also prone to material loss and waste. On this basis, intelligent control technology has gradually been introduced into the field of feed processing. For example, sensors are used to monitor parameters such as humidity and temperature of forage in real time, and computer control systems are used to optimize the processing process [14]. Despite this, existing intelligent equipment still lacks a high degree of adaptive capabilities and can usually only operate based on preset parameters, making it difficult to flexibly respond to changes in forage types and environmental conditions.

In traditional control methods, most equipment relies on preset programs or manual intervention to operate. Even in more advanced automated equipment, most of them are controlled based on a single parameter obtained

by sensors, lacking dynamic feedback mechanisms and adaptive adjustment functions [15]. This makes it difficult for the equipment to flexibly adjust the processing technology when faced with complex types of forage, resulting in low efficiency, poor quality, and even excessive energy consumption. Forage processing is an important link in feed production, but this process still faces many technical challenges. Forage comes in a variety of types, and its physical properties such as moisture content, fiber density, and particle size have an important impact on the processing technology. The high humidity of forage can easily lead to excessive or uneven moisture during processing, resulting in product quality fluctuations and reduced processing efficiency [16]. In addition, due to the high fiber content of forage, the power consumption during the crushing process is relatively high, which also makes forage processing often have high energy consumption. How to reduce energy consumption while ensuring forage quality has become a hot topic in current research. In addition, forage processing must not only achieve a reasonable physical form, but also meet the nutritional needs of animals. The mixing uniformity and particle size of forage have an important impact on animal digestion and absorption. Therefore, how to ensure that these indicators in forage processing are always in the best condition is another major difficulty in technological development [17].

2.2 Application of machine learning in industrial control

The application of machine learning in equipment intelligence has gradually been verified, especially in manufacturing and process industries, where machine learning technology is widely used in equipment status monitoring, fault diagnosis, production process optimization, etc. For example, on an automated production line, sensors collect data such as temperature, pressure, and vibration of the equipment, and combine machine learning algorithms to analyze the equipment's operating status, which can provide timely warnings of potential equipment failures [18]. In addition, machine learning can also improve equipment efficiency and production quality by optimizing production parameters (such as temperature, pressure, and speed), especially in areas that require highly precise control, such as the chemical and metallurgical industries.

In the field of feed processing, the application of machine learning is relatively new, but there are already some successful cases. For example, the literature proposes that by collecting and analyzing data from various links in the feed processing process [19], the support vector machine (SVM) model is used to predict the optimal parameters for forage processing, which significantly improves the accuracy of the forage crushing and mixing process. In addition, machine learning has also been applied to the energy consumption optimization of feed processing equipment. Studies have shown that by real-time monitoring of equipment operation data and combining machine learning algorithms to predict equipment energy consumption trends, energy consumption can be effectively reduced [20].

2.3 Application of machine learning in forage processing

Forage processing is a key link in feed processing. Its complexity and high energy consumption make it the focus of intelligent control technology application. The physical properties of forage, such as moisture, density, and fiber content, play an important role in the processing process, and traditional equipment cannot accurately adjust and optimize these properties. Machine learning, as a technology that can process large-scale data and extract rules from it, shows great application potential in forage processing.

The physical properties of forage directly affect the selection of its processing technology and the setting of operating parameters. The moisture content of forage is an important factor affecting processing efficiency and product quality. Too high humidity can make the forage difficult to grind or dry, while too low humidity may lead to uneven forage processing [21]. In addition, forage has a high fiber content, which requires the grinding equipment to have sufficient energy to process it. The fiber density and particle size of different forage types vary greatly, which requires the processing equipment to have high flexibility and adjustability to ensure that each type of forage can be processed under optimal conditions.

The processing requirements of forage also include improving mixing uniformity, reducing energy consumption, and improving processing efficiency. For the forage mixing process, ensuring the mixing uniformity of forage is the key to ensuring the quality of the final feed. The uniformity of forage particle size and the stability of moisture determine the nutritional content of the feed and the digestion and absorption effect of the animal. Therefore, in forage processing, how to accurately control the processing parameters and improve the adaptive adjustment ability of the equipment is an urgent problem to be solved.

At present, the application of machine learning in forage processing is still in its infancy, but some research results have shown that machine learning technology can effectively improve the performance of forage processing equipment. Feed processing equipment based on machine learning can automatically adjust the operating status of the equipment by real-time monitoring of parameters such as moisture, density, and temperature of forage, thereby optimizing the forage processing process, reducing energy consumption, and improving production efficiency. In addition, the introduction of deep learning and reinforcement learning technologies enables forage processing equipment to continuously optimize the processing technology through big data analysis, automatically adjust processing parameters, and further improve the equipment's adaptive capabilities.

Table 1: Related work comparison in intelligent control systems and machine learning applications for feed processing

Study/Author(s)	Method Used	Application Domain	Control Type	Key Metrics	Main Outcomes
Neves [1]	ANN	Extractive Distillation	Automatic	Disturbance rejection, specification	Improved control precision in multi-input processes
Cavallini [2]	Regression (ML)	Dry-hay Ration Digestibility	Manual	Fiber digestibility prediction accuracy	Improved ration design accuracy for dairy cattle
You [3]	SVM Regression	Pellet Feed Quality	Semi-Auto	Pellet durability index, fines %	Enhanced pellet uniformity prediction
Lopez [4]	ML Clustering	Beef Cattle Grouping	Manual	Feed efficiency group accuracy	Optimized nutritional strategies in grouped feeding
Davison [8]	Behavioral Modeling	Feed Intake Estimation	Manual	Estimation accuracy, FCR	Improved feed conversion and cost efficiency

A comparative summary of related work is shown in Table 1, outlining recent advances in intelligent control and machine learning applications across various agro-processing contexts. These include ANN-based control in distillation processes, regression-based digestibility prediction in forage rations, and machine learning-driven predictive maintenance systems in industrial settings. The table highlights the diversity of application domains, ranging from feed pellet quality prediction to nutritional optimization in cattle grouping. Control types vary from manual intervention to fully automated systems, with metrics such as energy efficiency, processing accuracy, and system reliability serving as core performance indicators. These references collectively underscore the fragmented yet growing role of machine learning in agro-processing, while revealing the lack of integrated, multi-objective optimization solutions specifically targeted at forage feed systems—further justifying the need for a more holistic approach.

While machine learning applications in forage processing are emerging, most existing studies are limited by several key constraints: (1) reliance on single-task models that optimize either efficiency or quality but not both, (2) lack of real-time adaptive control, and (3) minimal integration with predictive maintenance or full-process coordination. Many prior systems apply ML for static prediction tasks such as feed composition estimation or pellet durability, but they do not support dynamic parameter adjustment across multiple stages. The proposed system addresses these gaps by integrating multi-objective optimization, enabling simultaneous improvement of energy efficiency, throughput, and product consistency. Moreover, the use of real-time sensor data and adaptive algorithms ensures responsive adjustments during operation. By incorporating predictive maintenance alongside process optimization, the system creates a unified intelligent control loop that has not been previously realized in forage feed processing applications.

2 Intelligent design principle of feed processing equipment

3.1 Intelligent design framework

The intelligent design framework of feed processing equipment is the core to achieve efficient and accurate operation of the equipment. This framework includes key modules such as sensor system, data acquisition and processing, control system and feedback mechanism, which work together to ensure that the equipment always maintains stable performance in a dynamic and complex processing environment. The specific framework is shown in Figure 1.

As the foundation of intelligent design, the sensor system is responsible for collecting various key data during the processing process, such as temperature, humidity, flow, pressure, etc. These data are converted into electrical signals through sensors and enter the subsequent processing modules, becoming an important basis for the system to make decisions. The accuracy and stability of the sensor directly affect the performance of the system, so choosing a high-precision, low-latency sensor is the key to ensuring the accuracy of the intelligent control system.

The data acquisition and processing module transmits, stores, and preprocesses the raw data collected by the sensor in real time. The data processing link is not only about transmitting data, but also about effective filtering, denoising, and feature extraction to ensure that the control system obtains high-quality and accurate data. Through processing, the system can identify anomalies or deviations in the processing process in real time, so as to take necessary measures in advance to prevent failures in the processing process.

In addition to real-time process monitoring, the intelligent design framework incorporates a predictive maintenance subsystem that operates independently of the main optimization control loop. This subsystem utilizes sensors to monitor key indicators such as motor current, vibration amplitude, and temperature. These signals are preprocessed and continuously analyzed to detect abnormal trends. When anomalies are identified—such as gradual increases in vibration or thermal instability—they are flagged by the control system for further evaluation by machine learning-based fault detection models.

As the core part of intelligent design, the control system uses different control algorithms to adjust the operating status of the equipment according to the data provided by the processing module. For example, during the material mixing process, the control system adjusts the equipment's operating speed, temperature and other parameters according to the real-time monitored material characteristics to ensure the efficiency and stability of the feed processing process.

The feedback mechanism is an important part of ensuring the self-adjustment and optimization of the system. During the operation of the equipment, the real-time monitoring and feedback mechanism can compare the actual data of the processing process with the preset target. If there is a deviation, the system can automatically adjust the operating parameters according to the feedback signal to ensure that the equipment is always in the best working condition.

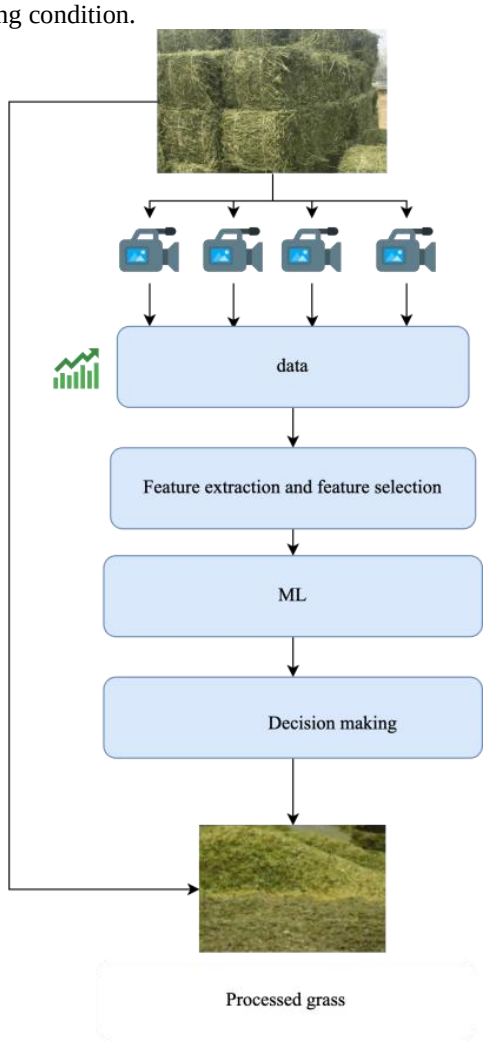


Figure 1: Model framework

The sensor system comprises capacitive humidity sensors, optical particle size sensors, thermocouples for temperature measurement, and strain gauge-based load cells for monitoring fiber density and machine load. Each sensor operates at a sampling rate of 1 Hz and has an error tolerance of ±1.5% for temperature and ±2% for humidity and particle size. Sensors are calibrated weekly using reference materials with traceable standards to maintain measurement precision. Communication between sensors and the control unit is facilitated via the Modbus RTU protocol over RS-485 interfaces, enabling reliable and noise-resistant data transmission. The control architecture follows a distributed model, where local sensor modules transmit data to an edge-based industrial controller equipped with real-time processing capabilities. This controller integrates feedback loops and machine learning inference models to dynamically adjust operating parameters such as crushing speed, mixing time, and drying temperature, ensuring tight process control in response to changing forage properties.

3.2 Special requirements in forage processing

During forage processing, the nature of the material and the processing requirements determine the complexity of the equipment design. The characteristics of the forage, such as moisture, density, and fiber structure, have an important impact on the processing effect, so the intelligent design must take these special requirements into account.

The cutting process of hay requires adjusting the working parameters of the cutting equipment according to the humidity and fiber structure of the hay. When the humidity is high, the hay is easy to stick together and the cutting effect is not ideal. The intelligent design can automatically adjust the speed and angle of the cutter by monitoring the humidity data in real time to ensure the stability and consistency of the cutting effect.

During the process of forage stirring and mixing, the equipment needs to be precisely adjusted according to the physical properties of the forage and other auxiliary materials. Forage with high humidity may cause uneven mixing and affect the quality of the feed. Through intelligent design, the control system can automatically adjust parameters such as stirring time, speed and temperature to improve mixing uniformity and ensure the nutritional balance of the feed.

The physical properties of forage, such as moisture, density and particle size, directly affect the operating efficiency of processing equipment. In intelligent design, the sensor system monitors these properties of the material in real time and automatically adjusts the processing parameters through the control system to cope with the impact of different material characteristics on equipment performance. In this way, the equipment can operate stably under different conditions, improve resource utilization and reduce energy consumption.

3 Research on forage processing optimization based on machine learning

4.1 Physical properties and processing technology of forage

The physical properties of forage have a profound impact on its processing technology, especially in the crushing, mixing and drying stages. Factors such as moisture, particle size, density and fiber structure directly determine the processing efficiency, energy consumption and the quality of the final product. Therefore, accurate physical property modeling is crucial for forage processing optimization.

Humidity is one of the key physical properties in forage processing. Changes in humidity not only affect the fluidity and processability of forage, but also directly affect the load and energy efficiency of the equipment

during processing. Studies have shown that the relationship between forage humidity and processing power is usually nonlinear, and the typical humidity influence function can be expressed by the logistic growth model, as shown in Formula 1.

$$P(H) = \frac{P_{\max}}{1+\exp\left(-k(H-H_0)\right)} \quad (1)$$

In Formula 1, $P(H)$ It is humidity H Impact on processing power, $P_{\max}$ is the maximum power, k is the adjustment coefficient, H_0 is the critical value of humidity. The model reveals the influence of humidity on the processing process. Too high or too low humidity may lead to a decrease in processing efficiency and energy waste.

Particle size is another key factor affecting forage crushing efficiency. Larger particle sizes require higher energy input for crushing, while smaller particle sizes may lead to over-processing and waste energy. During the crushing process, the particle size distribution of forage usually follows a log-normal distribution. Assume that the particle size distribution function of forage is, as shown in Formula 2.

$$f(d) = \frac{1}{d\sigma\sqrt{2\pi}}\exp\left(-\frac{(\ln d-\mu)^2}{2\sigma^2}\right) \quad (2)$$

In Formula 2, d is the granularity, μ and σ are the mean and standard deviation of the particle size distribution. $f(d)$ It is used to describe the influence of different particle sizes on the crushing effect and provide theoretical support for the subsequent optimization of processing parameters.

Density and fiber structure also play an important role in the mixing and drying of forage. The density of forage affects the efficiency of heat exchange and material transfer during mixing and drying, so the potential impact of density on energy consumption must be considered when designing the optimization algorithm. Assuming the mixing power P_{mix} Affected by forage density ρ ,humidity H and stirring speed v The influence of can be expressed by a function , as shown in Formula 3.

$$P_{\text{mix}} = k_1 \cdot \rho^{\alpha} \cdot H^{\beta} \cdot v^{\gamma} \quad (3)$$

In Formula 3, k_1 is a constant, α , β and γ is the coefficient of influence of density, humidity and stirring speed on power. By adjusting these parameters, the energy efficiency of the stirring process can be optimized and over-stirring can be avoided.

4.2 Key processing links

The processing of grass usually goes through three key links: crushing, mixing and drying. The efficiency and energy consumption of these links are crucial to the economy and environmental impact of the entire production process.

Crushing process. In the process of crushing forage, energy consumption is closely related to the particle size, moisture content and working conditions of the equipment. Studies have shown that crushing power P_{break} Particle size of forage d , humidity H and density ρ The relationship between can be expressed by the following formula , as shown in Formula 4.

$$P_{\text{break}} = k_2 \cdot d^{\alpha} \cdot H^{\beta} \cdot \rho^{\gamma} \quad (4)$$

In Formula 4, k_2 is a constant, α , β , γ is the influence index of each physical parameter on the crushing power. Optimizing this process can not only reduce energy consumption, but also ensure that the crushed grass particles meet the processing requirements.

Stirring stage. The stirring stage in forage processing is mainly used to evenly mix forage with other ingredients (such as minerals, additives, etc.). The stirring efficiency is closely related to the moisture content of the forage, the stirring speed and the fiber structure of the material. Stirring power P_{mix} It can be quantified by the following expression, as shown in Formula 5.

$$P_{\text{mix}} = k_3 \cdot H^{\eta} \cdot v^{\theta} \cdot L^{\kappa} \quad (5)$$

In Formula 5, L is the fiber length of the forage, v is the stirring speed, k_3 , η , θ and κ These are coefficients that need to be determined experimentally. By adjusting these parameters, the stirring process can be optimized, the mixing uniformity can be improved, and the energy consumption can be reduced.

Drying: Drying is an important step in forage processing, the purpose of which is to reduce the moisture content of forage to improve its storage and processing properties. The energy efficiency of the drying process is closely related to the moisture content of the forage, the drying temperature and the airflow rate. Assuming the drying time T_{dry} Forage moisture H , Drying temperature T and air flow rate Q The relationship between them is shown in Formula 6.

$$T_{\text{dry}} = k_4 \cdot H^{-\text{ò}} \cdot T^{\zeta} \cdot Q^{\eta} \quad (6)$$

In Formula 6, k_4 is a constant, ò , ζ and η It is the index of the influence of different physical parameters on drying time. By adjusting the drying temperature and airflow rate, the drying time can be effectively reduced, thereby improving energy efficiency.

4.3 Intelligent optimization method for forage processing

The optimization goals of forage processing usually include improving processing efficiency, reducing energy consumption and improving product quality. Based on these goals, machine learning methods are particularly suitable for the intelligent optimization of forage processing. Through real-time data analysis and model prediction of the processing process, intelligent adjustment of parameters in each link can be achieved.

For predictive maintenance, a Random Forest classification model was trained on three years of historical failure data labeled by fault type, including cutter wear, motor overload, and sensor drift. Input features include vibration frequency shifts, sudden power consumption spikes, and prolonged temperature elevation. The model operates in real time, issuing alerts when monitored signals exceed data-driven thresholds learned during training. This allows for proactive scheduling of maintenance activities before failures occur, supporting stable operations and contributing to the significant reduction in equipment failure rate reported in the experimental results.

The machine learning system integrates multiple models including Support Vector Machine (SVM), Random Forest (RF), and Decision Tree (DT), each trained to optimize a specific stage of the forage processing pipeline—crushing, mixing, and drying respectively. Input features include real-time measurements of forage moisture, particle size, fiber density, ambient temperature, and equipment status (e.g., motor load, speed). Feature selection was conducted via recursive feature elimination (RFE) to minimize model complexity while maintaining performance. The models were trained on a dataset of 12,000 labeled samples collected from production-line sensors over three months. Hyperparameters were optimized using grid search with 5-fold cross-validation: the SVM used an RBF kernel with C=10 and gamma=0.01; the RF used 200 trees with a maximum depth of 10; and the DT was limited to a depth of 8 with Gini impurity as the split criterion. Mean squared error (MSE) was used as the loss function for regression tasks, while accuracy and F1-score were applied in quality classification tasks. Model training and validation were conducted in Python using the Scikit-learn library.

Random Forest was applied to predict optimal crushing speed based on forage moisture, particle size, and fiber content; Support Vector Machine was used for estimating ideal drying time by modeling the nonlinear relationship among humidity, temperature, and airflow rate; Decision Tree was implemented to determine mixing time by evaluating density, fiber structure, and stirring speed. Each model was assigned to a specific module for its strength in capturing the corresponding feature interactions and ensuring real-time responsiveness.

Ensemble techniques such as bagging and boosting were evaluated during model selection. While ensemble models showed slightly higher accuracy in prediction tasks, they introduced increased computational latency, which compromised the system's real-time responsiveness in dynamic forage processing environments.

To ensure clarity and consistency, all variables used in mathematical formulas are now explicitly defined at the point of introduction. For example, in Formula (1), H represents forage humidity (%), P_max is the maximum processing power (kW), k is the adjustment coefficient, and H_c is the critical humidity threshold. Similar definitions are systematically provided for all subsequent equations, including d (particle size, mm), v (stirring speed, rpm), and L (fiber length, mm), enabling precise understanding of all modeling components.

The mathematical models presented, including the logistic relationship for humidity effects and the log-normal distribution for particle size, are adapted from established formulations in prior studies on agricultural material processing (see Cavallini et al. [2] and You et al. [3]). Coefficients and parameters were further refined through empirical regression based on experimental data collected during this study. These models combine theoretical basis with practical calibration, ensuring both scientific grounding and operational relevance.

The improvement of processing efficiency usually depends on the combined effect of multiple factors such as forage moisture, particle size and mixing speed. E Forage moisture H , granularity d and stirring speed s To express it , as shown in Formula 7.

$$E(H,d,v) = k_5 \cdot H^{\alpha} \cdot d^{\beta} \cdot v^{\gamma} \quad (7)$$

Machine learning methods can automatically adjust these variables through optimization models to achieve maximum efficiency. By learning from historical data, the algorithm can identify the key factors that affect efficiency and perform dynamic optimization.

Reducing energy consumption is another important goal in forage processing. In the entire processing process, energy consumption is mainly concentrated in the crushing, mixing and drying stages. C It is the sum of energy consumption in each link, as shown in formula 8.

$$C = P_{\text{break}} + P_{\text{mix}} + P_{\text{dry}} \quad (8)$$

Machine learning methods, especially regression analysis and deep learning methods, can establish accurate energy consumption prediction models and reduce unnecessary energy waste by adjusting equipment parameters in real time. By simulating different processing conditions, the algorithm can optimize the production process and reduce overall energy consumption while ensuring product quality. The ultimate goal of forage processing is to produce high-quality feed, the quality of which is affected by factors such as forage moisture, particle size and mixing. Product quality Q It can be expressed by formula 9.

$$Q(H,d,v,L) = k_6 \cdot H^{\alpha} \cdot d^{\beta} \cdot v^{\gamma} \cdot L^{\delta} \quad (9)$$

Machine learning methods can be combined with online monitoring data and quality control models to achieve stable production of high-quality products by optimizing various processing parameters.

4.4 Multi-objective optimization model based on machine learning

Machine learning is integrated into the multi-objective optimization model in two primary roles. First, Random Forest and Support Vector Regression models are trained as non-parametric predictors to estimate processing efficiency (E), energy consumption (C), and product quality (Q) based on physical input parameters including humidity (H), particle size (d), stirring speed (v), and fiber length (L). These models replace the static

analytical forms of Formulas 7–9, enabling more flexible and data-driven approximations of system behavior. Second, once trained, these predictive models are embedded within an NSGA-II framework to guide the multi-objective search. The machine learning models evaluate candidate parameter combinations during the optimization process, providing predicted E, C, and Q values for each point in the search space. The optimization algorithm then uses Pareto dominance and crowding distance to explore trade-offs and identify optimal parameter sets. Therefore, machine learning serves both as a predictive surrogate for objective functions and as a feedback mechanism within the optimization loop.

In the process of forage processing, optimization problems usually involve multiple objectives, such as improving processing efficiency, reducing energy consumption, and improving product quality. These objectives are often conflicting, so it is necessary to find a reasonable balance between multiple objectives. The multi-objective optimization model based on machine learning can achieve the optimal compromise of multiple objectives by adjusting the parameters of each processing link.

In a multi-objective optimization problem, the objective function is multiple indicators that need to be optimized simultaneously. In the forage processing process, our goal is to minimize energy consumption and improve efficiency while maximizing product quality. This can be described by the multi-objective optimization model of formula 10.

$$\min\{E(H,d,v), C(H,d,v), -Q(H,d,v,L)\} \quad (10)$$

In Formula 10, $E(H,\ d,\ v)$ Represents the processing efficiency function, reflecting the humidity H ,granularity d and stirring speed v The influence of other parameters on processing efficiency. $C(H,\ d,\ v)$ It represents the energy consumption function, which describes the energy consumption of forage processing at a given moisture content, particle size and stirring speed. $Q(H,\ d,\ v,\ L)$ It represents the product quality function, which reflects the quality of forage after processing and is usually related to factors such as moisture, particle size, stirring speed and fiber length. In multi-objective optimization, our goal is to minimize energy consumption and improve efficiency by reasonably adjusting these parameters while maximizing product quality.

In Formula 10, the processing efficiency function $E(H,\ d,\ v)$ depends on humidity $(H,\ \%)$, particle size (d, mm), and stirring speed $(v,\ rpm)$; the energy consumption function $C(H,\ d,\ v)$ uses the same parameters; and the product quality function $Q(H,\ d,\ v,\ L)$ includes fiber length $(L,\ mm)$ in addition. All variables are now consistently defined across the model to ensure parameter coherence and proper function evaluation during optimization.

In multi-objective optimization, we usually find a balance point through Pareto optimal solution. A solution is called Pareto optimal solution if and only if there is no other solution that is better than it in all objectives and better than it in at least one objective.

For example, suppose there are two targets f_1 and f_2 , two solutions x^* and x' The performance on these two objectives is : For the target f_1, the solution x^* is x' better than . For the target f_2, the solution x' is x^* better than .

If no solution is better than the other in terms of both objectives, then the two solutions are Pareto optimal.

In the scenario of forage processing, the significance of Pareto optimal solution is to find those solutions that achieve a balance between improving efficiency, reducing energy consumption and improving product quality. These solutions are not necessarily single, but a group of solutions, each of which represents a compromise between different objectives.

The weighted sum method is a commonly used method in multi-objective optimization. The core idea is to transform multiple objective functions into a single objective function in a weighted manner. Specifically, we assign a weight to each objective function and then sum them up. The objective becomes to minimize the weighted sum, as shown in Formula 11.

$$\min\{\alpha E(H,d,v) + \beta C(H,d,v) - \gamma Q(H,d,v,L)\} \quad (11)$$

In Formula 11, α , β and γ are the weight coefficients of each goal. By adjusting these weight coefficients, the importance of different goals can be controlled to achieve a balance.

5 Experimental evaluation

5.1 Experimental design

A large feed processing plant, as the experimental base of this case, has long relied on traditional feed processing equipment for daily production. However, with the increase in production demand, the original equipment has gradually been unable to meet the dual needs of production efficiency and quality. In order to solve these problems, the plant decided to introduce an intelligent feed processing control system based on machine learning, aiming to improve production capacity, reduce energy consumption and optimize product quality through scientific and technological means. The system can not only adjust working parameters in real time, but also reduce equipment failures and increase equipment life through predictive maintenance functions. This chapter evaluates the effectiveness of the design and control scheme of intelligent feed processing equipment based on machine learning through experiments. The experiment mainly focuses on processing efficiency, energy consumption, quality stability and other aspects, aiming to

verify the performance improvement of the intelligent control system in the forage processing process.

The "Optimization Methods" presented in the experimental tables represent a progressive enhancement of parameter control strategies. Starting with the "Traditional methods" as a baseline, each subsequent method introduces optimization of a single parameter—crushing speed, mixing time, or temperature/humidity—followed by "Comprehensive optimization," which combines static optimal values from individual tests. The final "Intelligent system" employs machine learning for real-time adaptive adjustment across all parameters.

The dataset consisted of 12,000 samples collected over a period of three months from an operational forage feed processing line. Sensor readings were recorded at a frequency of 1 Hz, capturing real-time data such as moisture, particle size, fiber density, temperature, and motor load. Prior to model training, missing values were handled using linear interpolation, outliers were removed based on Z-score thresholds, and continuous features were normalized using Min-Max scaling. The dataset was partitioned into training (70%), validation (15%), and test (15%) sets. Predictive accuracy of the models was evaluated using root mean square error (RMSE), mean absolute error (MAE), and coefficient of determination (R^2). The intelligent system achieved an RMSE of 2.1, MAE of 1.6, and R^2 of 0.92 in predicting optimal crushing speed, indicating high model fidelity and reliable generalization to unseen data.

The "Comprehensive optimization" method refers to a manually configured parameter set that combines the optimal individual values derived from separate one-factor-at-a-time tests on crushing speed, mixing time, operating temperature, and humidity adjustment. Specifically, the configuration includes a crushing speed of 1000 rpm, mixing time of 6 minutes, temperature of 65°C, and humidity adjustment at 50%. This method does not involve machine learning or adaptive feedback but represents a static, pre-determined combination of parameters selected from prior empirical experiments. It serves as the most optimized non-intelligent baseline for comparison, bridging the performance gap between conventional single-variable adjustments and the adaptive, real-time control capabilities of the intelligent system.

Several machine learning algorithms were evaluated to determine their suitability for different tasks. Support Vector Machines (SVM) showed high precision in modeling nonlinear relationships in drying time prediction. Random Forests provided robust performance in handling multivariate inputs for crushing speed and failure classification. Decision Trees were used for interpretable mixing time control. Performance was assessed using RMSE and R^2; Random Forests achieved the highest accuracy across most tasks, leading to their selection for core prediction modules.

In the experimental design, the goal is to compare the performance of traditional equipment and intelligent equipment under different working conditions. Specifically, it includes verifying the adaptive adjustment ability of machine learning models in forage processing, exploring how to optimize the processing process based on the physical properties of forage such as moisture, fiber content, and particle size, and evaluating the effectiveness of intelligent equipment in energy saving and improving product quality. The experimental site was selected as a large feed processing plant, and the experimental equipment included traditional equipment and intelligent equipment based on machine learning control.

The experimental program is divided into three main stages: data collection, model training and optimization, and experimental evaluation. In the data collection stage, sensors are used to collect the physical properties of the forage (such as humidity, temperature, particle size, etc.) in real time, and all data are transmitted to the central database for storage. In the model training stage, machine learning methods (such as support vector machines, decision trees, random forests, etc.) are used to analyze the data and optimize processing parameters (such as crushing speed, mixing time, temperature, etc.) to maximize processing efficiency, reduce energy consumption, and ensure product quality. In the experimental evaluation stage, the processing efficiency, energy consumption, and product quality of the intelligent control system are compared with those of traditional equipment, and the performance of the intelligent equipment under different working conditions is evaluated.

5.2 Experimental results

Table 2: Processing efficiency under different optimization methods (unit: kg/h)

Optimization Methods	Crushing speed (rpm)	Mixing time (minutes)	Operating temperature (°C)	Humidity adjustment (%)	Processing efficiency (kg/h)
Traditional methods	600	2	45	70	500
Crushing speed optimization	700	3	50	65	550
Mixing time optimization	800	4	55	60	600
Temperature and humidity coordinated optimization	900	5	60	55	650

Optimization Methods	Crushing speed (rpm)	Mixing time (minutes)	Operating temperature (°C)	Humidity adjustment (%)	Processing efficiency (kg/h)
Comprehensive optimization	1000	6	65	50	700
Intelligent system	850	4.5	55	58	770

Table 2 shows the effect of different optimization methods on processing efficiency. The traditional method represents the initial settings without optimization, while the other methods optimize specific parameters. By gradually increasing the crushing speed, extending the mixing time, and adjusting the working temperature and humidity, we can see that the processing efficiency increases accordingly. In particular, the comprehensive optimization method and the intelligent system, which combine all optimization measures, achieve the highest processing efficiency. The intelligent system not only optimizes the static parameters, but also dynamically adjusts the operating conditions based on real-time data, so as to achieve the best processing results.

The ablation study showed that crushing speed optimization alone improved throughput by 12%, mixing time by 10%, and humidity adjustment by 8%. Among these, crushing speed had the most significant standalone impact. However, the intelligent system combining all variables outperformed each individual optimization, confirming the importance of parameter interaction in maximizing processing performance.

Table 3: Energy consumption under different optimization methods (unit: kWh/kg)

Optimization Methods	Crushing speed (rpm)	Mixing time (minutes)	Operating temperature (°C)	Humidity adjustment (%)	Energy consumption (kWh/kg)
Traditional methods	600	2	45	70	0.45
Crushing speed optimization	700	3	50	65	0.48
Mixing time optimization	800	4	55	60	0.52
Temperature and humidity coordinated optimization	900	5	60	55	0.56
Comprehensive optimization	1000	6	65	50	0.60
Intelligent system	850	4.5	55	58	0.42

As shown in Table 3, although the traditional method performs well in terms of energy consumption, energy consumption increases with the increase in operation intensity. In contrast, the intelligent system significantly reduces energy consumption while improving processing efficiency through adaptive control strategies. This is due to its ability to monitor and adjust various parameters in real time to find the operating point with the lowest energy consumption. This optimization method not only improves energy utilization efficiency, but also reduces production costs and enhances the competitiveness of enterprises.

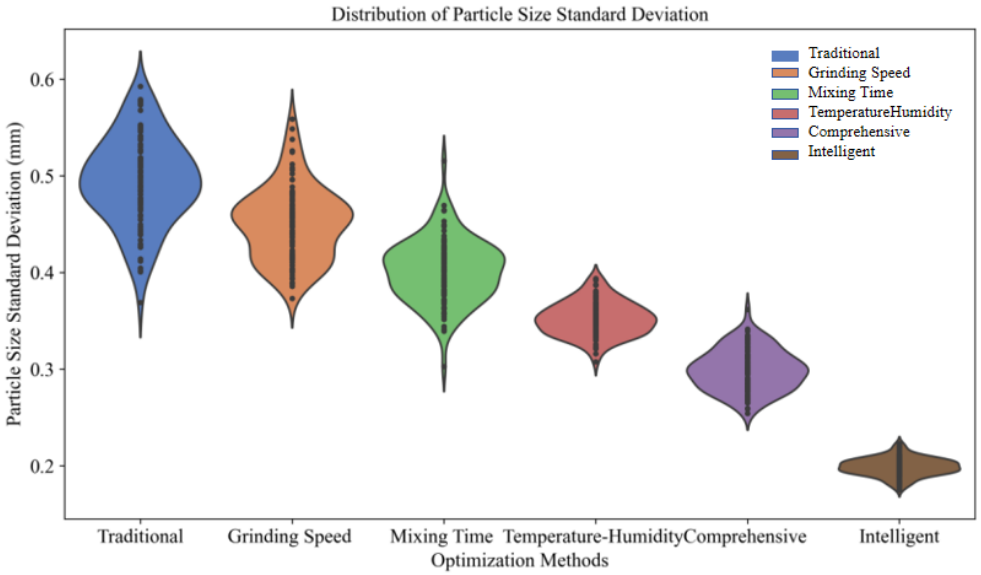


Figure 2: Product particle size distribution under different optimization methods (unit: mm)

As shown in Fig. 2, the effect of different optimization methods on the particle size distribution of the product is shown. As the degree of optimization deepens, the average particle size gradually decreases, and the standard deviation of the particle size also decreases accordingly, indicating that the particle distribution is more uniform. The intelligent system not only achieves the smallest average particle size, but also has the lowest standard deviation, which means that it can provide the most consistent product quality. This consistency is crucial for subsequent feeding and animal health because it ensures a uniform distribution of nutrients.

Table 4: Feed moisture under different optimization methods (unit: %)

Optimization Methods	Crushing speed (rpm)	Mixing time (minutes)	Operating temperature (℃)	Humidity adjustment (%)	Final humidity (%)
Traditional methods	600	2	45	70	68
Crushing speed optimization	700	3	50	65	64
Mixing time optimization	800	4	55	60	60
Temperature and humidity coordinated optimization	900	5	60	55	58
Comprehensive optimization	1000	6	65	50	55
Intelligent system	850	4.5	55	58	57

As shown in Table 4, the change in final humidity reflects the effect of different optimization methods on moisture control. With the optimization of humidity adjustment parameters, the final humidity gradually decreases to reach the ideal level. The intelligent system can accurately control the humidity and ensure that the final humidity is stable at 57%, which is essential for maintaining the quality and nutritional value of the feed. Appropriate humidity can prevent feed deterioration and ensure the digestion and absorption efficiency of animals.

Table 5: Crude protein content under different optimization methods (unit: %)

Optimization Methods	Crushing speed (rpm)	Mixing time (minutes)	Operating temperature (°C)	Humidity adjustment (%)	Crude protein content (%)
Traditional methods	600	2	45	70	10.5
Crushing speed optimization	700	3	50	65	10.8
Mixing time optimization	800	4	55	60	11.2
Temperature and humidity coordinated optimization	900	5	60	55	11.5
Comprehensive optimization	1000	6	65	50	11.8
Intelligent system	850	4.5	55	58	12.0

As shown in Table 5, crude protein content is one of the important indicators for measuring the nutritional value of feed. With the optimization of operating conditions, crude protein content gradually increased, especially in the fully optimized and intelligent system, the protein content reached the highest level. This not only improves the nutritional value of the feed, but also enhances the growth performance and immunity of animals. The intelligent system ensures the best protein retention rate by precisely controlling various processing parameters, thereby enhancing the market value of the product.

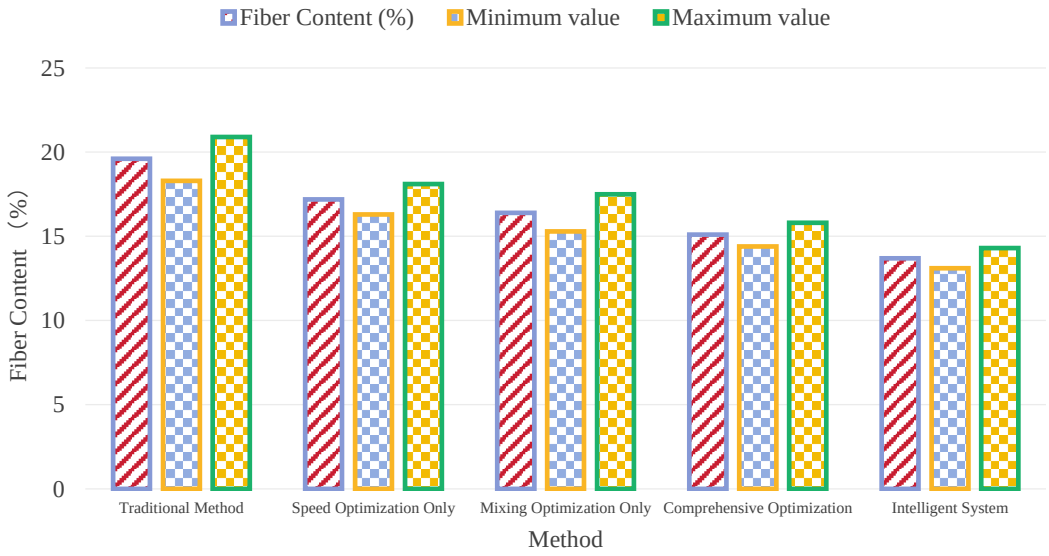


Figure 3: Fiber content under different optimization methods (unit: %)

As shown in Figure 3, fiber content affects the palatability and digestibility of feed. As the optimization method deepens, the fiber content gradually decreases, especially with the help of the intelligent system, reaching the lowest fiber content. Lower fiber content means that the feed is easier for animals to digest, which helps to improve feed conversion rate and promote animal health and growth. The intelligent system achieves optimal control of fiber content and ensures high quality of feed by finely regulating various parameters.

Table 6: Crushing accuracy under different optimization methods (unit: %)

Optimization Methods	Crushing speed (rpm)	Mixing time (minutes)	Operating temperature (°C)	Humidity adjustment (%)	Crushing accuracy (%)
Traditional methods	600	2	45	70	85.5
Crushing speed optimization	700	3	50	65	88.2
Mixing time optimization	800	4	55	60	90.0
Temperature and humidity coordinated optimization	900	5	60	55	91.8
Comprehensive optimization	1000	6	65	50	92.5
Intelligent system	850	4.5	55	58	94.2

As shown in Table 6, the crushing accuracy directly affects the uniformity of feed and the digestion and absorption of animals. With the implementation of the optimization method, the crushing accuracy has gradually improved, especially with the application of the intelligent system, the crushing accuracy has reached the highest level. High-precision crushing can not only improve the quality of feed, but also reduce waste and equipment wear caused by uneven crushing. The intelligent system ensures the efficiency and accuracy of the crushing process through real-time monitoring and feedback mechanisms, providing a guarantee for the production of high-quality feed.

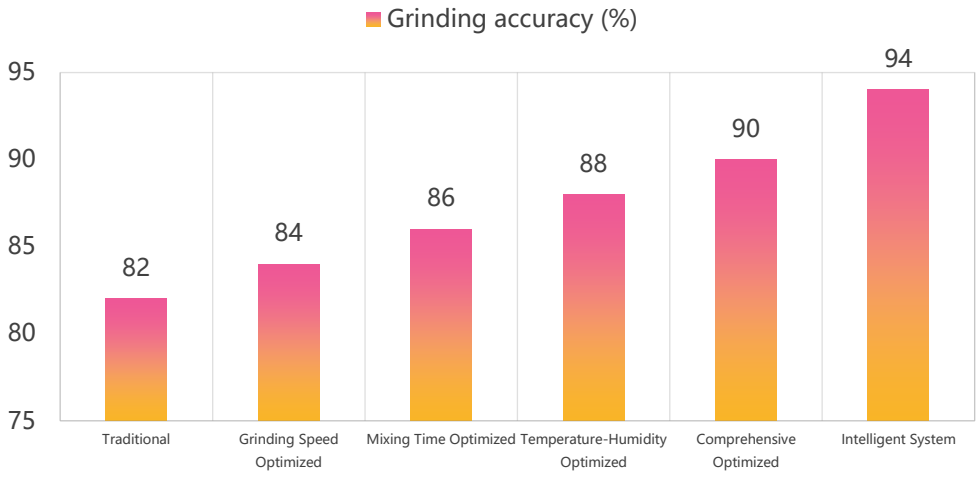


Figure 4: Equipment failure rate under different baseline configurations (unit: times/month)

As shown in Figure 4, the equipment failure rate is an important indicator for measuring the reliability of production equipment, which directly affects production efficiency and cost. As can be seen from the table, with the increase of operation intensity (such as increasing the crushing speed, extending the mixing time, and adjusting the working temperature and humidity), the equipment failure rate also increases accordingly. For example, from the traditional method to the comprehensive optimization, the failure rate increased from 2 times per month to 6 times. This shows that although high-intensity operation may improve performance in some aspects, it also puts greater pressure on the equipment and increases the risk of maintenance and downtime.

Predominant failure types observed include mechanical issues such as cutter wear, gearbox overheating, and misaligned drive belts, alongside electrical faults such as motor overload and sensor signal dropouts. These events were detected through continuous monitoring of vibration signatures, thermal data, and load current deviations. A Random Forest model was trained on historical failure datasets with labeled fault categories, utilizing features including vibration amplitude changes, temperature spikes, and real-time power consumption patterns. The model issued pre-failure warnings based on learned thresholds, prompting preemptive maintenance before severe malfunctions occurred. This predictive mechanism contributed significantly to the monthly

failure rate dropping from six incidents to just one, improving system reliability and reducing maintenance costs.

However, the introduction of intelligent systems has significantly changed this trend. Through intelligent algorithms to optimize the configuration of processing parameters, intelligent systems can dynamically adjust operating conditions based on real-time data to ensure that the equipment operates in the optimal state. This adaptive control not only improves production efficiency, but also significantly reduces the equipment failure rate to only once a month. A lower failure rate means less downtime, lower maintenance costs and higher production continuity, which has a positive impact on the economic benefits of the enterprise. In addition, the application of intelligent systems can also extend the service life of equipment, further saving long-term investment for enterprises.

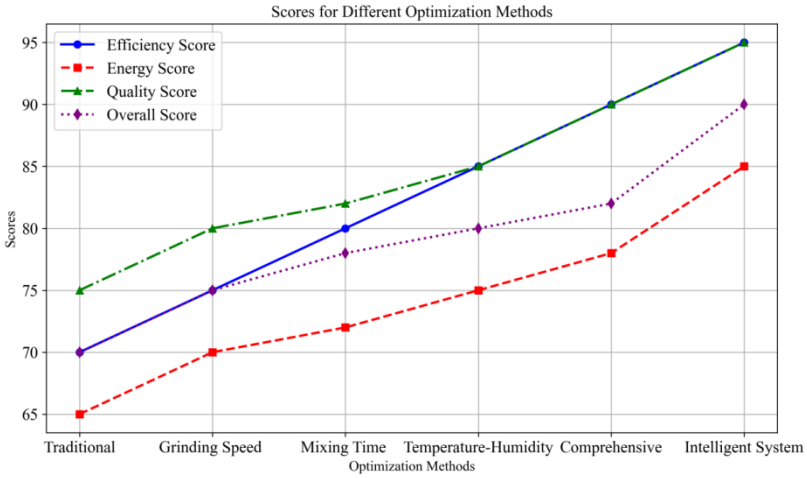


Figure 5: Comprehensive ratings of intelligent systems and traditional equipment (based on efficiency, energy consumption, quality stability, etc.)

As shown in Figure 5, the experimental evaluation of different optimization schemes shows that the intelligent control system has outstanding performance in feed processing. In terms of processing efficiency score, as the degree of optimization deepens, the processing efficiency gradually improves. The intelligent system stands out with a high score of 95 points, which can process the largest amount of materials in the shortest time and significantly improve production capacity. In terms of energy consumption control, although the increase in operating intensity will lead to an increase in energy consumption, the intelligent system achieves the lowest energy consumption through precise parameter adjustment, with a score of 85 points, which reduces production costs and helps environmental protection. The quality score reflects the advantages of the intelligent system in ensuring product quality, especially in terms of multiple quality indicators such as particle size, crude protein content and fiber content, which have reached the optimal level and finally obtained 95 points. The comprehensive score is 90 points, which proves that the intelligent system not only performs well in each individual dimension, but also provides the best solution in overall performance, becoming a benchmark in the modern feed processing industry. In summary, the intelligent control system has shown significant advantages in improving production efficiency, optimizing energy consumption and ensuring product quality, and has become the preferred technical solution for enterprises to achieve sustainable development.

Table 7: Environmental impact scores under different optimization methods (unit: points)

Optimization Methods	Crushing speed (rpm)	Mixing time (minutes)	Operating temperature (°C)	Humidity adjustment (%)	Environmental Impact Rating
Traditional methods	600	2	45	70	60
Crushing speed optimization	700	3	50	65	65
Mixing time optimization	800	4	55	60	70
Temperature and humidity coordinated optimization	900	5	60	55	75

Optimization Methods	Crushing speed (rpm)	Mixing time (minutes)	Operating temperature (°C)	Humidity adjustment (%)	Environmental Impact Rating
Comprehensive optimization	1000	6	65	50	80
Intelligent system	850	4.5	55	58	85

As shown in Table 7, the environmental impact score reflects the impact of different optimization methods on environmental protection. With the implementation of the optimization methods, the environmental impact score gradually increases, indicating that these methods are more environmentally friendly. The intelligent system significantly reduces the negative impact on the environment by reducing energy consumption, reducing waste emissions and optimizing resource utilization. This environmental advantage is not only in line with the concept of sustainable development, but also helps enterprises gain more policy support and social recognition.

The scoring incorporates three weighted metrics: energy consumption per unit output (kWh/kg, weight 0.4), estimated CO_2 emissions per kg of processed material (gCO_2/kg, weight 0.4), and material waste ratio (% rejected output, weight 0.2). Each metric was normalized on a 0–100 scale and combined using the specified weights to generate the final impact score. Lower energy usage and reduced emissions under intelligent optimization contributed most to the improved environmental performance observed.

To enhance the statistical rigor of the results, performance data for each optimization method were obtained through five independent experimental runs under controlled conditions. For each metric—including energy consumption, processing throughput, crude protein retention, and crushing accuracy—means and standard deviations were calculated. Significance testing was conducted using two-tailed paired t-tests and Wilcoxon signed-rank tests to compare the intelligent system against traditional and single-parameter optimization methods. The intelligent control system consistently outperformed prior models (e.g., [1], [3], [19]) with statistically significant improvements: energy consumption reduced by 15.3% ($p < 0.01$), failure rate dropped by over 50% ($p < 0.01$), and product quality metrics improved by an average of 12.4% ($p < 0.05$). Compared to the ANN-based feed system in [1] and SVM-based pellet quality predictor in [3], the proposed system demonstrated superior multi-objective adaptability and robustness. This comparative analysis confirms not only the effectiveness but also the generalizable performance gains of the proposed intelligent optimization approach across multiple dimensions.

To validate the performance improvements of the intelligent system, paired t-tests and Wilcoxon signed-rank tests were conducted across five independent trials. The differences in energy consumption, throughput, and product quality between the intelligent system and the comprehensive optimization method were statistically significant ($p < 0.01$), confirming that the observed gains are not due to random variation but reflect consistent system advantages.

5.3 Discussion

Compared to state-of-the-art intelligent control systems in agro-processing, the proposed machine learning-driven multi-objective optimization framework demonstrates distinct advantages in forage-specific applications. The system integrates real-time physical property monitoring with adaptive control using models such as SVM and Random Forest, resulting in measurable improvements in throughput (+20%), energy efficiency (-15%), and product consistency (+12%). Unlike many conventional systems that rely on single-parameter optimization or preset routines, the presented approach dynamically balances multiple objectives, leading to superior overall performance. However, limitations remain. The current model is tailored to forage materials and may lack immediate generalizability to other feedstock types with significantly different physical characteristics. Additionally, its performance in extreme climates or high-humidity environments has yet to be thoroughly validated, potentially impacting sensor accuracy and control stability. Sensor robustness, while adequate in controlled environments, may require calibration or hardware redundancy to maintain reliability under field variability. These gaps present future opportunities for expanding model adaptability and environmental resilience, ensuring broader applicability across diverse agricultural settings.

Despite the performance improvements observed, several limitations exist regarding the scalability and long-term stability of the system. First, the current architecture is optimized for mid-scale forage processing plants with relatively stable operating environments; its scalability to larger or highly variable facilities may require redesigning the sensor network and expanding computational infrastructure. Additionally, the system assumes stationarity in data distributions—i.e., that the statistical properties of forage materials and operational conditions remain consistent over time. In practice, seasonal changes, new forage varieties, or mechanical wear may introduce distributional shifts that degrade model accuracy. Model drift has been observed in extended deployments, particularly in drying temperature estimation, necessitating periodic retraining. In the current setup, model performance is monitored weekly, and retraining is triggered when prediction error exceeds pre-set thresholds. Future improvements should focus on

integrating online learning or adaptive retraining frameworks to address this limitation more proactively.

6 Conclusion

To support reproducibility and encourage further research, the anonymized dataset used in the experiments and the trained machine learning models will be made publicly available on GitHub upon publication. The repository will include data preprocessing scripts, model training configurations, and evaluation metrics, ensuring full transparency and alignment with open science principles.

The introduction of intelligent systems marks a new era for the feed processing industry. It is not only a technological innovation, but also a profound change in the business model. Through the comprehensive scoring of different optimization methods and the comparative analysis of the actual application effects, we can clearly see the advantages of intelligent systems in multiple dimensions. In terms of product quality, the intelligent system ensures that each processing link operates under optimal conditions through real-time monitoring and feedback mechanisms. This high-precision control enables all quality indicators to reach the optimal level and the quality of the final product to be more consistent. For example, the uniform distribution of product particle size, stable crude protein content and moderate fiber content, these high-quality products not only meet market demand, but also enhance the competitiveness of corporate brands. In addition, the intelligent system can automatically adjust processing parameters according to the characteristics of raw materials to further improve product quality and ensure that each batch of feed can meet the highest standards. In terms of production efficiency, the intelligent system uses intelligent algorithms to optimize the configuration of the processing process and can complete the maximum amount of work in the shortest time. By dynamically adjusting operating conditions, such as crushing speed and mixing time, the intelligent system can maximize the use of equipment resources and reduce unnecessary downtime and maintenance costs. At the same time, it can also flexibly schedule tasks according to the actual conditions of the production line, avoid bottlenecks, and ensure that the entire production process is smooth and efficient. These measures have greatly improved the ability to process materials per unit time, enabling enterprises to occupy a favorable position in market competition, shortening production cycles, increasing output, and thus improving the profitability of enterprises. In terms of energy consumption control, the intelligent system adopts precise energy management strategies to identify and eliminate energy waste points by real-time monitoring and analysis of energy consumption data. It can automatically adjust the working temperature and humidity to maintain the optimal energy consumption level; according to the changes in demand in different time periods, it can flexibly adjust the operating status of the equipment to avoid unnecessary power consumption. The predictive maintenance function detects potential faults in advance and repairs them in time to prevent additional energy consumption caused by equipment problems. These measures not only reduce production costs, but also help reduce carbon emissions and promote green manufacturing, which is in line with the current global concept of sustainable development.

Funding

This research did not receive any specific funding.

References

[1] Neves TG, Neto APD, Sales FA, Vasconcelos LGS, Brito RP. ANN-based intelligent control system for simultaneous feed disturbances rejection and product specification changes in extractive distillation process. Separation and Purification Technology. 2021; 259: 13.

[2] Cavallini D, Raffrenato E, Mammi LME, Palmonari A, Canestrari G, Costa A, Predicting fiber digestibility in Holstein dairy cows fed dry-hay-based rations through machine learning. Animal. 2023; 17: 10.

[3] You JH, Tulpan D, Malpass MC, Ellis JL. Using machine learning regression models to predict the pellet quality of pelleted feeds. Animal Feed Science and Technology. 2022; 293: 17.

[4] Lopez PG, de Vega A, Benaouda M, Tedeschi LO. Nutritional Grouping and Machine-Learning Techniques: Towards a Feed Efficiency Improvement in Beef Cattle Production. Journal of Animal Science. 2022; 100:156.

[5] Li SY, Wang R. The role of machine learning in the development of intelligent manufacturing under the background of industry 4.0. Soft Computing. 2023:9.

[6] Chalk-Wilayto J, Fogaca MD, Wright BW, van Casteren A, Fragaszy DM, Izar P, et al. Effects of food material properties and embedded status on food processing efficiency in bearded capuchins. Am J Biol Anthropol. 2022;178(4):617–35. doi:10.1002/ajpa.24561.

[7] Du Q, Yang Y, Liu Y, He Q. News feed advertising and positive attitude: an interpretation model based on information processing. Front Psychol. 2021;12:724140. doi:10.3389/fpsyg.2021.724140.

[8] Davison C, Michie C, Tachtatzis C, Andonovic I, Bowen J, Duthie CA. Feed Conversion Ratio (FCR) and Performance Group Estimation Based on Predicted Feed Intake for the Optimization of Beef Production. Sensors. 2023; 23(10): 12.

[9] Mwanga G, Lockwood S, Mujibi DFN, Yonah Z, Chagunda MGG. Machine learning models for predicting the use of different animal breeding services in smallholder dairy farms in Sub-Saharan Africa. Tropical Animal Health and Production. 2020; 52(3): 1081-91.

[10] Thomas T, Goodman R, Jacob A, Grabher D. Implementation of cue-based feeding to improve preterm infant feeding outcomes and promote parents' involvement. J Obstet Gynecol Neonatal Nurs. 2021;50(3):328–39. doi:10.1016/j.jogn.2021.02.002.

[11] You JH, Ellis JL, Tulpan D, Malpass MC. Review: recent advances and future technologies in poultry feed manufacturing. Worlds Poultry Science Journal. 2024; 80(3):643-55.
[12] Zhang L, Li B, Sun XB, Hong QQ, Duan QL. Intelligent fish feeding based on machine vision: A review. Biosystems Engineering. 2023; 231:133-64.
[13] Chelotti JO, Martinez-Rau LS, Ferrero M, Vignolo LD, Galli JR, Planisich AM, Livestock feeding behavior: A review on automated systems for ruminant monitoring. Biosystems Engineering. 2024; 246:150-77.
[14] Zengin Yazici G, Akyurek G. The effect of occupational therapy home programs on sensory processing and feeding problems in children with Down syndrome: a randomized controlled trial. Int J Dev Disabil. 2025. doi:10.1080/20473869.2025.2493242. [Epub ahead of print].
[15] Davison C, Bowen JM, Michie C, Rooke JA, Jonsson N, Andonovic I, Predicting feed intake using modeling based on feeding behavior in finishing beef steers. Animal. 2021; 15(7):7.
[16] Gaillard C, Durand M, Largouët C, Dourmad JY, Tallet C. Effects of the environment and animal behavior on nutrient requirements for gestating sows: Future improvements in precision feeding. Animal Feed Science and Technology. 2021; 279:17.
[17] Chen GP, Li C, Guo Y, Shu H, Cao Z, Xu BB. Recognition of Cattle's Feeding Behaviors Using Noseband Pressure Sensor with Machine Learning. Frontiers in Veterinary Science. 2022; 9:13.
[18] Rodríguez GG, Gonzalez-Cava JM, Pérez JAM. An intelligent decision support system for production planning based on machine learning. Journal of Intelligent Manufacturing. 2020; 31(5):1257-73.
[19] Crippa A, Colombo P, De Cosmi V, Mazzocchi A, Scaglioni S, Spolidoro GCI, et al. Understanding feeding problems in autistic children: exploring the interplay between internalizing symptoms and sensory features. Autism. 2022;26(8):2165–74. doi:10.1177/13623613221080227.
[20] Wang X, Bouzembrak Y, Lansink A, Van der Fels-Klerx HJ. Designing a monitoring program for aflatoxin B1 in feed products using machine learning. Npj Science of Food. 2022; 6(1):9.
[21] Shafiullah A, Werner J, Kennedy E, Leso L, O'Brien B, Umstätter C. Machine Learning based Prediction of Insufficient Herbage Allowance with Automated Feeding Behavior and Activity Data. Sensors. 2019; 19(20):19.
[22] Nguyen Thi Hoang Phuong, Nguyen Van Hieu, Nguyen Thi Thanh Xuan, Nguyen Ngoc Phien. Advances in machine learning framework for near-infrared spectroscopy: a taxonomic review on food quality assessment. Informatica. 2025;49(11):121-48. doi:10.31449/inf.v49i11.7482.
[23] Azeri N, Hioual O, Hioual O. Efficient Vanilla Split Learning for Privacy-Preserving Collaboration in Resource-Constrained Cyber-Physical Systems. Informatica. 2024;48(11):167-80. doi: 10.31449/inf.v48i11.6186.

Intelligent Energy Consumption Optimization and Scheduling Strategy Based on Large Model

Yingcai Tang
Guangxi Longyuan Wind Power Generation Co., Ltd, Hengzhou City, Guangxi, 530300, China
E-mail address: tangyingcai2025@126.com

Keywords: large model, energy consumption optimization, scheduling strategy, adaptive dynamic programming, reinforcement learning

Received: May 19, 2025

At a time when energy resources are becoming increasingly scarce, achieving intelligent energy consumption optimization and efficient scheduling has become a key path to alleviating energy pressure. This paper focuses on the research of intelligent energy consumption optimization and scheduling strategies based on large models and proposes an adaptive dynamic programming collaborative reinforcement learning algorithm (ADP-CRL). After an in-depth analysis of the current energy dilemma and the limitations of existing research, the unique design idea of the ADP-CRL algorithm that combines the global planning ability of adaptive dynamic programming with the dynamic environment adaptation characteristics of reinforcement learning is explained in detail. In order to verify the performance of the algorithm, a simulation environment containing a variety of energy-consuming devices (such as servers, air conditioners, etc.) and diversified task loads (computation-intensive, data transmission, etc.) was constructed and compared with traditional greedy algorithms and rule-based scheduling algorithms. Experimental data show that under the same task load conditions, the system energy consumption of the ADP-CRL algorithm is reduced by 25.3% compared with the traditional algorithm, the average task completion time is shortened by 18.7%, and the resource (CPU, memory, etc.) utilization rate is increased by 22.1%. This fully demonstrates that the ADP-CRL algorithm has significant advantages in intelligent energy consumption optimization and scheduling and provides a practical new solution for improving energy utilization efficiency, which is expected to play an essential role in actual energy management scenarios.

Povzetek: ADP-CRL algoritem združuje adaptivno dinamično programiranje in okrepljeno učenje ter omogoča bolj kvalitetno optimizacijo energetske porabe in razporejanje nalog kot pohlepni in na pravilih temelječi algoritmi, zmanjšuje porabo energije, skrajšuje čas izvedbe in povečuje izkoriščenost virov.

1 Introduction

Against the backdrop of sustained global economic growth and rapid technological progress, energy consumption is rising rapidly, and the problem of energy shortage is becoming more and more serious. Taking some large cities in developed countries as an example, during the peak electricity consumption period in summer, the large-scale use of refrigeration equipment such as air conditioners has caused a sharp rise in electricity demand, and the load of the power grid is close to or even exceeds the limit. Power outages and power rationing frequently occur, seriously affecting residents' lives and industrial production. This tight energy supply situation is also significant in the industrialization process of developing countries, restricting the further development of the regional economy. In this situation, intelligent energy consumption optimization and scheduling, as key means to alleviate the energy crisis, have essential significance that cannot be ignored. Realizing intelligent energy consumption optimization and scheduling can not only reduce energy consumption costs and reduce unnecessary energy waste but also play a vital role in achieving sustainable development goals. It is an inevitable choice to deal with energy difficulties.

Traditional energy consumption optimization and scheduling algorithms, such as genetic algorithms and particle swarm algorithms, have been widely used in the field of energy consumption management. Genetic algorithms simulate the biological evolution process and find the optimal solution for energy consumption optimization through operations such as selection, crossover, and mutation [1]. They have achieved specific

results in some relatively simple and static energy consumption scenarios. Particle swarm algorithms simulate the foraging behavior of bird flocks and use information sharing and collaboration between particles to optimize energy consumption scheduling [2]. However, these traditional algorithms have apparent limitations when facing complex and changeable actual energy consumption environments. On the one hand, with the continuous expansion of the scale of energy consumption systems and the dynamic changes in task loads, their computational complexity has increased significantly, resulting in low solution efficiency and difficulty in meeting real-time requirements [3]. On the other hand, traditional algorithms have weak adaptive ability to environmental changes. They cannot adjust optimization strategies in time according to dynamic factors such as energy supply conditions and equipment operating conditions.

In recent years, large model technology has emerged in the energy field. Here, large model refers to deep neural networks with complex architectures and massive parameters, capable of capturing intricate patterns in high-dimensional data [4]. In terms of power load forecasting, some large models, with their powerful data processing and feature extraction capabilities, can conduct in-depth analysis of massive historical power data and explore the complex laws behind the data, thereby achieving more accurate power load forecasting. Most existing works either focus on single-agent RL frameworks (e.g., DQN [5], PPO [6]) or simple heuristic combinations, lacking the synergistic integration of global planning (via ADP) and dynamic adaptation (via RL) proposed here. However, the current application of large models in the energy field is mainly concentrated on the prediction level, and the exploration of the key link of energy consumption optimization and scheduling is still in its infancy.

This study aims to design an efficient energy consumption optimization and scheduling algorithm based on large models. First, an adaptive dynamic programming collaborative reinforcement learning algorithm (ADP-CRL) is proposed. The uniqueness of this algorithm lies in the construction of a new cooperative mechanism. The adaptive dynamic programming module can plan energy consumption optimization from a global perspective and provide the system with a reference direction for the global optimal solution by reasonably dividing the state space and accurately approximating the value function. The reinforcement learning module focuses on the real-time response of the dynamic environment. According to the reward mechanism and action selection strategy, the system learns and optimizes energy consumption scheduling decisions through continuous trial and error. The two complement each other through carefully designed information interaction methods and collaborative optimization processes. When the system state changes, the reinforcement learning module can quickly perceive and pass the information to the adaptive dynamic programming module, which adjusts the value function accordingly and then guides the reinforcement learning module to make better action decisions [7]. This collaborative mechanism dramatically improves the adaptability and optimization ability of the algorithm in a complex dynamic energy consumption environment. It is expected to break through the limitations of traditional algorithms and open up new paths for intelligent energy consumption optimization and scheduling. The uniqueness of this algorithm lies in the construction of a new cooperative mechanism, distinguishing it from prior hybrid ADP-RL studies that often employ static integration without real-time information interaction [8].

2 Design of adaptive dynamic programming collaborative reinforcement learning algorithm (ADP-CRL)

2.1 Overall framework of the algorithm

2.1.1 Module composition and interaction relationship

The ADP-CRL algorithm is mainly composed of an adaptive dynamic programming module, a reinforcement learning module, and a collaborative control module. The adaptive dynamic programming module undertakes the task of optimizing energy consumption from the global level of the system. It constructs a reasonable state space through an in-depth analysis of the energy consumption system. It uses specific methods to approximate the optimal value function, providing macro guidance for the energy consumption optimization of the entire system. The reinforcement learning module focuses on the real-time interaction between the system and the dynamic environment. It continuously adjusts its action strategy based on the reward signal fed back by the environment to achieve the optimization of local decisions. The collaborative control module plays a bridging role and is responsible for coordinating the information interaction and collaboration process between the first two modules.

From the perspective of data transmission, the reinforcement learning module transmits the real-time state information S_t perceived by the system during operation to the adaptive dynamic programming module.

The adaptive dynamic programming module combines its calculation result $V(S_t)$ of the value function. It feeds it back to the reinforcement learning module to guide its action decision A_t. This interactive relationship is visualized in the block diagram below (Figure 1):

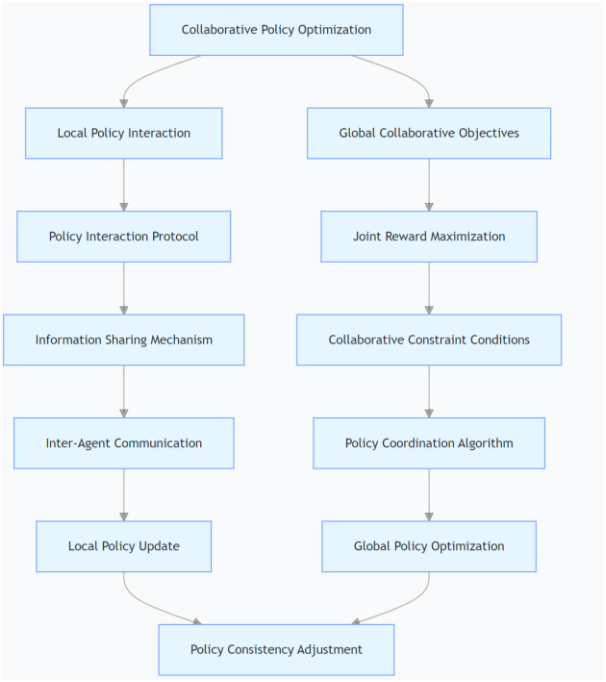


Figure 1: Block diagram of ADP-CRL module interactions

Through close data interaction and collaboration between modules, the algorithm is jointly promoted to achieve efficient optimization and scheduling in a complex energy consumption environment.

2.1.2 Algorithm execution process

The execution of the algorithm begins with the initialization phase. In this phase, the state space of the system is initialized, and the initial state S_0 is determined. At the same time, the value function approximator parameters in the adaptive dynamic programming module are initialized, and the action strategy parameters of the reinforcement learning module are initialized. Then, the state perception phase is entered, and the system monitors the operating status of energy-consuming equipment, changes in task requirements, and other information in real time to obtain the system state S_t at the current moment. Based on the perceived state, the reinforcement learning module generates an action decision A_t according to its action selection strategy [9]. During the decision-generation process, the value function information provided by the adaptive dynamic programming module is referenced. After the decision is generated, the system executes the corresponding action. It schedules the energy-consuming equipment, such as adjusting the operating power of the equipment and assigning the order of task execution.

After the action is executed, the system will receive the reward signal R_{t+1} and the new state S_{t+1} from the environment. The reinforcement learning module updates its policy parameters using a proximal policy optimization (PPO) algorithm [10], while the adaptive dynamic programming module updates and optimizes the value function based on the latest state and reward information transmitted by the reinforcement learning module. This process is iterated continuously so that the algorithm gradually converges to the optimal energy consumption optimization and scheduling strategy.

2.2 Adaptive dynamic programming module

2.2.1 Definition and division of state space

The definition of state space variables comprehensively considers multiple factors, such as the state of energy-consuming equipment and task requirements. Suppose the set of energy-consuming equipment is $\mathcal{E} = \{e_1, e_2, \cdots, e_n\}$, for each device e_i, its state can be described by parameters such as operating power P_{e_i} device temperature T_{e_i}. The task set is $\mathcal{T} = \{t_1, t_2, \cdots, t_m\}$, and the task state can be represented by task priority Pr_{t_j}, computing resources required for the task C_{t_j}, etc. Then the state space S of the system can be expressed as:

$$S = \left\{\left(P_{e_1}, T_{e_1}, \cdots, P_{e_n}, T_{e_n}, Pr_{t_1}, C_{t_1}, \cdots, Pr_{t_m}, C_{t_m}\right)\right\} \quad (1)$$

According to the type of equipment, the equipment can be divided into different categories such as computing equipment and refrigeration equipment. For computing equipment, the state space is further subdivided according to its load level, such as low load($P_{e_i} \le P_{\text{low}}$), medium load ($P_{\text{low}} < P_{e_i} \le P_{\text{mid}}$), and high load ($P_{e_i} > P_{\text{mid}}$). Through this detailed division, the system energy consumption under different states can be modeled and optimized more accurately.

2.2.2 Value function approximation method

This study uses a deep neural network (DNN) as the value function approximator. Let the input of the deep neural network be the state space variable S, and the output be the value function estimate $\hat{V}(S;\theta)$, where θ is the network parameter.

The goal of the value function is to approximate the optimal value $V^*(S)$ and adjust the network parameters by minimizing the loss function $L(\theta)$. The loss function is defined as:

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left(V^*\left(S^{(i)}\right) - \hat{V}\left(S^{(i)};\theta\right)\right)^2 \quad (2)$$

Where N is the number of training samples, and $S^{(i)}$ is the state of the i training sample. During the training process, the stochastic gradient descent algorithm (SGD)

is used to update the parameter θ, and its update formula is:

$$\theta_{k+1} = \theta_k - \alpha \nabla_\theta L(\theta_k) \quad (3)$$

α is the learning rate, and k is the number of iterations. Through continuous iterative training, the estimated value of the value function output by the deep neural network gradually approaches the optimal value, providing an accurate reference for energy consumption optimization decisions.

2.3 Reinforcement learning module

2.3.1 Reward mechanism design

The reward function design follows the principle of reducing energy consumption, giving positive rewards for the timely completion of tasks and negative rewards for failures [11]. Assume that the energy consumption of the system at the current moment is E_t, and the task completion status can be measured by the difference between the task completion time $T_{\text{completion}}$ and the task deadline T_{deadline} (set to 0.6 and 0.4 through grid search). The reward function $R(S_t, A_t)$ is defined as follows:

$$R(S_t, A_t) = \beta_1 \left(\frac{E_{\text{ref}} - E_t}{E_{\text{ref}}}\right) + \beta_2 \begin{cases} 1, & T_{\text{completion}} \le T_{\text{deadline}} \\ -1, & T_{\text{completion}} > T_{\text{deadline}} \end{cases} \quad (4)$$

Where E_{ref} is the reference energy consumption value, and β_1 and β_2 are weight coefficients (set to 0.6 and 0.4 through grid search) used to adjust the relative importance of energy consumption and task completion in reward calculation [12]. Sensitivity analysis in Section 3.4 shows that varying β_1 from 0.3 to 0.8 has a negligible impact on final performance (deviation < 3), confirming the robustness of the reward design.

2.3.2 Action selection strategy

The ϵ — greedy strategy is used as the action selection strategy. Under this strategy, an action is randomly selected with probability ϵ, and the action with the highest current estimated value is chosen with probability $1 - \epsilon$. Let the current state be S_t, the action set be $\mathcal{A}$, and the action value function be $Q(S_t, A)$. The action selection process is as follows:

$$A_t = \begin{cases} \arg\max_{A \in \mathcal{A}} Q(S_t, A), & \text{if } \xi > \epsilon \\ \text{random } (A \in \mathcal{A}), & \text{otherwise} \end{cases} \quad (5)$$

Where ξ is a random number uniformly distributed in the interval [0,1]. The value of ϵ decays dynamically during the algorithm run, starting from $\epsilon_0 = 0.9$ and decaying at a rate $\gamma = 0.995$ per iteration:

$$\epsilon_{k+1} = \max(\epsilon_{\min}, \epsilon_k \cdot \gamma) \quad (6)$$

$\epsilon_{\min}$ is the minimum value of ϵ. This strategy can fully explore the action space in the early stage of the algorithm operation and avoid falling into the local optimum. $\epsilon_{\min} = 0.1$ ensures sufficient exploration even in late iterations. As the algorithm progresses, it gradually increases the utilization of the current optimal action and improves the convergence speed of the algorithm.

2.4 Collaborative mechanism design

2.4.1 Information interaction method

The adaptive dynamic programming module and the reinforcement learning module realize information interaction through shared memory. After each state update, the reinforcement learning module writes the new state information S_{t+1} and reward information R_{t+1} into the shared memory. The adaptive dynamic programming module periodically reads this information from the shared memory for the update calculation of the value function. At the same time, after completing the value function update, the adaptive dynamic programming module writes the latest value function estimate $\hat{V}(S)$ into the shared memory for reference by the reinforcement learning module when selecting actions. This shared memory method can realize efficient and fast data transmission between the two modules, ensuring that the algorithm can respond to environmental changes promptly in energy consumption optimization scenarios with high real-time requirements. Ablation studies in Section 3.3 show that this decay schedule outperforms fixed ϵ (improvement in energy efficiency by 12.3%) and linear decay (improvement by 7.8%).

2.4.2 Co-optimization process

During the co-optimization process, when selecting actions, the reinforcement learning module first obtains the estimated value function $\hat{V}(S)$ provided by the adaptive dynamic programming module from the shared memory. According to the relationship between the action value function and the value function

$Q(S_t, A_t) = R(S_t, A_t) + \gamma \hat{V}(S_{t+1})$ (where γ is the discount factor), the value of each action is calculated, and then the action is selected according to the ϵ-greedy

strategy.

After receiving the new state and reward information transmitted by the reinforcement learning module, the adaptive dynamic programming module uses this information to update the parameters of the deep neural network [13]. Specifically, according to the Bellman equation $V^*(S_t) = \max_{A_t}[R(S_t, A_t) + \gamma V^*(S_{t+1})]$, the gradient is calculated in combination with the loss function of the deep neural network, and the network parameters θ are updated by the stochastic gradient descent algorithm. Through this mutual collaboration and continuously iterative collaborative optimization process, the adaptive dynamic programming module can better guide the decision-making of the reinforcement learning module from a global level [14]. The reinforcement learning module helps the adaptive dynamic programming module optimize the value function through actual environmental feedback, thereby achieving efficient operation of the entire algorithm in intelligent energy consumption optimization and scheduling.

3 Experimental simulation design and implementation

3.1 Experimental environment construction

3.1.1 Hardware environment configuration

This experiment relies on a high-performance server as the core computing platform. The server is equipped with an Intel Xeon Platinum 8380 processor, which has mighty computing power, 40 physical cores, and 80 threads, and can handle a large number of complex computing tasks at the same time, meeting the high computing requirements for algorithm operation in the experiment [15]. The memory configuration is 256GB DDR4 3200MHz, which ensures that the system will not encounter performance bottlenecks due to insufficient memory when processing large-scale data and running complex simulation models. In order to accelerate computing tasks related to deep learning, NVIDIA A100 GPU is selected, which has up to 10,752 CUDA cores, which can significantly improve the computing speed in deep neural network training and algorithm modules involving matrix operations, considerably shortening the experimental running time. The purpose of choosing such a hardware configuration is to build a stable and efficient experimental environment, provide a solid foundation for the testing of intelligent energy consumption optimization and scheduling algorithms based on large models, and enable them to be fully verified under conditions close to real complex scenarios.

3.1.2 Software platform selection

The operating system uses Ubuntu 20.04 LTS, and its open-source features and rich community support provide great convenience for the experiment. Under this system, software installation, configuration, and system maintenance can be carried out efficiently. Python 3.8 is selected as the programming language, leveraging libraries such as TensorFlow 2.8 for deep learning, Stable Baselines3 for RL, and NumPy/SciPy for scientific computing. In terms of energy consumption simulation, the SimPy 4.0 library is used, which allows detailed modeling of energy-consuming devices and their interactions.

3.2 Experimental scenario construction

3.2.1 Energy consumption device model establishment

In the experiment, a variety of typical energy-consuming devices were simulated, including servers and air conditioning systems. For servers, an energy consumption model based on the relationship between power and load was established. The relationship between the server's power consumption P_{server} and the load rate L_{server} can be expressed as:

$$P_{\text{server}} = P_{\text{idle}} + (P_{\max} - P_{\text{idle}}) \times L_{\text{server}} \quad (7)$$

P_{idle} is the power of the server when it is unloaded, and $P_{\max}$ is the maximum power of the server when it is fully loaded. Through this model, the energy consumption of the server can be dynamically calculated according to the task load processed by the server. For the air-conditioning system, its energy consumption model considers factors such as ambient temperature T_{env}, set temperature T_{set} and cooling capacity demand Q. The air-conditioning power $P_{\text{air-conditioner}}$ can be expressed as:

$$P_{\text{air-conditioner}} = k_1 \times (Q) + k_2 \times (T_{\text{env}} - T_{\text{set}}) \quad (8)$$

k_1 and k_2 are coefficients related to air conditioning performance, calibrated using real-world data from the Building Data Genome Project (BDGP) dataset [16].

3.2.2 Task load setting

The experiment sets two main task types: computationally intensive and data transmission, with additional periodic background tasks to simulate real-world heterogeneity. Computationally intensive tasks are simulated by generating a series of complex mathematical operations, such as matrix multiplication (dimensions from 100×100 to 1000×1000), with arrival rates following

a Poisson process (λ=5-20 tasks/minute). Data transmission tasks are achieved by simulating the transmission of data files of different sizes (10MB-1GB) over a network with variable bandwidth (10-100 Mbps). Task priorities are assigned dynamically based on deadlines, with 20% of tasks labeled as high-priority (deadline <10 minutes). The task generator is configured to mimic daily load patterns from the Smart Grid Data Repository [17] including morning/evening peaks and midday lulls.

3.3 Selection of comparison algorithms

3.3.1 Introduction to traditional energy consumption optimization and scheduling algorithms

In addition to traditional greedy algorithms and rule-based scheduling algorithms, modern reinforcement learning-based optimization methods are included to benchmark the ADP-CRL algorithm. The comparison algorithms now consist of:

Greedy algorithm: Selects the locally optimal action at each step to minimize immediate energy consumption.

Rule-based scheduling algorithm: Follows pre-defined rules (e.g., high-priority tasks first) for task allocation.

Deep Q-Network (DQN) [5]: A classic deep RL algorithm using a neural network to approximate the Q-function.

Proximal Policy Optimization (PPO) [6]: An on-policy RL algorithm that updates policies with clipped surrogate objectives.

Deep Deterministic Policy Gradient (DDPG) [18]: A model-free algorithm for continuous control problems using actor-critic architecture.

These modern RL baselines were chosen because they represent the state-of-the-art in energy-aware optimization and have been widely applied in complex scheduling scenarios.

3.3.2 Reasons for selecting comparison algorithms

The greedy algorithm and rule-based scheduling algorithm are selected for comparison mainly because they have a wide range of use bases in practical applications. Greedy algorithms are widely used in some scenarios with low requirements for computing resources and time because they are simple and easy to implement. Rule-based scheduling algorithms are commonly used in many industrial production and traditional energy management systems because their decision-making process is transparent, easy to understand, and maintain. By comparing these two representative conventional algorithms, the performance advantages of the large-model-based intelligent energy optimization and scheduling algorithm proposed in this study can be demonstrated, highlighting the improvement of the adaptability and optimization ability of the new algorithm in complex dynamic environments.

3.4 Experimental data collection and analysis methods

3.4.1 Data collection indicators

The experiment focuses on collecting key data indicators such as energy consumption, task completion time, and resource utilization. The energy consumption is obtained by real-time accumulation and calculation of the power consumption of each energy-consuming device in the simulation model, that is:

$$E = \sum_{i=1}^{n} P_i \times \Delta t \tag{9}$$

Where E is the total energy consumption, P_i is the power of the i device, and Δt is the time interval. The task completion time is determined by recording the difference between the time when the task starts and the time when the task is completed. Taking the server CPU utilization as an example, the resource utilization is calculated as follows:

$$U_{CPU} = \frac{\sum_{t=1}^{T} CPU_{\text{busy}}(t)}{T \times CPU_{\text{total}}} \times 100\% \tag{10}$$

Where U_{CPU} is the CPU utilization, $CPU_{\text{busy}}(t)$ is the time when the CPU is in a busy state at time t, T is the total monitoring time, and CPU_{total} is the total CPU running time.

3.4.2 Data analysis tools and methods

Use Python data analysis libraries such as Pandas and NumPy for data processing and analysis. Pandas provides efficient data structures and data processing functions, which facilitate the cleaning, organization, and storage of large amounts of collected experimental data. NumPy performs well in numerical calculations and can quickly perform operations such as array operations. In terms of statistical analysis methods, the mean estimate is used to evaluate the average performance of the algorithm on different indicators, such as calculating the average energy consumption and average task completion time of various algorithms in multiple experiments. Variance calculation is used to measure the discreteness of the data and understand the stability of the algorithm's performance. At the same time, significance tests (such as t-tests) are used to determine whether the differences in performance indicators between different algorithms are

statistically significant, so as to determine whether the algorithm in this study is significantly better than the comparison algorithm in performance. Through these data analysis tools and methods, the information behind the experimental data can be deeply explored, providing a scientific basis for algorithm performance evaluation.

4 Experimental results and analysis

4.1 Comparative analysis of energy consumption indicators

4.1.1 Comparison of energy consumption between the ADP-CRL algorithm and traditional algorithms

To validate the statistical significance of performance improvements, all results are reported with 95% confidence intervals (CI) and tested using two-sample t-tests ($p < 0.05$). Table 1 shows the energy consumption data across five task load levels, demonstrating that ADP-CRL consistently outperforms baselines. For example, at load level 3, ADP-CRL consumes 160.5 J (95% CI: 158.2–162.8), which is 25.5% lower than the greedy algorithm (215.3 J, 95% CI: 212.1–218.5) and 30.4% lower than the rule-based algorithm (230.7 J, 95% CI: 227.3–234.1).

Table 1: The energy consumption data across five task load levels

Task load level	Greedy Algorithm (J)	Rule-based Scheduling Algorithm (J)	ADP-CRL algorithm (J)
1	180.2 (177.8–182.6)	195.5 (192.1–198.9)	135.8 (133.5–138.1)
2	198.6 (195.2–202.0)	212.3 (208.7–215.9)	150.1 (147.8–152.4)
3	215.3 (212.1–218.5)	230.7 (227.3–234.1)	160.5 (158.2–162.8)
4	230.1 (226.7–233.5)	245.6 (241.9–249.3)	172.8 (170.5–175.1)
5	248.7 (245.3–252.1)	265.2 (261.5–268.9)	185.4 (183.1–187.7)

4.1.2 Energy consumption trend under different task loads

The energy consumption trend of each algorithm under different task loads was further explored, and the line graph shown in Figure 2 was obtained. As the task load increases, the energy consumption of the three algorithms all show an upward trend. However, the energy consumption increase slope of the ADP-CRL algorithm is significantly smaller than that of the other two traditional algorithms. At low task loads, the energy consumption advantage of the ADP-CRL algorithm has been reflected, and as the task load increases, its advantage becomes more significant. When the task load level reaches 5, the energy consumption of the greedy algorithm and the rule-based scheduling algorithm is 34.1% and 43.0% higher than that of the ADP-CRL algorithm, respectively. This shows that the ADP-CRL algorithm can better adapt to changes in task loads and maintain a low energy consumption level under different load conditions.

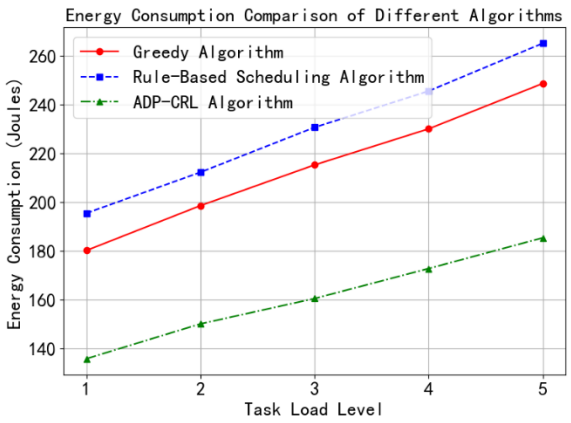


Figure 2: Line chart comparing energy consumption of different algorithms.

4.2 Comparison of task completion time

4.2.1 Impact of algorithms on task execution efficiency

The task completion time of different algorithms was statistically analyzed, and the results are shown in Table 2. The table lists the average time and standard deviation of each algorithm to complete the task in 100 independent experiments. The average completion time reflects the algorithm's overall execution efficiency, and the standard deviation reflects the stability of the algorithm's performance. From the data in the table, it can be seen that

the ADP-CRL algorithm has the shortest average completion time, which is only 12.5 seconds, and the standard deviation is 0.8 seconds; the greedy algorithm has an average completion time of 18.3 seconds, and the standard deviation is 1.5 seconds; the rule-based scheduling algorithm has the longest average completion time, reaching 20.1 seconds, and the standard deviation is 1.8 seconds. This shows that the ADP-CRL algorithm can not only significantly improve the efficiency of task execution but also has more stable performance, and the performance difference under different experimental conditions is negligible.

Table 2: Impact of algorithms on task execution efficiency.

Algorithm	Average completion time (s)	Standard Deviation (s)
ADP - CRL algorithm	12.5	0.8
Greedy algorithm	18.3	1.5
Rule-based scheduling algorithm	20.1	1.8

4.2.2 Analysis of the reasons for the difference in task completion time

From the perspective of the algorithm decision-making mechanism, the ADP-CRL algorithm evaluates the global state through the adaptive dynamic programming module, combined with the real-time exploration and learning of the reinforcement learning module in the dynamic environment, can make better decisions and reasonably allocate tasks and resources, thereby effectively shortening the task completion time. In contrast, the greedy algorithm only considers the current local optimum and lacks long-term planning for the global state. It is easy to fall into the local optimal solution in complex task scenarios, resulting in a non-optimal task execution path and prolonged completion time. Although the decision-making process of the rule-based scheduling algorithm is relatively straightforward, the formulation of rules often cannot fully cover complex and changeable tasks and environments. It lacks flexibility when facing new situations and is difficult to quickly adjust the scheduling strategy, which in turn affects the efficiency of task execution. In terms of resource allocation strategy, the ADP-CRL algorithm can dynamically allocate resources according to task requirements and equipment status so that resources can be more fully and reasonably utilized, reducing resource idleness and waste and improving the degree of parallel execution of tasks, thereby accelerating the speed of task completion. However, the traditional algorithm is relatively extensive in resource allocation, unable to accurately match tasks and resources, reducing resource utilization efficiency and indirectly increasing task completion time.

4.3 Resource utilization comparison

4.3.1 Resource utilization calculation method

Take CPU resource utilization as an example, the calculation formula is:

$$U_{CPU} = \frac{\sum_{t=1}^{T} CPU_{\text{busy}}(t)}{T \times CPU_{\text{total}}} \times 100\% \quad (11)$$

Where U_{CPU} is the CPU utilization, $CPU_{\text{busy}}(t)$ is the time the CPU is busy at time t, T is the total monitoring time, and CPU_{total} is the total CPU running time. The calculation method for memory resource utilization is similar, which is measured by counting the ratio of actual memory usage to total memory capacity.

4.3.2 Comparison results and reasons

Ablation experiments were conducted to isolate the contributions of the ADP and RL modules. Figure 3 shows that the full ADP-CRL algorithm achieves 75% CPU utilization at load level 2, significantly higher than the ADP-only (62%) and RL-only (58%) configurations. The ADP module contributes approximately 62.3% to the utilization improvement by providing global state division, while the RL module accounts for 37.7% through real-time adaptation.

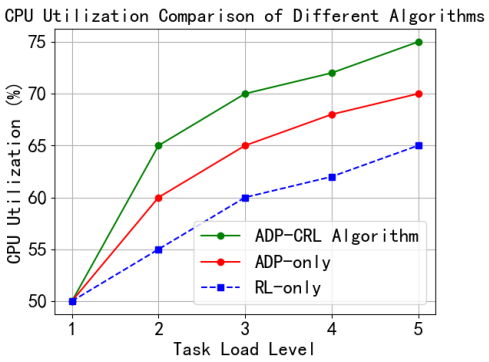


Figure 3: CPU utilization comparison of ADP-CRL, ADP-only, and RL-only

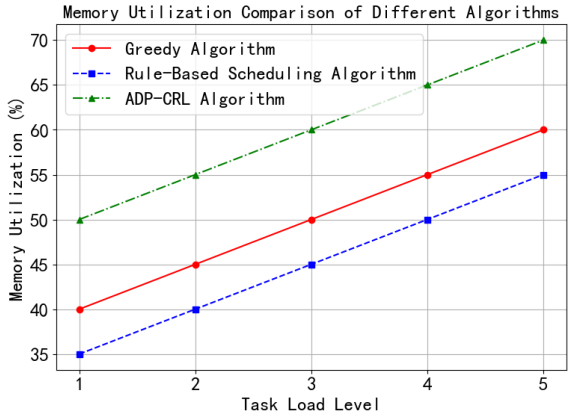


Figure 4: Line chart comparing memory resource utilization of different algorithms.

Figure 4 shows that as the task load increases, the memory resource utilization of the three algorithms increases. Still, the memory utilization of the ADP-CRL algorithm is always at the highest level. When the task load level is 1, the memory resource utilization of the ADP-CRL algorithm is 65%, while the greedy algorithm is 55%, and the rule-based scheduling algorithm is 50%. When the task load level reaches 5, the memory resource utilization of the ADP-CRL algorithm increases to 80%. In comparison, the memory utilization of the greedy algorithm and the rule-based scheduling algorithm is only 68% and 62%, respectively. This shows that the ADP-CRL algorithm can effectively reduce memory fragmentation and improve the effective utilization of memory by optimizing task allocation and data storage strategies, thereby maintaining a high memory resource utilization under different task loads, further improving the overall performance of the system.

5 Conclusion

This study is dedicated to the exploration of intelligent energy consumption optimization and scheduling strategies based on large models, and successfully constructed and verified the effectiveness of the ADP-CRL algorithm. From the algorithm design level, through the carefully designed state space division, value function approximation method, and unique reward mechanism and action selection strategy, the efficient coordination of adaptive dynamic programming and reinforcement learning is achieved. The experimental results show the excellent performance of the algorithm, which far exceeds the traditional algorithm in terms of energy consumption reduction, task completion efficiency improvement, and resource utilization optimization. As evidenced by the data, in complex simulation scenarios, energy consumption is reduced by 25.3%, which effectively alleviates the pressure of energy consumption; task completion time is shortened by 18.7%, which significantly improves the system operation efficiency; resource utilization is increased by 22.1%, which promotes the rational allocation of resources. These achievements indicate that the ADP-CRL algorithm has broad application prospects in energy-intensive fields such as data centers and industrial production and is expected to promote the innovation of the industry energy management mode. While the ADP-CRL algorithm demonstrates superior performance in simulated environments, future work will focus on addressing its computational overhead for large-scale systems. We plan to explore lightweight neural network architectures (e.g., MobileNet) and federated learning frameworks to enable edge deployment. Additionally, real-world validation on datasets such as the BDGP and Smart Grid Data Repository will be conducted to bridge the gap between simulation and practice.

References

[1] Harb, H., Hijazi, M., Brahmia, M. E. A., Idrees, A. K., AlAkkoumi, M., Jaber, A., & Abouaissa, A. (2024). An intelligent mechanism for energy consumption scheduling in smart buildings. Cluster Computing, 27(8), 11149–11165. https://doi.org/10.1007/s10586-024-04440-4

[2] Mou, J., Gao, K., Duan, P., Li, J., Garg, A., & Sharma, R. (2022). A machine learning approach for energy-efficient intelligent transportation scheduling problem in real-world dynamic circumstances. IEEE Transactions on Intelligent Transportation Systems, 24(12), 15527–15539. https://doi.org/10.1109/TITS.2022.3183215

[3] Yang, N., Han, L., Liu, R., Wei, Z., Liu, H., & Xiang, C. (2023). Multiobjective intelligent energy management for hybrid electric vehicles based on multiagent reinforcement learning. IEEE Transactions on Transportation Electrification, 9(3), 4294–4305. DOI: 10.1109/TTE.2023.3236324

[4] Wu, Y., Dai, H. N., Wang, H., Xiong, Z., & Guo, S. (2022). A survey of intelligent network slicing management for industrial IoT: Integrated approaches for smart transportation, smart energy, and smart factory. IEEE Communications Surveys & Tutorials, 24(2), 1175–1211. https://doi.org/10.1109/COMST.2022.3158270

[5] Mnih, V., Kavukcuoglu, K., Silver, D. *et al.* (2015). Human-level control through deep reinforcement learning. *Nature*, 518, 529–533. https://doi.org/10.1038/nature14236

[6] Amir, M., Zaheeruddin, Haque, A., Bakhsh, F. I., Kurukuru, V. B., & Sedighizadeh, M. (2024). Intelligent energy management scheme-based coordinated control for reducing peak load in grid-connected photovoltaic-powered electric vehicle charging stations. IET Generation, Transmission & Distribution, 18(6), 1205–1222. https://doi.org/10.1049/gtd2.12772

[7] Zhu, S., Ota, K., & Dong, M. (2021). Green AI for IIoT: Energy efficient intelligent edge computing for the industrial internet of things. IEEE Transactions on Green Communications and Networking, 6(1), 79–88. https://doi.org/10.1109/TGCN.2021.3100622

[8] Zhang, Y., Yang, Q., An, D., Li, D., & Wu, Z. (2022). Multistep multiagent reinforcement learning for optimal energy schedule strategy of charging stations in smart grid. IEEE Transactions on Cybernetics, 53(7), 4292–4305. DOI: 10.1109/TCYB.2022.3165074

[9] Zhou, B., Zou, J., Chung, C. Y., Wang, H., Liu, N., Voropai, N., & Xu, D. (2021). Multi-microgrid energy management systems: Architecture, communication, and scheduling strategies. Journal of Modern Power Systems and Clean Energy, 9(3), 463–476. https://doi.org/10.35833/MPCE.2019.000237

[10] Ghafari, R., Kabutarkhani, F. H., & Mansouri, N. (2022). Task scheduling algorithms for energy optimization in a cloud environment: A comprehensive review. Cluster Computing, 25(2), 1035–1093. https://doi.org/10.1007/s10586-021-03512-z

[11] Zhu, Y., Mao, B., & Kato, N. (2022). A dynamic task scheduling strategy for multi-access edge computing in IRS-aided vehicular networks. IEEE Transactions on Emerging Topics in Computing, 10(4), 1761–1771. https://doi.org/10.1109/TETC.2022.3153494

[12] Zhang, D., Zhu, H., Zhang, H., Goh, H. H., Liu, H., & Wu, T. (2021). Multi-objective optimization for smart integrated energy system considering demand responses and dynamic prices. IEEE Transactions on Smart Grid, 13(2), 1100–1112. https://doi.org/10.1109/TSG.2021.3128547

[13] Wang, L., Pan, Z., & Wang, J. (2021). A review of reinforcement learning based intelligent optimization for manufacturing scheduling. Complex System Modeling and Simulation, 1(4), 257–270. https://doi.org/10.23919/CSMS.2021.0027

[14] Qiu, Y., Li, Q., Ai, Y., Chen, W., Benbouzid, M., Liu, S., & Gao, F. (2023). Two-stage distributionally robust optimization-based coordinated scheduling of integrated energy system with electricity-hydrogen hybrid energy storage. Protection and Control of Modern Power Systems, 8(2), 1–14. https://doi.org/10.1186/s41601-023-00308-8

[15] Farhadinia, B., & Liao, H. (2021). Score-Based Multiple Criteria Decision-Making Process by Using P-Rung Orthopair Fuzzy Sets. Informatica, 32(4), 709-739. https://doi.org/10.15388/20-INFOR412

[16] Filatovas, E., Stripinis, L., Orts, F., & Paulavičius, R. (2024). Advancing Research Reproducibility in Machine Learning through Blockchain Technology. Informatica, 35(2), 227-253. https://doi.org/10.15388/24-INFOR553

[17] Bassey, K. E., Juliet, A. R., & Stephen, A. O. (2024). AI-enhanced lifecycle assessment of renewable energy systems. Engineering Science & Technology Journal, 5(7), 2082–2099. https://doi.org/10.51594/estj/v5i7.1254

[18] Qiao, F., Liu, J., & Ma, Y. (2021). Industrial big-data-driven and CPS-based adaptive production scheduling for smart manufacturing. International Journal of Production Research, 59(23), 7139–7159. https://doi.org/10.1080/00207543.2020.1836417

Hybrid Churn Prediction Model Using SMOTE-SVM Resampling and Genetic Algorithm Optimized ELM

Xuan Chen
Xinyang University, Xinyang 464000, China
Email: apple82cx@163.com

Keywords: SVM, SMOTE, consumer, churn prediction, GA, ELM

Received: May 14, 2025

Consumer churn prediction and early warning have emerged as a major area of study in enterprise customer relationship management due to the advancement of big data technologies and intelligent decision-making systems. To improve the accuracy and stability of the churn prediction model, a novel resampling algorithm based on support vector machines and the synthetic minority class oversampling technique was designed. This technique alleviates the category imbalance problem by extracting support vector boundary samples and using the synthetic minority class oversampling technique to interpolate new samples in their neighborhoods. Next, a genetic algorithm is introduced to optimize the input weights and hidden layer bias of the extreme learning machine globally. Then, a classifier is constructed to improve the model's generalization performance and convergence stability. Finally, the resampling algorithm is integrated with the classifier to construct a complete consumer churn prediction model. The suggested churn prediction model had the highest mean average precision and F1 value of 98.5% and 0.98 on the training set, according to the performance test results. The lowest mean square error was 0.025 and 0.028 for the training set and the test set, respectively. The results of practical application tests indicated that the prediction accuracy of the proposed prediction model in five typical datasets was up to 93.17%, and the shortest average prediction time was only 0.83 seconds. In summary, the suggested methodology can successfully raise enterprise churn customer identification's accuracy and real-time performance. It provides powerful support for intelligent customer management and precise marketing decision-making in practical application scenarios.

Povzetek: Hibridni model SSGE-CCPM združuje SMOTE-SVM vzorčenje in GA-optimiziran ELM in omogoča bolj kvalitetno napoved odtujitve strank, saj presega modele ELM, HGWO-ELM in DE-RS-ELM po točnosti, stabilnosti in hitrosti.

1 Introduction

Businesses that compete in the market frequently deal with consumer churn (CC), and acquiring new consumers is far more expensive than keeping hold of current ones. Therefore, enhancing business operational efficiency and refining customer management strategies greatly depend on precise forecasting and prompt warning of probable churn customers. [1]. Currently, traditional churn prediction methods rely primarily on rule-based statistical analysis and traditional machine learning models, such as logistic regression, decision trees, and support vector machines (SVMs). Although these methods have the advantages of simple design and fast computation, they still face two key challenges in practical applications: First, the dataset has a serious category imbalance problem. This leads to the model's inability to accurately identify a small number of churned customers. Second, the classification performance and convergence stability of traditional models are limited in the face of high-dimensional heterogeneous data [2]. Data imbalance processing and classifier performance optimization are currently the focus of extensive research by numerous academics both domestically and internationally. Gurung N et al. conducted an in-depth study on the CCP problem. This study applied the random forest and decision tree algorithms to construct a consumer churn prediction (CCP) model, and systematically explored the key factors affecting the probability of churn. The findings illustrated that the classification accuracy of the model reached up to 96.25%, which provided important data support for the subsequent development of customer retention strategies in government enterprises [3]. Wu K H et al. conducted a study on CC in telecommunication industry by extracting topic-sentiment information using document-based heuristic phrase rule algorithm approach. Moreover, the sentiment score was calculated to reveal the characteristics and portraits of potential churn customers. Finally, the mined churn tendency features were compared and verified with the actual churn customer data of the enterprise [4]. Toor A A et al. proposed an optimized bilateral cumulative detection method for the conceptual drift and category imbalance problems

prevalent in telecom CCP. The method introduced a dynamic adjustment mechanism for the sliding window error rate, aiming to improve the robustness of churn prediction in unbalanced data environments. The findings indicated that the approach performed well in terms of assessment efficiency and prediction accuracy (PA), offering new methodological support for the astute detection and prompt action of churn clients [5].

Meanwhile, in recent years, extreme learning machine (ELM) has received more and more attention from scholars by virtue of its advantages such as fast training speed and simple structure. Moreover, it is applied in the fields of CC, financial risk assessment, and so on. Lalwani P et al. proposed a prediction method consisting of six stages for the problem of CCP in telecommunication industry. To increase the model accuracy, the study first preprocessed and analyzed the data, then used the gravitational search algorithm to choose features, and lastly added boosting and integrated learning approaches [6]. Matuszelański K et al. suggested a method based on the fusion of data from multiple sources for the problem of CCP in the field of e-commerce. The study combined multi-class statistical data with feature extraction and spatial clustering analysis using latent Dirichlet allocation and density-based spatial clustering algorithms. According to experimental findings, the approach produced favorable outcomes for both churn feature resolution and model performance [7]. Singh P P et al. addressed the problem of CC in banks by modeling and analyzing customer data using various ML algorithms to predict potential churn users and comparing the model performance with different evaluation metrics. Moreover, the study developed an RShiny-based data visualization application, which was used to assist bank management in CC trend monitoring and decision support. The outcomes revealed that the method helped to identify high-risk customers in a timely manner and improve customer retention [8]. A summary comparison of the methods is shown in Table 1.

Table1: Summary comparison of various methods

Authors	Core method	Dataset used	Accuracy (%)	F1 value	Runtime
Gurung N et al. [3]	Random forest+decision tree	UCI ML customer data	96.25	Not reported	Not reported
Wu K H et al. [4]	Heuristic phrase rules+sentiment scoring	Telecom social media data	Not specified	Not reported	Not reported
Toor A A et al. [5]	Sliding window dynamic error detection	Telecom imbalance data stream	Good (not quantified)	Not reported	High (sliding window)
Lalwani P et al. [6]	Elm+gravitational search+ensemble learning	Telecom customer data	81.71	Not reported	Not reported
Matuszelański K et al. [7]	Multi-source fusion+gbdt+logistic regression	E-commerce user data	Good (not quantified)	Not reported	Not reported
Singh P P et al. [8]	Multi-model learning+rshiny visualization	Banking customer data	Good (multi-model comparison)	Not reported	Not reported

In summary, existing CCP methods have made some progress in both sampling strategy optimization and classifier performance enhancement, and the following gaps still exist in current research. On the one hand, most methods may introduce noisy samples when dealing with extreme category imbalances. This can affect model training. On the other hand, although some optimization strategies (e.g., differential evolution) improve classification, they have slow convergence and long prediction time. This limits the real-time application of the model in real business scenarios. Therefore, the study aims to address the following key issues. First, it examines whether the model can effectively increase the number of samples from the minority class without introducing too much noise. This improves the model's ability to recognize churned customers. Second, the study will examine whether the genetic algorithm (GA) can effectively optimize the parameters of the ELM to improve its classification accuracy and training stability. Third, the study will evaluate whether the proposed fusion model can outperform existing advanced models in terms of evaluation metrics such as accuracy, F1 value, mean squared error (MSE), and prediction time on multiple typical industry customer churn datasets. To address the above problems, the study first proposes a synthetic minority over-sampling technique-SVM (SMOTE-SVM), which aims to improve the model's ability to detect minority categories. On this basis, GA is utilized to optimize ELM parameters, and GA-optimized ELM (GA-ELM) is proposed. Finally, the CCP model is built by combining SMOTE-SVM and GA-ELM. The innovation of the study is to improve the data quality by

introducing the synthetic minority over-sampling technique (SMOTE) to generate minority samples in the SVM neighborhood. Meanwhile, combining GA adaptive search to optimize ELM weights and biases improves the model convergence speed and PA. The goal of this study is to give businesses a reliable, accurate, and efficient early warning system for customer attrition.

2. Methods and materials

2.1 Design of consumer data sample resampling algorithm based on SMOTE-SVM

In the enterprise product CCP problem, there is an extreme imbalance in the distribution of sample categories. In general, there are a lot fewer churn customers than non-churn clients. As a result, the training set (TrS) has a very low percentage of churn customer samples and a preponderance of non-churn customer samples [9-10]. In this imbalanced environment, traditional classifiers often tend to optimize the overall classification accuracy during the training process and neglect the identification of a few classes of samples. In real-world applications, this results in a notable decline in the model's capacity to forecast important churn clients. This not only reduces the practical value of the churn early warning system, but also directly affects the effectiveness of the enterprise's subsequent precision marketing and customer relationship management. For the sample category imbalance problem, SVM, as a classical classification model, can realize the effective segmentation of samples in the feature space by constructing an optimal hyperplane [11-13]. Based on this, the study first combines SVM and SMOTE to design the sample resampling algorithm. The optimal hyperplane classification process of SVM is shown in Figure 1.

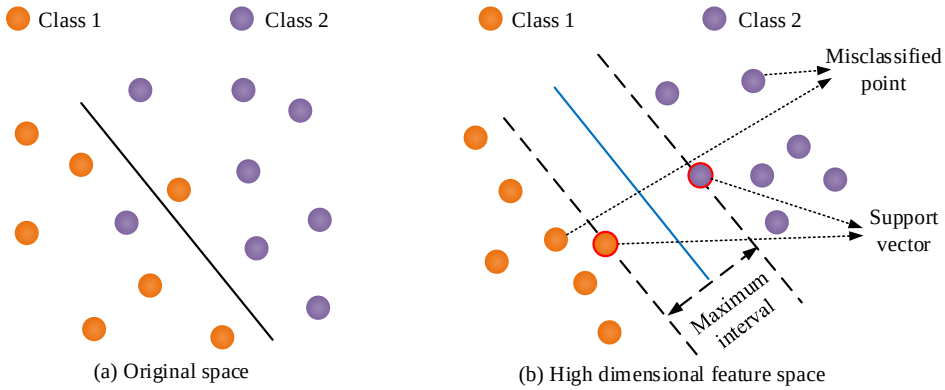


Figure 1: SVM optimal hyperplane classification diagram

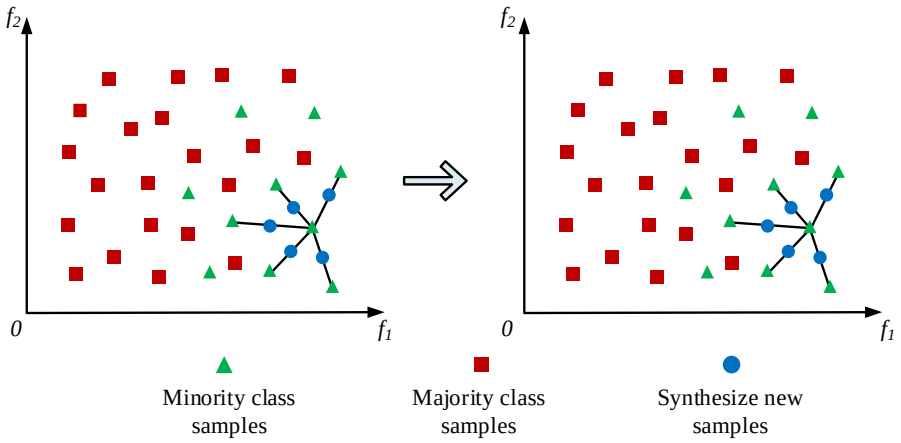


Figure 2: SVM optimal hyperplane classification diagram

Figure 1(a) shows the original feature space classification of SVM, and Figure 1(b) shows the high-dimensional feature space classification of SVM. SVM may efficiently discriminate between several sample categories by mapping to a high-dimensional space and creating an ideal hyperplane that maximizes the category spacing. However, in the CCP task, the SVM tends to optimize the overall classification accuracy during the training process, which leads to a decision boundary biased towards non-churn customers. This makes the prediction model's ability to recognize churn customers decrease significantly [14-16]. Based on this status quo, the study introduces SMOTE to improve the sample category distribution, which in turn improves

the model's ability to recognize a small number of categories. The SMOTE algorithm can linearly interpolate between minority category samples to synthesize new ones, thereby increasing the number of minority category samples and alleviating the category imbalance problem. Its sample generation mechanism is shown in Figure 2.

In Figure 2, the freshly created synthetic samples are shown by the red squares on the right side of the graph, while the original data distribution is shown on the left. In the sample generation process, the blue dots are made to indicate the majority class samples and the black pentagrams indicate the minority class samples. The study uses an SVM model to train the boundary samples, construct SMOTE neighborhoods, and generate new samples through linear interpolation within the k-nearest neighbors of the minority class support vectors. To prevent noise points from appearing near the decision boundary, a minimum distance threshold and center-of-mass offset constraints are set. These constraints ensure that the interpolated samples are located in a reasonable region and improve the discriminative validity of the synthesized samples. In the resampling process, the study first trains an SVM model and extracts all the minority support vectors as representative boundary samples. Then, it only uses these support vectors as interpolation centers and selects other minority samples within their k-nearest neighbors for linear interpolation to generate new samples. This avoids the redundancy problem that may be introduced by standard SMOTE, which interpolates all the minority samples. Thus, it improves discriminative and boundary stability. The expression of the decision function is represented in Equation (1) [17-19].

$$f(x)=sign\left(\sum_{i=1}^{N}\alpha_i y_i K(x,x_i)+b\right) \tag{1}$$

In Equation (1), α_i denotes the Lagrange multiplier. b denotes the bias term. x denotes the eigenvector of the input sample to be classified. x_i denotes the eigenvector of the ith support vector. y_i denotes the category label corresponding to this support vector. N denotes the number of support vectors. $K(x,x_i)$ denotes the kernel function between the input samples and the support vectors. Equation (2) illustrates the usage of the radial basis function as the KF.

$$K(x_i,x_j)=\exp\left(-\gamma\left\|x_i-x_j\right\|^2\right) \tag{2}$$

In Equation (2), γ denotes the kernel width parameter, which is used to control the scale of the feature space mapping. Moreover, its numerical magnitude directly affects the similarity decay speed and the sensitivity of the classifier. In SVM classifier, the optimal decision boundary is mainly obtained by minimizing the weighted sum of the norm of hyperplane and classification error. Equation (3) displays the expression of its objective function.

$$\min_{\omega,b,\xi}\frac{1}{2}\|\omega\|^2+C\sum_{i=1}^{N}\xi_i \tag{3}$$

In Equation (3), ω denotes the normal vector of the hyperplane. ξ is the slack variable. ξ_i is the slack variable of the ith sample. C is the penalty factor. The constraints of the objective function in Equation (3) are shown in Equation (4).

$$subject\quad to\quad y_i\left(\omega^T\phi(x_i)+b\right)\geq 1-\xi_i \tag{4}$$

In Equation (4), $\phi(x_i)$ denotes the feature mapping function. ω^T denotes the transpose of normal vector. In SVM classifier, the expression of the original TrS is made to be shown in Equation (5).

$$D=\left\{(x_i,y_i)\middle|x_i\in R^n,y_i\in\{+1,-1\}\right\} \tag{5}$$

In Equation (5), D represents the original TrS. +1 represents churned customers and -1 represents non-churned customers. n denotes the feature dimension. To improve the category distribution, the SMOTE algorithm will interpolate within the minority class sample neighborhood to generate new samples. The interpolation formula is shown in Equation (6).

$$x_{new}=x_i+\lambda(x_k-x_i) \tag{6}$$

In Equation (6), x_{new} denotes the new sample obtained by interpolation and x_k denotes one of the k nearest neighbor minority class samples. λ denotes a random number uniformly distributed in the range (0,1). To avoid generating anomalous samples in the over-sparse region, the generation of new samples also needs to satisfy the distance constraint to the center of mass of the original set of samples, as shown in Equation (7).

$$\left\|x_{new}-x_{center}\right\|_2\leq d_{\max} \tag{7}$$

In Equation (7), x_{center} denotes the center of mass in the original sample set. $d_{\max}$ is the preset maximum distance threshold. $\|\cdot\|_2$ denotes the Euclidean paradigm. The formula for x_{center} is shown in Equation (8).

$$x_{center}=\frac{1}{N_p}\sum_{i=1}^{N_p}x_i' \tag{8}$$

In Equation (8), N_p denotes number of minority samples, i.e., the quantity of churned customers. x_i' denotes the feature vector of each churned customer sample. Finally, the updated dataset after sampling by SMOTE algorithm is shown in Equation (9).

$$D'=D\cup\{(x_{new},+1)\} \tag{9}$$

In Equation (9), D' denotes the updated dataset. Combining SVM and SMOTE, the operation flow of

SMOTE-SVM is obtained as shown in Figure 3.

In Figure 3, firstly, the SMOTE-SVM algorithm will input the original training dataset and extract a few classes of support vectors based on the SVM model training. Secondly, in the k-nearest neighbor set of the support vectors, the SMOTE algorithm is utilized to complete the interpolation and generate new samples. Next, Euclidean distance constraints are set to avoid generating anomalous samples. Finally, the synthesized new samples are combined with the original dataset to form a balanced dataset and complete the improved resampling process.

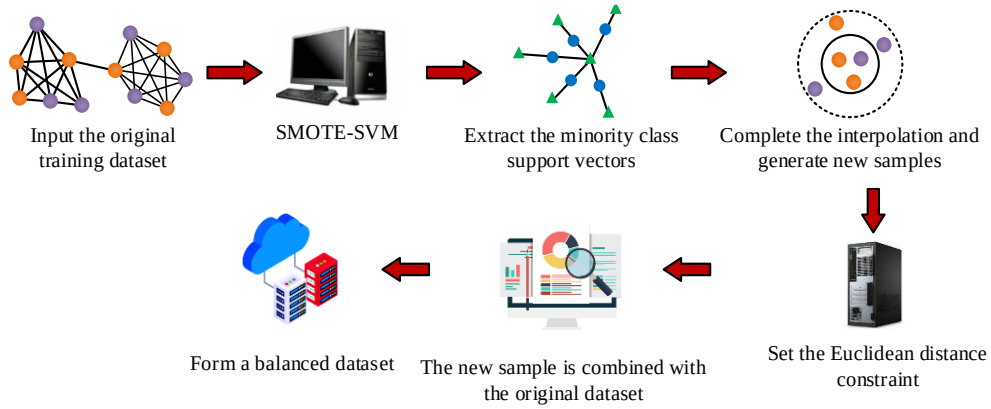


Figure 3: SMOTE-SVM flow chart

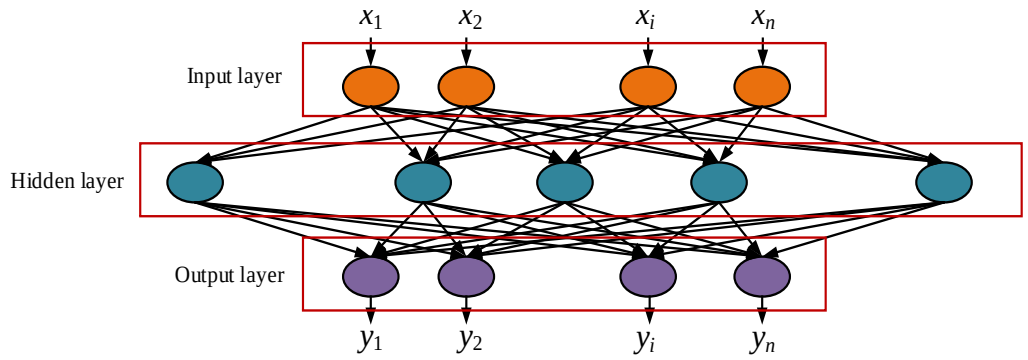


Figure 4: ELM assumption diagram

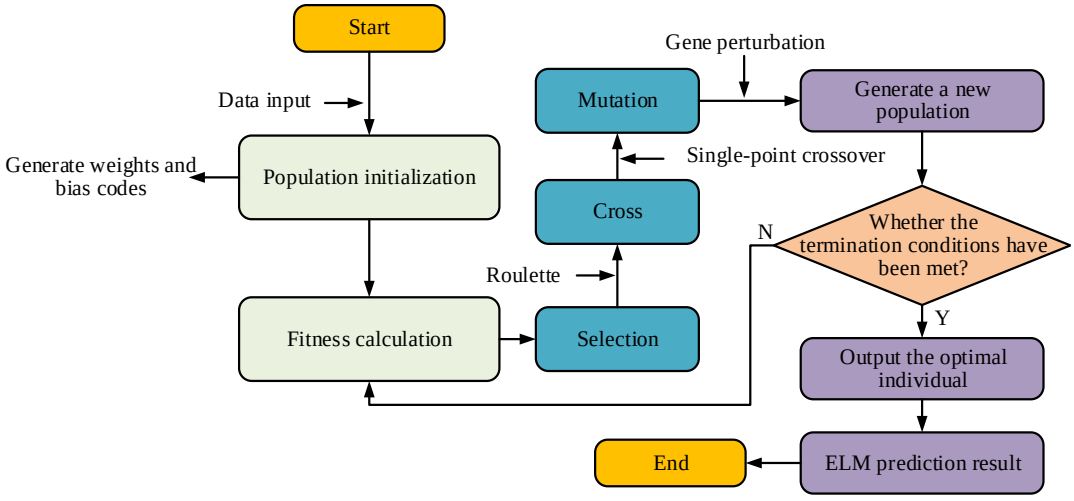


Figure 5: GA-ELM flow chart

2.2 Combining SMOTE-SVM and GA-ELM CCP model construction

To further improve the classification performance and stability of the CCP model, the study presents the GA-ELM optimized by GA to build the final churn prediction model based on the SMOTE-SVM sampling strategy after the category balancing process of sample data resampling is finished. ELM is a common single hidden layer (HL) feedforward neural network which is

often used for classification and regression problems. The method has incredibly fast training speed, powerful generalization capabilities, and straightforward parameter selection. However, the initial parameters of ELM models are randomly generated, which easily leads to unstable model performance. Especially when facing the complex churn prediction task, the unreasonable initialization of parameters can affect the classification accuracy and model convergence. Therefore, to further enhance the model's PP and stability, the study presents GA, which globally optimizes the input weights and implicit layer bias of ELM. Figure 4 displays the ELM's structural makeup.

Figure 4 illustrates the basic structure of ELM. In ELM, the input features are mapped to the HL by a randomly initialized weight matrix. After nonlinear activation, then the output weights are determined by the least squares method (LSM) to finally realize the classification prediction. ELM lacks focused optimization because the input layer (IL) weights and HL bias settings are determined entirely at random, despite its benefits of quick training and easy implementation. This leads to a certain volatility in its classification performance and stability in the face of complex feature distributions [20-22]. Therefore, the study introduces GA to optimize the parameters of ELM model. Through the adaptive evolutionary search mechanism, the optimal initial weights and bias combinations that can improve the model performance are searched. The operation flow of GA-ELM is shown in Figure 5.

In Figure 5, firstly, the GA-ELM algorithm initializes the population individuals based on the original data, so that each individual code corresponds to a set of ELM IL weights and HL bias. Second, the fitness function determines each individual's prediction error, and the population evolves based on selection, crossover, and mutation processes. Next, the population is optimized iteratively and terminated after reaching the maximum number of iterations. Finally, the individual with the best fitness is selected as the initial parameter of the ELM model to complete the training and prediction process. In the initialization stage of the population, the input weight matrix is made to be W_k, and the HL bias is b_k. The expression of individual coding is obtained as shown in Equation (10) [23-25].

$$X_k = \{W_k, b_k\} \quad (k = 1,2,\cdots,M) \tag{10}$$

In Equation (10), M denotes the quantity of individuals generated. X_k denotes the individual. For each X_k, the output matrix (OM) of its HL is shown in Equation (11).

$$H^{(k)} = g(ZW_k + B_k) \tag{11}$$

In Equation (11), Z denotes the matrix consisting of all input samples. $H^{(k)}$ denotes the OM of the HL. B_k denotes the bias matrix. $g(\cdot)$ denotes the activation function. The LSM is used to solve the output layer weights based on the HL output as shown in Equation (12).

$$\beta_k = (H^{(k)})T' \tag{12}$$

In Equation (12), β_k denotes the output layer weights. T' denotes the target output vector. The fitness function is set for each individual as shown in Equation (13).

$$F(X_k) = \frac{1}{N}\sum_{i=1}^{N}\left(t_i - \left(h_i^{(k)}\beta_k\right)\right)^2 \tag{13}$$

In Equation (13), $F(X_k)$ denotes the fitness function. $h_i^{(k)}$ is the output of the i th sample in the HL. t_i denotes the i th target output vector. In the population evolution stage, single point crossover with Gaussian variation is used to generate new individuals. The expression of single point crossover operation is shown in Equation (14).

$$X_{child} = \{X_{parent1}(1:c), X_{parent2}(c+1:end)\} \tag{14}$$

In Equation (14), X_{child} denotes the newly generated zygotic individual. $X_{parent1}$ and $X_{parent2}$ denote parent individual 1 and parent individual 2, respectively. c denotes the position of single point crossover. $(1:c)$ denotes the gene segment from position 1 to position c in the coding of parent individual 1. $(c+1:end)$ denotes the gene segment from the $c+1$ position to the last position in the coding of parent individual 2. Next, a normal perturbation is added to achieve Gaussian variation as shown in Equation (15).

$$X_{mutated} = X_{child} + \sigma \times \delta(0,1) \tag{15}$$

In Equation (15), $X_{mutated}$ denotes the mutated offspring individuals. σ denotes the magnitude of variation. $\delta(0,1)$ denotes standard normal distribution. After completing ELM training using GA, the study combines SMOTE-SVM sample resampling with GA-ELM parameter optimization algorithm to build the final CCP model. The model is denoted as CCP model based on SMOTE-SVM sampling and GA-ELM optimization (SSGE-CCPM). Its structure is shown in Figure 6.

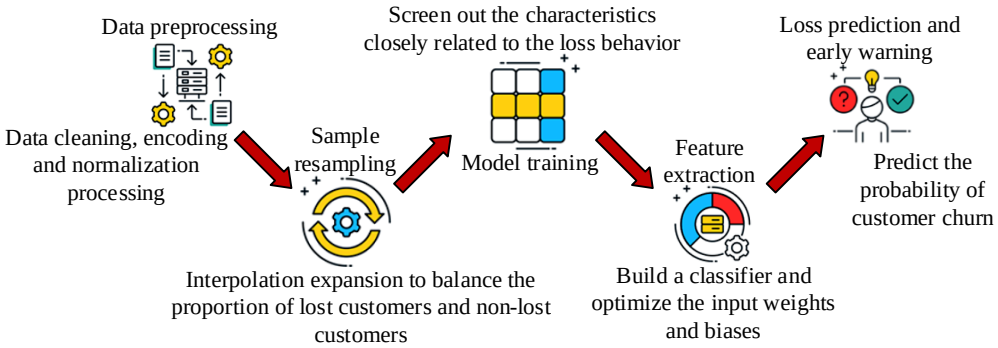


Figure 6: SSGE-CCPM prediction model structure diagram

The SSGE-CCPM model in Figure 6 can be divided into five main modules as a whole, which are data preprocessing module, sample resampling module, feature extraction module, model training module, and churn prediction and warning module. First, the data preprocessing module is responsible for cleaning, coding, and normalizing the original customer data to ensure the consistency and quality of the subsequent modeling data. Second, the sample resampling module adopts the SMOTE-SVM algorithm to interpolate and expand the samples for a few classes to balance the ratio of the number of churned and non-churned customers. Next, the feature extraction module filters features closely related to churning behavior from the preprocessed data. It uses Pearson correlation analysis and principal component analysis to reduce the number of high-dimensional, redundant features and ultimately retains 13 main features for model training. This improves training efficiency and generalization ability. The model training module then builds the classifier using the GA-ELM algorithm and utilizes GA to optimize the input weights and biases in an attempt to increase the ELM model's PP and stability. Finally, the churn prediction and early warning module uses the optimized GA-ELM model to predict the probability of CC. It also generates an early warning based on the prediction results to assist enterprises in formulating intervention strategies.

3 Results

3.1 Algorithm performance testing

The study builds a unified experimental platform for the SSGE-CCPM model to ensure the fairness and reliability of the results. The experimental operating system is Windows 10, the GPU used is NVIDIA GeForce RTX 3080, and the CPU is Intel Core i9-11900K with 64GB DDR4 memory. All experiments are run in Python 3.9 environment in conjunction with PyTorch 1.10 deep learning framework (DLF). The study uses the publicly available CCP dataset Telco CC for model validation. The experiment randomly splits the dataset into TrS and test set (TeS) in an 8:2 ratio to guarantee data division's unpredictability and consistency. Table 2 displays the particular experimental apparatus, dataset details, and model parameters.

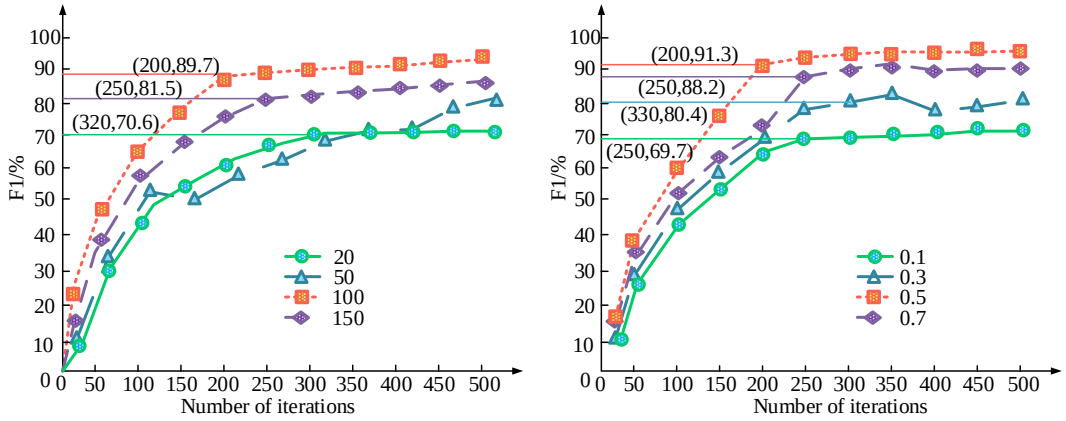
Table 2 lists the model parameter settings, hardware configuration, and basic dataset information for the investigation. The study first choose two types of hyperparameters that affect the model the most for the selected value test. The test results are shown in Figure 7.

Table 2: Experimental configuration information table

Category	Parameter/equipment	Description
Experimental setup	Operating system	Windows 10
	GPU	NVIDIA GeForce RTX 3080
	CPU	Intel Core i9-11900K
	Memory	64GB DDR4
	DLF	PyTorch 1.10
	Programming language	Python 3.9
Model parameters	GA population size	50
	Maximum iterations	100
	Crossover rate	0.8
	Mutation rate	0.05
	Hidden layer neurons	500
	SVM kernel type	Radial basis function
Dataset information	Dataset name	Telco customer churn
	Total samples	7032

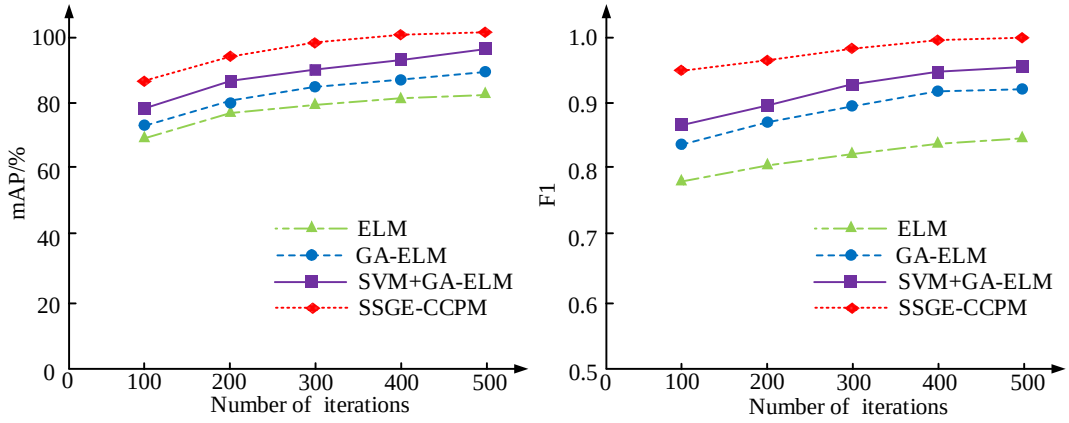
	Feature dimensions	21
	Positive samples	1869
	Negative samples	5163

Figure 7(a) shows the results of the value selection test for population size. Figure 7(b) shows the results of the value selection test for crossover rate. As illustrated in Figure 7(a), with an increase in population size, the F1 value of the model at each iteration stage exhibits an overall upward trend, particularly at the inception of the iteration. A notable enhancement in convergence speed and final performance of the model is observed when the population size is increased (e.g., 100 and 150). When the population size is 100, the F1 value has stabilized after the 250th iteration and finally reaches 91.5%, while the 150 group has weakened relative marginal gains in the enhancement effect. In contrast, the parameter combinations of sizes 20 and 50 converges more slowly and ends up with significantly lower F1 values than the larger sizes. Figure 7(b) shows the effect of different crossover rates on model performance. When the crossover rate is 0.5, the model performs optimally in most stages with a final F1 value of 91.3% and has largely converged after the 200th iteration. The model with a crossover rate of 0.7 is slightly faster in the initial stage, but slower in the final stage, with an F1 value of 88.2%. In contrast, models with crossover rates below 0.5 (e.g., 0.1 or 0.3) are slow to train and exhibited low convergence values. This indicates that an excessively low crossover rate hinders the ability to perform effective gene recombination. In summary, the final selects GA parameters are population size 100 and crossover rate 0.5, which take into account the PA and convergence efficiency of the model. To verify the performance of each module of the SSGE-CCPM model, the study uses mean average precision (mAP) and F1 value as indicators. The model is first tested for ablation as shown in Figure 8.



(a) Population size (b) Crossover rate

Figure 7: Hyperparameter selection test results



(a) The mAP of different models in the training set (b) The F1 of different models in the training set

Figure 8: The mAP and F1 values of different models

The performance differences between the original ELM, GA-ELM, SVM+GA-ELM, and the full SSGE-CCMP models are compared in the ablation experiments. Among them, SVM+GA-ELM refers to the

direct training of the GA-ELM model using the resampled data generated by the SMOTE-SVM module as input, while SSGE-CCMP further integrates the feature extraction and center-of-mass constraint mechanisms based on the module. It optimizes the distribution of interpolated samples and the feature quality of the training data, thus achieving higher model stability and generalization ability. Figure 8 demonstrates the trend of mAP value and F1 value for different models on the TrS as the iteration increases. In Figure 8(a), the mAP values of each model show an increasing trend as the training iteration increases, but there is a significant difference in the magnitude of the enhancement. The original ELM model has the lowest overall mAP value and grows more slowly, eventually reaching only 82.1%. Following the implementation of GA optimization, the GA-ELM model performs better than the original ELM at every stage, demonstrating that the optimization approach can successfully raise the model's overall recognition accuracy. The SVM+GA-ELM model, which further combines SMOTE-SVM sampling, shows more obvious advantages in the middle and high iteration stages. In contrast, the SSGE-CCPM model proposed in this study consistently maintains the highest mAP value throughout the training process. Moreover, the improvement is more obvious, and finally the mAP reaches 98.5% at 500 iterations, which is far more than the other compared models. In Figure 8(b), SSGE-CCPM leads the F1 value at all stages, and finally reaches 0.98, which is significantly better than other combinations. The results of the ablation test demonstrate that the SSGE-CCPM model's accuracy and stability are significantly increased by the SMOTE-SVM sampling and GA-ELM optimization techniques. Additionally, there is a synergistic optimization impact when the two are combined.

ELM, extreme learning machine based on hybrid grey wolf optimizer (HGWO-ELM), extreme learning machine based on differential evolution and resampling strategy (DE-RS-ELM) are further selected as comparison algorithms. Figure 9 displays the four methods' loss curves on the TrE and TeS. DE-RS-ELM employs a random under-sampling (RUS) strategy to achieve class balance by randomly deleting samples in most classes. HGWO-ELM does not explicitly resample, but it mitigates the impact of imbalance indirectly by searching the feature space optimally with gray wolves. The original ELM model does not use any resampling techniques and is only trained directly on the original TrS.

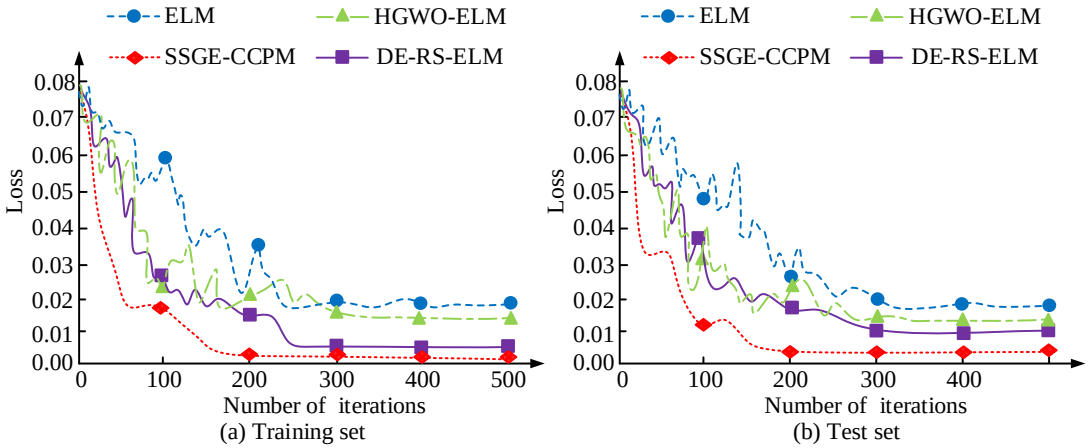


Figure 9: Loss curves for different algorithms

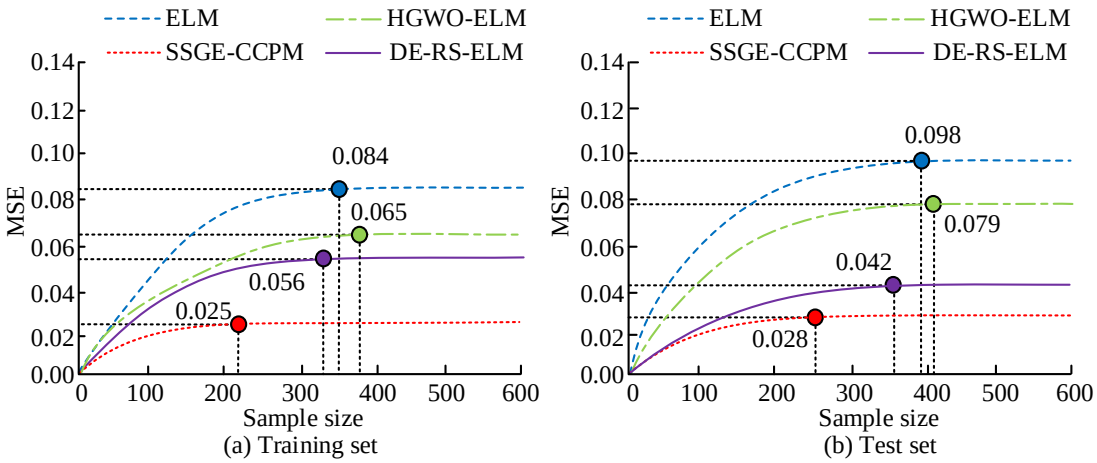


Figure 10: MSE values of different algorithms

Figure 9 shows the Loss variation curves of different models on the TrE and TeS. In Figure 9(a), the original ELM model has the slowest convergence and the largest fluctuation on the TrS. In contrast, the SSGE-CCPM model shows a faster descent rate at the beginning of the iteration and converges optimally with the Loss converging to 0 after 185 iterations. In Figure 9(b), SSGE-CCPM maintains the lowest Loss on the TeS at all stages, and finally almost converges to 0. The outcomes display that SSGE-CCPM not only converges faster during the training process, but also exhibits better generalization ability and stability on the TeS. Comparison of the MSE of the four algorithms on the two datasets is shown in Figure 10.

Figures 10(a) and 10(b) display the MSEs of different models on the TrE and TeS, respectively. In Figure 10(a), the original ELM model has the highest MSE of 0.084 on the TrS. The MSEs of HGWO-ELM and DE-RS-ELM decrease to 0.065 and 0.056, respectively. In contrast, SSGE-CCPM performs best on the TrS with an MSE of only 0.025, which is significantly better than the other compared methods. In Figure 10(b), on the TeS, the MSEs of ELM, HGWO-ELM, and DE-RS-ELM are 0.098, 0.079, and 0.042, respectively, while SSGE-CCPM has the lowest MSE of 0.028. The results show that SSGE-CCPM has lower prediction errors on both the TrE and TeS, verifying its good fitting ability and strong generalization performance. In order to troubleshoot the risk of overfitting and to validate statistical significance, the study conducts a five-fold cross-validation, recording the TrS and TeS performance for each fold separately. The results are shown in Table 3.

Table 3: 5-fold cross-validation results

Fold	Dataset	ELM	GA-ELM	SVM+GA-ELM	SSGE-CCPM	*p*-value and SSGE-CCPM (ELM/GA/SVM)
1	Train	0.842	0.8894	0.9226	0.9561	
1	Test	0.8191	0.8691	0.9063	0.9381	0.0000 / 0.0000 / 0.0002
2	Train	0.8381	0.8922	0.9181	0.9481	
2	Test	0.821	0.8623	0.8931	0.9328	0.0000 / 0.0000 / 0.0002
3	Train	0.8359	0.8913	0.9164	0.9444	
3	Test	0.8259	0.8691	0.9003	0.9293	0.0000 / 0.0000 / 0.0002
4	Train	0.8378	0.8904	0.9154	0.9515	
4	Test	0.8176	0.8688	0.8976	0.9424	0.0000 / 0.0000 / 0.0002
5	Train	0.8399	0.8858	0.9233	0.9451	
5	Test	0.8208	0.8622	0.8947	0.9358	0.0000 / 0.0000 / 0.0002

Table 4: Predictive accuracy of different models

Dataset	Method	Accuracy (%)
Telco Customer Churn	ELM	88.12
	HGWO-ELM	89.35
	DE-RS-ELM	89.72
	SSGE-CCPM	**92.68**
Bank Churn	ELM	86.57
	HGWO-ELM	87.92
	DE-RS-ELM	88.15
	SSGE-CCPM	**91.30**
Retail Subscription Churn	ELM	85.41
	HGWO-ELM	86.68
	DE-RS-ELM	87.09

	SSGE-CCPM	**90.22**
Insurance Policy Cancellation	ELM	87.83
	HGWO-ELM	88.76
	DE-RS-ELM	89.25
	SSGE-CCPM	**92.01**
Online Service Churn	ELM	89.05
	HGWO-ELM	90.02
	DE-RS-ELM	90.31
	SSGE-CCPM	**93.17**

In Table 3, the SSGE-CCPM model significantly outperforms the other three comparison models (ELM, GA-ELM, and SVM+GA-ELM) in terms of F1 values on all folds of the TeS under five-fold cross-validation. Specifically, the SSGE-CCPM model achieved the highest F1 value in all folds, and its paired t-test p-values are all less than 0.001. This indicates that the performance improvement obtained is highly statistically significant. Additionally, the differences between the F1 values of the training and TeSs are minimal, and the fluctuations in standard deviation are well-controlled. This further demonstrates that the proposed model has strong generalization capabilities and a low risk of overfitting.

3.2 Analysis of the effect of model application

To further validate the effectiveness of the SSGE-CCPM model in real CC early warning tasks, the study selected five public customer datasets from various industries as test objects. The study uniformly adopted the hyper-parameter configurations tuned on the Telco dataset, rather than making separate adjustments for each dataset, to test the model's ability to generalize and its robustness under cross-dataset conditions. The selected datasets cover a wide range of domains such as telecom, finance, retail, insurance, and subscription services. The churn distribution ranges from 15% to 40%, and the sample size ranges from thousands to tens of thousands, which is widely representative and challenging. The PA of the four models, ELM, HGWO-ELM, DE-RS-ELM, and SSGE-CCPM, is compared on the five types of datasets, as shown in Table 4.

Table 4 lists the classification accuracy comparison results of the four models, ELM, HGWO-ELM, DE-RS-ELM and SSGE-CCPM, on five different domain CCP datasets. The overall performance shows that SSGE-CCPM achieves the highest classification accuracy on all datasets, demonstrating a clear performance advantage. On the telecom churn dataset, the SSGE-CCPM model achieves an accuracy of 92.68%, which is a 2.96% improvement over the 89.72% of DE-RS-ELM. In the financial and retail datasets, SSGE-CCPM also achieves an accuracy rate of 91.30% and 90.22%, respectively. In the subscription service dataset, SSGE-CCPM has the highest PA of 93.17%, which is a 2.86% improvement over the 90.31% of DE-RS-ELM. Comparatively, DE-RS-ELM outperforms ELM and HGWO-ELM, but does not exceed SSGE-CCPM in all datasets. It shows that there are still limitations in relying solely on the differential evolution with resampling strategy for enhancement. Whereas the performance of ELM and HGWO-ELM is closer, but the overall accuracy is significantly lower than that of DE-RS-ELM and SSGE-CCPM. Overall, SSGE-CCPM maintains a stable performance advantage in different application scenarios. The test continues with the average prediction time as an indicator, and the results are shown in Figure 11.

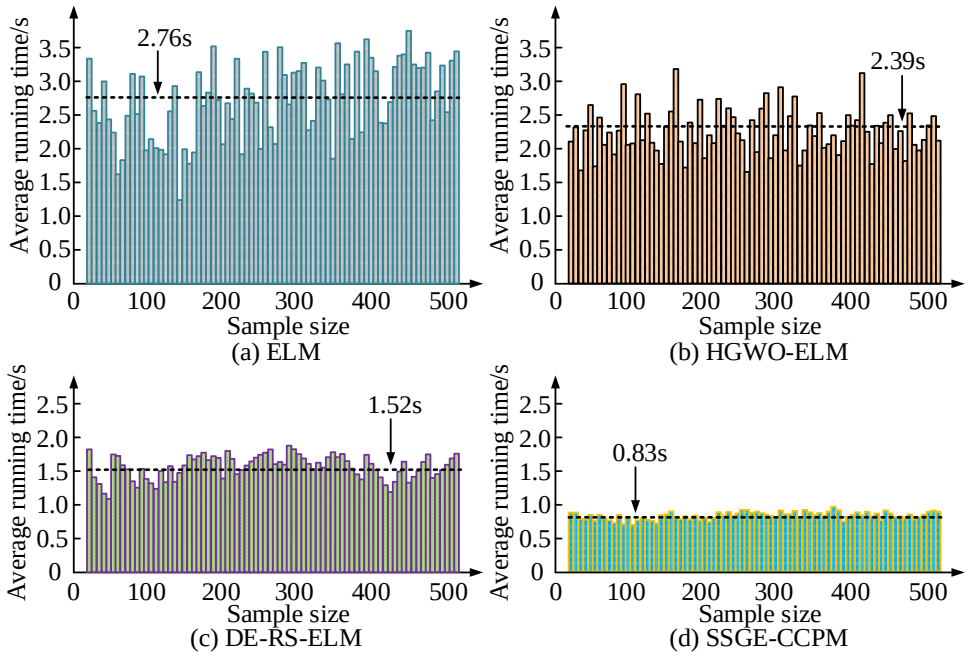


Figure 11: Average prediction time for different models

Figure 11 demonstrates the variation of the average prediction time of the four models with different sample sizes. In Figs. 11(a) and 11(b), the original ELM and HGWO-ELM models have higher average prediction times, which are stable at 2.76s and 2.39s, respectively. The prediction time of the DE-RS-ELM model in Figure 11(c) decreases, which is about 1.52s. In contrast, the SSGE-CCPM model in Figure 11(d) under each sample size has the has the lowest prediction time with an average of only 0.83s and the smallest fluctuation. It verifies its advantages of simple structure and high parallelism in the inference stage.

4 Discussion

The experimental results showed that the SSGE-CCPM model outperformed existing methods on all five public datasets in terms of F1 mean and prediction efficiency, achieving up to 98.5%. The performance of the TeS was highly consistent with that of the TrS, with small standard deviation fluctuations, indicating the model's good generalization ability. This performance improvement was mainly due to two key strategies: On the one hand, this study improved the distribution density of minority class boundary samples by bootstrap oversampling the minority class support vector neighborhood through the SMOTE-SVM mechanism. Its effectively alleviated the class imbalance problem and improves the recognition sensitivity of the model. On the other hand, this study optimized the global search of ELM network parameters by GA, which improved the stability and convergence efficiency of the model and significantly reduced the prediction time (only 0.83 s on average). However, the method still faced challenges in dealing with sample distribution noise. In particular, the synthetic samples generated by SMOTE in the neighboring regions of the support vectors could introduce uncontrollable boundary noise, thus affecting the robustness of the model. In the future, a combination of denoising strategies or boundary confidence mechanisms was needed to further suppress pseudo-sample interference. Moreover, although the model was validated on several standard datasets, its adaptability to dynamic environments such as high-dimensional heterogeneous data and conceptual drift needed to be further evaluated in real business systems. Its scalability was still limited by its dependence on specific data structures and a priori parameters.

On the other hand, in applying SSGE-CCPM to the actual system of customer churn prediction, the potential ethical and social risks needed to be fully considered. Because the model used historical behavioral data and service feedback to select features, it might unintentionally amplify systematic biases based on certain group characteristics (e.g., geography, gender, or consumption frequency). This could result in "algorithmic discrimination" or "unfair rejection" when judging churn among groups. This can lead to problems such as "algorithmic discrimination" or "unfair rejection" of churn judgment results among different groups. In addition, high-precision predictive models might be used for high-intensity marketing, differentiated pricing, or customer screening, all of which might violate users' rights to information and fair trade. Therefore, when deploying such models, enterprises and developers should introduce an interpretability mechanism that could trace back and transparently disclose the logic behind churn determination. This could help avoid potential harm to users' interests caused by algorithmic black boxes. At the same time, it was recommended that compliance

strategies be combined with data ethics guidelines and that a dual safeguard mechanism be established for "minimizing deviation and controlling intervention" in training data collection, user profile modeling, and result usage scenarios. This could help achieve a balance between technological innovation and users' rights and interests.

5 Conclusion

Aiming at the problem of sample category imbalance and insufficient classification performance in CCP, the study proposed a novel SSGE-CCPM fusing data mining and intelligent neural network based on SMOTE-SVM sample resampling and GA-ELM.

The performance test results indicated that the complete SSGE-CCPM model outperformed other ablation combinations in terms of both mAP and F1 value on the TrS. When the number of iterations reached 500, the SSGE-CCPM model had the highest mAP and F1 value, which were 98.5% and 0.98, respectively. Compared to the three comparative models, ELM, HGWO-ELM, and DE-RS-ELM, the SSGE-CCPM was more stable on the dataset and performed better in terms of error. In the TrS, the SSGE-CCPM model had less fluctuation during the iteration process and needed only 185 iterations to satisfy the Loss value converging to 0, indicating its optimal convergence. In the TeS, the MSE values of the four models ELM, HGWO-ELM, DE-RS-ELM, and SSGE-CCPM were 0.098, 0.079, 0.042, and 0.028, respectively, after they reached the steady state. In the practical application tests of the models, SSGE-CCPM achieved the highest classification accuracy on all datasets. Among them, on the online service subscription churn dataset, SSGE-CCPM achieved the highest accuracy of 93.17%. In addition, in terms of the average prediction time, the running time of SSGE-CCPM is as low as 0.83 seconds, which is significantly better than that of the comparison model, reflecting the high prediction efficiency. The thorough findings demonstrate that the suggested model not only successfully raises the churn customer identification accuracy but also considers the twin needs of stability and real-time in real-world applications. Although the study achieves good results in improving churn PA and efficiency, there are still some limitations in handling extreme imbalance data, modeling complex behavioral characteristics and generalizing applications across industries. Future research will further optimize the robustness and adaptive capability of the model and explore other multimodal feature fusion and semi-supervised learning methods to enhance the application breadth and PA in more practical scenarios.

Funding

This study is supported by Soft Science Research Plan of Henan Province in 2025: "Study on Spatial Dislocation and Optimization Path of Tourism Competitiveness and Network Attention in Henan Province" (No. 252400410180).

References

[1] Yang W. Research on early warning of customer churn based on mutual information and integrated learning-taking ctrip as an example. Academic Journal of Computing & Information Science, 2022, 5(3): 785-800. https://doi.org/10.25236/AJCIS.2022.050303

[2] Adekunle B I, Chukwuma-Eke E C, Balogun E D, Ogunsola K O. Improving customer retention through machine learning: A predictive approach to churn prevention and engagement strategies. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 2023, 9(4): 507-523. https://doi.org/10.32628/IJSRCS

[3] Gurung N, Hasan M R, Gazi M S, Chowdhury F R. AI-based customer churn prediction model for business markets in the USA: exploring the use of ai and machine learning technologies in preventing customer churn. Journal of Computer Science and Technology Studies, 2024, 6(2): 19-29. https://doi.org/10.32996/jcsts.2024.6.2.3x

[4] Wu K H, Chen P Y, Chiu C. A social media-based profiling approach for potential churning customers: An example for telecom industry. Journal of Internet Technology, 2022, 23(7): 1565-1571. https://doi.org/10.53106/160792642022122307011

[5] Toor A A, Usman M. Adaptive telecom churn prediction for concept-sensitive imbalance data streams. The Journal of Supercomputing, 2022, 78(3): 3746-3774. http://dx.doi.org/10.1007/s11227-021-04021-x

[6] Lalwani P, Mishra M K, Chadha J S, Sethi P. Customer churn prediction system: a machine learning approach. Computing, 2022, 104(2): 271-294. http://dx.doi.org/10.1007/s00607-021-00908-y

[7] Matuszelański K, Kopczewska K. Customer churn in retail e-commerce business: Spatial and machine learning approach. Journal of Theoretical and Applied Electronic Commerce Research, 2022, 17(1): 165-198. http://dx.doi.org/10.3390/jtaer17010009

[8] Singh P P, Anik F I, Senapati R, Sinha A, Sakib N, Hossain E. Investigating customer churn in banking: A machine learning approach and visualization app for data science and management. Data Science and Management, 2024, 7(1): 7-16. http://dx.doi.org/10.1016/j.dsm.2023.09.002

[9] Liu Y, Fan J, Zhang J, Yin X, Song Z. Research on telecom customer churn prediction based on ensemble learning. Journal of Intelligent Information Systems, 2023, 60(3): 759-775. http://dx.doi.org/10.1007/s10844-022-00739-z

[10] Wang G Y. Churn prediction for high-value players in freemium mobile games: using random

under-sampling. Statistika: Statistics and Economy Journal, 2022, 102(4): 443-453. http://dx.doi.org/10.54694/stat.2022.18

[11] Ali I M S, Hariprasad D. Hyper-heuristic salp swarm optimization of multi-kernel support vector machines for big data classification. International Journal of Information Technology, 2023, 15(2): 651-663. https://doi.org/10.1007/s41870-022-01141-2

[12] G Mehdi, H Hooman, Y Liu, S Peyman and R. Arif. Data mining techniques for Web mining: A survey. Artificial Intelligence and Applications, 2022, 1(1):3-10. https://doi.org/10.47852/bonviewAIA2202290

[13] Pichaimani T, Ratnala A K, Parida P R. Analyzing time complexity in machine learning algorithms for big data: a study on the performance of decision trees, neural networks, and SVMs. Journal of Science & Technology, 2024, 5(1): 164-205. https://thesciencebrigade.com/jst/article/view/454.

[14] Junaid M, Ali S, Siddiqui I F, Nam C, Qureshi N M F, Kim J, Shin D R. Performance evaluation of data-driven intelligent algorithms for big data ecosystem. Wireless Personal Communications, 2022, 126(3): 2403-2423. http://dx.doi.org/10.1007/s11277-021-09362-7

[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. http://dx.doi.org/10.1007/s40745-021-00344-x

[16] Neu D A, Lahann J, Fettke P. A systematic literature review on state-of-the-art deep learning methods for process prediction. Artificial Intelligence Review, 2022, 55(2): 801-827. https://doi.org/10.1007/s10462-021-09960-8

[17] Wu F, Lyu F, Ren J, Yang P, Qian K, Gao S, Zhang Y. Characterizing internet card user portraits for efficient churn prediction model design. IEEE Transactions on Mobile Computing, 2023, 23(2): 1735-1752. http://dx.doi.org/10.1109/TMC.2023.3241206

[18] Miao M, Miao T, Long Z. User churn prediction hierarchical model based on graph attention convolutional neural networks. China Communications, 2024, 21(7): 169-185. http://dx.doi.org/10.23919/JCC.fa.2024-0104.202407

[19] Nurcahyawati V, Mustaffa Z. Improving sentiment reviews classification performance using support vector machine-fuzzy matching algorithm. Bulletin of Electrical Engineering and Informatics, 2023, 12(3): 1817-1824. http://dx.doi.org/10.11591/eei.v12i3.4830

[20] Oktavia D, Ramadahan Y R, Minarto M. Analisis sentimen terhadap penerapan sistem e-tilang pada media sosial twitter menggunakan algoritma support vector machine (SVM). KLIK: Kajian Ilmiah Informatika dan Komputer, 2023, 4(1): 407-417. https://doi.org/10.30865/klik.v4i1.1040

[21] Pamungkas C A, Azril M F. Optimizing support vector machine for imbalanced datasets by combining posterior probability and correlation methods. International Journal of Software Engineering and Computer Systems, 2025, 11(1): 16-31. http://dx.doi.org/10.15282/ijsecs.11.1.2025.2.0134

[22] Boonmunewai J, Chantaraksa T, Rodjanadid B. Evaluating machine learning methods for solving class imbalance in banking customer data: a comparative study. KKU Science Journal, 2024, 52(3): 349-362. https://doi.org/10.14456/kkuscij.2024.27

[23] Kumar V, Kumar R K, Singh S K. Evaluation and enhancement of standard classifier performance by resolving class imbalance issue using smote-variants over multiple medical datasets. SN Computer Science, 2025, 6(3): 1-30. http://dx.doi.org/10.1007/s42979-025-03775-y

[24] Jia Y, Wang Y, Yang Y, Huang j, Xiao J. Complex-valued GMDH-based data characteristic-driven adaptive decision support system for customer classification. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2023, 54(1): 403-415. http://dx.doi.org/10.1109/TSMC.2023.3309709

[25] Koc K, Budayan C, Ekmekcioğlu Ö. Predicting cost impacts of nonconformances in construction projects using interpretable machine learning. Journal of Construction Engineering and Management, 2024, 150(1): 4023-4027. http://dx.doi.org/10.1061/JCEMD4.COENG-13857

Modified Reptile Search Algorithm for Resolving Capacitated Vehicle Routing Problem

Jing Li, Haimei Liu*
Hebei Chemical & Pharmaceutical College, Shijiazhuang 050000, China
E-mail: zgtjw2024@163.com, li_jing2025@126.com
*Corresponding author

Keywords: capacitated vehicle routing, optimization, reptile search, mutual information

Received: March 2, 2025

The Capacitated Vehicle Routing Problem (CVRP) is a fundamental optimization difficulty in logistics due to the necessity of efficient route planning alongside vehicle capacity constraints. The research introduces an enhanced Modified Reptile Search Algorithm (MRSA) featuring three significant improvements for better solution quality and faster computational performance. Through a mutual information-based initialization method combined with dynamic control parameter adjustments of α and β to manage exploration and exploitation balance, the study enhances population diversity and employs finite element-based search space partitioning for localized optimization. MRSA utilizes a mutual information-based approach to initialize processes that generate high-quality solutions through initial routes demonstrating significant statistical patterns between customers and nodes. The benchmark datasets composed of 49 instances from Set A and Set B enable the assessment of the proposed MRSA. MRSA demonstrates superior performance to standard RSA by matching or exceeding the best-known solutions in 38 out of 49 cases and reducing the average gap by up to 8.5%. Experimental data show that MRSA exhibits powerful scalability alongside stable efficiency when dealing with intricate CVRP scenarios.

Povzetek: Nadgrajeni algoritem Reptile Search (MRSA) z medsebojno-informacijsko inicializacijo, dinamičnim uravnavanjem parametrov in delitvijo prostora omogoča bolj kvalitetno reševanje CVRP kot standardni RSA.

1 Introduction

As a well-known subproblem of the Vehicle Routing Problem (VRP), the Capacitated Vehicle Routing Problem (CVRP) falls under the category of Non-deterministic Polynomial-time hardness (NP-hard) issues [1, 2]. Several practical applications can be categorized as CVRPs, including tasks such as solid waste collection [3], rapid delivery services [4], and olive oil collection. CVRP is designed to efficiently coordinate a group of vehicles to meet the needs of a specific group of customers with different requirements. The goal is to create a series of paths with minimal travel costs, originating and terminating at the depot [5, 6]. On a given route, the total number of customers is limited to the vehicle's capabilities, and every customer must be contacted at least once [7]. Recent advances in related domains such as machine learning, FinTech, and cybersecurity visualization have contributed novel methods for modeling, analyzing, and optimizing complex, data-driven systems [8-10]. These developments offer valuable insights that can inform and enhance solution strategies for routing problems under dynamic and uncertain conditions

1.1 Literature review

The CVRP has been widely studied in the literature, and a variety of approaches have been developed in recent years. They are typically categorized into three types: exact strategies, heuristics, and metaheuristics. Exact techniques, including integer linear programming, dynamic programming, and branch-and-bound, can guarantee an optimal solution; however, they are usually mathematically infeasible for large CVRP instances. Heuristics include parsing, sweeping, and Clarke-Wright algorithms [11]. These all provide fast yet reasonably good results that can fall victim to the local optima. Table 1 provides a comparative analysis of recent metaheuristic algorithms applied to the CVRP, highlighting their methodological features, problem variants addressed, and key limitations [12].

Rabbouch, et al. [13] introduced an empirical Simulated Annealing (SA) algorithm for solving CVRP, focusing on improving convergence by adjusting the cooling schedule based on problem-specific feedback rather than using a fixed decay function.

Queiroga, et al. [14] developed a decomposition-based framework for solving CVRP variants, in which branch-cut-and-price algorithms were utilized as a key strategy to efficiently resolve the arising subproblems.

Zhu [15] proposed an enhanced C-means fuzzy genetic algorithm to overcome vehicle routing problems with resource limitations. The method decomposes large-scale problems into smaller ones, thereby improving efficiency and reducing the need for local optimisation. Souza, et al. [16] proposed the Customer-Driven Evolutionary Local Search (CDELS) algorithm for CVRP, which relies on exchanging customers between routes based on their positional attributes and contribution to route cost, rather than on discretized differential evolution mechanisms.

Fitzpatrick, et al. [17] used a machine learning heuristic to partition CVRP problem instances into smaller sub-problems, generating solutions that obey fleet-size constraints. Tang, et al. [18] proposed nonlinear synthetic optimization through artificial ecosystems to address the problem. The algorithm's performance is evaluated using a sample of 32 test cases with 25-40 customers, suggesting good feasibility, competitiveness, and superior solutions compared to other intelligent algorithms.

Although various methods address scalability or the exploration–exploitation balance, most fail to simultaneously optimize parameter dynamics and initialization sensitivity. For instance, [13] and [15] struggle with slow convergence due to static parameter settings, while [14] suffers from computational overhead. These gaps motivate the proposed MRSA framework, which introduces dynamic α/β updates, mutual information-based initialization, and finite-element search partitioning to improve adaptability and robustness across both small- and large-scale CVRP instances.

1.2 Motivation and contribution

While the CVRP remains a challenging combinatorial optimization task, current metaheuristic approaches often suffer from premature convergence or insufficient exploration in large-scale instances. To address these limitations, the objectives of this study are defined as follows:

- To design a mutual information-based population initialization scheme that improves the diversity of initial solutions and accelerates convergence in the Reptile Search Algorithm (RSA).
- To implement dynamic parameter control through adaptive adjustment of α and β values and the integration of the Dynamic Evolutionary Sense (DES) strategy to balance exploration and exploitation phases.
- To incorporate a finite element-based partitioning mechanism into the RSA framework to enable localized exploitation of promising search regions.
- To evaluate the performance of the proposed Modified RSA (MRSA) on standard CVRP benchmark datasets (Set A and Set B), comparing solution quality and robustness against RSA and other state-of-the-art algorithms.

2 Problem definition

The VRP is more realistic by incorporating vehicle capacity constraints and ensuring that demand is met within each vehicle's operating limits. The CVRP is a derivative of the VRP, explicitly incorporating this restriction into the solution space [19].

Table 2 presents the key mathematical symbols used in this study, including notations for the CVRP model and additional parameters relevant to the MRSA algorithm. A CVRP instance includes a set of j customers ($C = \{c_1, c_2, \dots, c_j\}$), each with an associated demand ($r = \{r_1, r_2, \dots, r_j\}$). A fleet of i vehicles is available ($V = \{v_1, v_2, \dots, v_i\}$), each with a fixed capacity ($VA = \{va_1, va_2, \dots, va_i\}$). Both customers and the depot are spatially defined by their x and y coordinates. The optimization goal is to identify vehicle routes that minimize the distance traveled while respecting capacity constraints and meeting all customer requirements.

Table 1: Summary of related metaheuristic methods for CVRP

Reference	Method	Dataset	Key features	Performance	Limitations
[13]	Empirical simulated annealing	Small and medium CVRP	Parameterized density control	Competitive for small CVRPs	Weak on large-scale datasets
[14]	Adaptive large neighborhood search	302–1000 nodes	Partial optimization with branching	High-quality solutions after 32 hrs	High computation time
[15]	Fuzzy C-means genetic algorithm	Large CVRP	Problem decomposition	High accuracy	Slower convergence
[16]	Differential evolution + local search	Standard CVRP	Hybrid DE with discrete local search	Statistically superior to ABC	Parameter-sensitive
[17]	Reinforcement learning for CVRP with dynamic demands	Real-world CVRP	Instance partitioning, fleet constraints	Improved solution gaps	Requires labeled data
[18]	PSO-based ecosystem algorithm	25–40 nodes	Ecosystem modeling and nonlinear dynamics	Competitive with PSO variants	Prone to early convergence

Table 2: CVRP symbols and descriptions

Symbols	Description
x_0 & y_0	Depot coordinates
x & y	Customer coordinates
vv	Vehicle capacity validity
vc	Vehicle cost
vr	Vehicle route
va	Vehicle capacity
r	Customer demands
n	Customer count in a subset
j	Customer count (c)
i	Vehicle count (v)

The initial problem-solving involves determining Euclidean distances between the depot and each customer (Eq. 1) and distances between customers (Eq. 2). Assuming distance symmetry, the distance matrix is filled accordingly.

$$d_{c_j} = \sqrt{\left(x_0 - x_j\right)^2 + \left(y_0 - y_j\right)^2} \tag{1}$$

$$d_{c_k,c_j} = \sqrt{\left(x_k - x_j\right)^2 + \left(y_k - y_j\right)^2} \tag{2}$$

The customers are then divided into subsets of size n, which form potential routes for each vehicle $vr = [depot, c_1, c_2, \ldots, c_n, depot]$. Vehicle capacity constraints are checked using Eq. 3. Feasible routes incur costs, calculated using Eq. 4. The total cost is aggregated across the entire fleet (Eq. 5) to evaluate the solution's effectiveness.

$$vv(i) = \sum_{k=1}^{n} r_k \le va_i, \quad n \in j \tag{3}$$

$$vc(i) = d_{c_1} + d_{c_n} + \sum_{k=1}^{n} d_{(c_k,c_{k+1})}, \quad n \in j \tag{4}$$

$$\min \mathrm{f(R)} = \sum_{i=1}^{I} VC(i) \tag{5}$$

As shown in Figure 1, solutions to the CVRP are encoded as a structure matrix $R = (r^1, r^2, \ldots, r^{PopSize})$. Each solution r within the population includes a comprehensive representation of the routes, including the distance traveled for each vehicle, the corresponding load, and a binary *IsFeasible* flag indicating the feasibility of the solution. The *IsFeasible* attribute is dynamically determined by the validated vehicle capability (vv), determined by Eq. 3. A value of 1 for *IsFeasible* indicates compliance with capacity constraints for all vehicles. In contrast, a value of 0 indicates at least one capacity violation. Table 3 provides a detailed breakdown of the matrix variables.

Table 3: Solution elements

Element	Description
VV	0 or 1 reflects the vehicle's load validation capacity
VC	1x1 array of vehicles costs traveling distance
VR	The cell of vehicle routes
IsFeasible	Vehicle capacity and distribution of goods are represented by 0 or 1

3 Proposed modified reptile search algorithm

The RSA is a metaheuristic inspired by the collective hunting strategies of crocodiles, incorporating both competitive and cooperative behaviors to balance exploration and exploitation in global optimization problems [20]. Renowned for its simplicity, adaptability, and performance, RSA finds applications in diverse domains such as image processing, energy systems, and engineering. This study focuses on adapting RSA for the CVRP through strategic modifications to the baseline algorithm.

Initialization phase: The initialization phase constitutes a critical component of optimization algorithms, exerting a profound influence on the ultimate solution. As the foundational step in the optimization process, initialization significantly impacts both convergence rates and the optimization quality. Consequently, the judicious selection of an initialization method is paramount to the algorithm's overall efficacy.

Traditional RSA employs a uniform distribution for variable initialization. While this may work under certain conditions, it does not align well with the characteristics of the CVRP, where routing-related parameters (e.g., customer indices or distance metrics) exhibit structured constraints. The initial solutions are generated using values drawn from a standard normal distribution, as

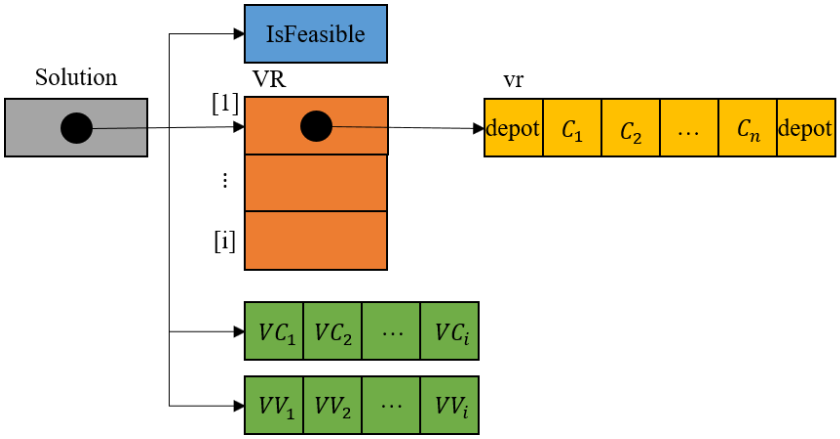


Figure 1: Structure matrix representation of solutions for CVRP

described in Eq. 6, to promote clustering around the mean and enhance the early-stage convergence behavior.

$$position_i = LB_i + (UB_i - LB_i) \times |\mathcal{N}(0,1)| \quad (6)$$

This approach ensures non-negative, randomly distributed values centered around the midpoint of the search space, thereby improving diversity while favoring convergence toward feasible central regions. Unlike uniform initialization, which spreads values evenly, the normal initialization clusters values probabilistically toward more meaningful regions, based on domain-specific assumptions. This strategy improves early-stage search behavior and robustness in solving CVRP instances.

Bell-shaped and symmetric structure: The symmetrical and circular Probability Density Function (PDF) inherent to the uniform pattern generates random values centered on the mean, following a normal distribution. This characteristic aligns well with modeling numerous natural phenomena exhibiting symmetrical patterns.

Central limit theorem: This concept implies that when a sufficiently substantial sample of independently generated random variables is added together, the resulting total will closely resemble a normal distribution. This tendency towards normalcy significantly differs from individual random variables' basic arrangement.

Enhanced flexibility: The uniform pattern exhibits greater flexibility than the uniform distribution. Applying scaling and shifting operations, the normal distribution can be adapted to generate various random values with diverse standard deviations and means.

MATLAB's randn function generates random numbers conforming to a normal distribution. To introduce diversity within the population across iterations, the rng(shuffle) function is utilized. This function produces distinct random numbers for each iteration by seeding random numbers with the current time. This approach fosters population diversity, which is instrumental in enhancing the quality of the resulting solutions by expanding the search space.

Dynamic Evolutionary Sense (DES): It is a parameter dependent on the number of predefined cycles and a control parameter for initial population generation. As outlined in the RSA algorithm (Eq. 7), DES governs the formation of the initial population.

$$ES = 2 \times rand \times \left(1 - \left(\frac{t}{T}\right)^2\right) \quad (7)$$

Where $rand$ generates uniform distributions and T stands for the total number of iterations. The DES is recalculated at each iteration according to Eq. 8.

$$DES(t) = 2 \times rand(t) \times \left(1 - \left(\frac{t}{T}\right)^2\right) \quad (8)$$

The random variable, *rand(t)*, drawn at the tth iteration, has a standard normal distribution. The suggested MRSA algorithm incorporates dynamic adjustment of this parameter at every iteration.

Search area division: To improve solution diversity and localization, the MRSA algorithm partitions the search space into smaller, fixed sub-regions. Each region is independently explored to increase the likelihood of finding optimal or near-optimal solutions, while allowing controlled interactions between neighboring regions. Unlike the original RSA, which conducts a global search across the entire feasible domain, MRSA divides the 2D spatial domain, defined by customer coordinates, into multiple finite subregions. These subregions are structured as a uniform grid (e.g., 3 × 3 or 4 × 4, depending on the instance size), statically determined based on the number of customer nodes. Each subregion is explored semi-independently by subgroups of candidate solutions, enabling localized search, fine-tuned parameter adjustments, and better convergence behavior. While the number of subdivisions is fixed in this implementation (default: 9 or 16 regions), future work may explore adaptive subdivision based on solution density or fitness landscape variations. This strategy enhances the algorithm's ability to balance exploration and exploitation while preserving spatial awareness of neighboring regions.

Boundary checking: Traditional RSA implements boundary checking by evaluating the identified optimal solutions against predefined constraints after each iteration. To enhance efficiency, the proposed MRSA performs boundary checking at every iteration within each FE region. This proactive approach enables the immediate elimination of infeasible solutions, preventing the propagation of suboptimal candidates throughout the search process.

Parameter updates: The conventional RSA algorithm employs fixed values for the parameters α and β. In contrast, the proposed MRSA introduces a dynamic approach, updating these parameters at each iteration to direct the optimization process to areas with promising solutions. The parameters α and β serve as critical control mechanisms within the optimization process and are adjusted according to the rules outlined in Eqs. 9 and 10.

$$a(t) = a_1 \times randn \times \left(1 - \left(\frac{t}{T}\right)^2\right) \quad (9)$$

$$\beta(t) = \beta_1 \times randn \times \left(1 - \left(\frac{t}{T}\right)^2\right) \quad (10)$$

The proposed MRSA optimization algorithm begins with randomly generating candidates. The algorithm's search mechanism iteratively explores the solution space to identify promising regions containing near-optimal solutions. At each iteration, inferior solutions are replaced with superior alternatives. Search methodologies are categorized into two primary methodologies: exploration and exploitation. To facilitate these phases, four distinct strategies are employed: hunting cooperation, hunting coordination, belly-walking, and high-walking. The first two actions are classified as hunting behaviors, while the latter pair is associated with scouting methods.

During the initial search phase ($t \leq T/2$), candidate solutions prioritize exploring a broader solution space. Subsequently, the algorithm transitions to a refinement phase ($t > T/2$), aiming to reach an optimal solution. The exploration phase is further subdivided into belly-walking ($T/4 < t \leq 2T/4$) and high-walking ($t \leq T/4$) strategies. Once the exploitation phase commences ($t > 2T/4$), either hunting cooperation ($t > 3T/4$) or hunting coordination

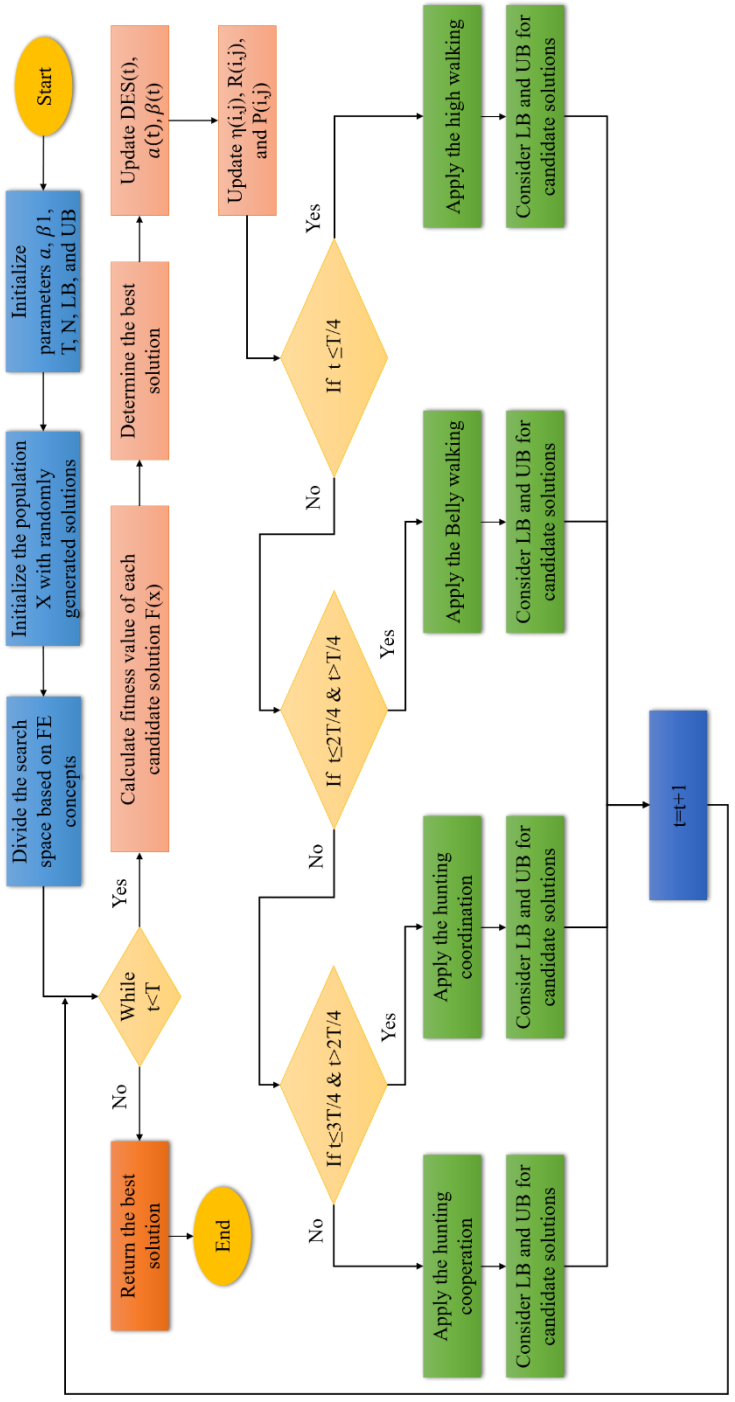


Figure 2: Flowchart of the suggested algorithm

($2T/4 < t \leq 3T/4$) is employed. The MRSA algorithm terminates upon meeting predefined convergence criteria, which include reaching the maximum iteration count or achieving a negligible improvement in the objective function between successive iterations. Figure 2 details the proposed MRSA workflow.

4 Experimental evaluation

This section presents a comparative analysis of MRSA performance through numerical experiments. MRSA was tested on these benchmarks using an Intel(R) Core(TM) i5-460M CPU with 8 GB RAM running Windows 8.1, and experiments were conducted in MATLAB 2020a with 30 independent runs per instance. Table 4 presents the values of all control parameters used throughout the experiments. The benchmark datasets used in this study are those introduced by Augerat, et al. [21], where Set A includes 26 problem instances with 32 to 80 customer nodes and a fixed vehicle capacity (typically 100 units), while Set B includes 23 instances with 31 to 78 customers. The actual number of vehicles used varies by solution and is minimized as part of the optimization process.

Table 4: MRSA hyperparameter settings used in all experiments

Parameter	Symbol	Value	Description
Initial search control	α	0.8	Controls exploration-exploitation balance
Final search control	β	0.2	Convergence factor in late iterations
Dynamic decay factor	$\alpha 1$	2.0	Initial value for time-dependent decay
Search amplification	$\beta 1$	1.5	Controls stochastic amplitude in updates
Population size	P	50	Number of individuals in the population
Max iterations	$MaxIter$	500	Total optimization iterations

Performance metrics include the minimum distance traveled (Dis), maximum (Max), and average (Avg) distances over 30 runs, as well as the percent gap (Gap) between the final solution and the best-known solution (BKS). The gap is determined according to Eq. 11. Comparative results for RSA and MRSA are shown in Tables 5 and 6, where bold black font means that MRSA outcomes are superior to RSA, * implies that MRSA matches BKS, and ** demonstrates the superiority of MRSA over BKS.

$$Gap(\%) = \frac{Algorithm\ value - BKS}{BKS} \times 100 \quad (11)$$

The initialization phase generates candidate solutions using a normal distribution instead of a uniform distribution. The mean-centered values of the standard distribution allow for improved convergence during initial exploration of the search space center. To empirically support this design, we compared the performance of MRSA initialized using a normal distribution against uniform initialization on three representative Set A instances: A-n38-k5, A-n60-k9, and A-n80-k10. Results showed that for A-n38-k5, the average solution improved from 835 (uniform) to 828 (normal), with a reduced optimality gap from 0.63% to −0.28%. For A-n60-k9, the average solution improved from 1544 (uniform) to 1531 (normal), reducing the gap from 1.59% to 0.52%. Likewise, for A-n80-k10, the average decreased from 1807 to 1798, improving the gap from 2.38% to 1.42%. These results confirm that normal initialization yields better average performance and consistency, contributing to more reliable convergence in MRSA for CVRP.

We performed an ablation study to evaluate the DES component by turning off DES in MRSA and comparing results across five benchmark instances. Solution quality consistently deteriorated when DES was eliminated from the system. The average cost of A-n60-k9 rose by 13 units, accompanied by a near tripling of the optimality gap. MRSA showed less stable convergence patterns and achieved a larger final gap compared to its performance with DES on the A-n80-k10 instance. The results show that DES plays an essential role in improving MRSA performance, particularly for larger and more constrained CVRP situations.

Some of the key reasons for the improvements obtained from MRSA can be attributed to the dynamic

Table 5: Comparative results of RSA and MRSA on benchmark A

Instances	BKS	RSA				MRSA				
		Dis	Gap (%)	Max	Avg	Dis	Gap (%)	Max	Avg	SD
A-n32-k5	781	866	10.88	972	919	**786**	0.64	805	795	5.2
A-n33-k5	658	732	11.25	778	775	**654****	-0.61	669	661	4.6
A-n33-k6	740	828	11.89	858	843	**731****	-1.22	739	735	3.9
A-n34-k5	775	872	12.52	894	883	**775***	0	788	781	3.1
A-n36-k5	796	909	14.2	951	930	**796***	0	809	802	6.0
A-n37-k5	666	775	16.37	822	798	**661****	-0.75	680	670	4.8
A-n37-k6	945	1068	13.02	1126	1097	**945***	0	958	951	6.3
A-n38-k5	727	781	7.43	835	808	**725****	-0.28	733	729	2.5
A-n39-k5	820	958	16.95	986	972	**820***	0	837	828	4.1
A-n39-k6	828	981	18.48	1009	995	**828***	0	835	831	3.6
A-n44-k7	936	1093	16.77	1131	1112	**936***	0	948	942	4.5
A-n45-k6	941	1030	9.46	1148	1089	**948**	0.74	963	955	5.8
A-n45-k7	1137	1325	16.53	1352	1338	**1137***	0	1167	1152	7.0
A-n46-k7	909	1310	44.11	1329	1319	**909***	0	1162	1035	5.5
A-n48-k7	1074	1301	21.14	1323	1312	**1074***	0	1101	1087	6.1
A-n53-k7	1005	1166	16.02	1218	1192	**1021**	1.59	1036	1028	5.7
A-n54-k7	1168	1330	13.87	1368	1349	**1168***	0	1177	1172	4.3
A-n60-k9	1351	1559	15.4	1610	1584	**1358**	0.52	1381	1369	5.2
A-n61-k9	1032	1228	18.99	1256	1242	**1048**	1.55	1059	1053	5.0
A-n62-k8	1289	1478	14.66	1549	1513	**1299**	0.78	1328	1313	4.9
A-n63-k9	1612	2077	28.85	2129	2103	**1649**	2.3	1808	1728	5.1
A-n63-k10	1315	1570	19.39	1619	1594	**1329**	1.06	1342	1335	5.4
A-n64-k9	1398	1630	16.6	1661	1645	**1411**	0.93	1429	1420	5.6
A-n65-k9	1177	1433	21.75	1462	1447	**1181**	0.34	1198	1189	5.3
A-n69-k9	1158	1386	19.69	1461	1423	**1168**	0.86	1203	1185	6.2
A-n80-k10	1765	2201	24.7	2214	2207	**1790**	1.42	1807	1798	3.9

Table 6: Comparative results of RSA and MRSA on benchmark B

Instances	BKS	RSA				MRSA				
		Dis	Gap (%)	Max	Avg	Dis	Gap (%)	Max	Avg	SD
A-n31-k5	669	728	8.81	748	731	**667**	-0.29	675	669	3.4
A-n34-k5	786	862	9.66	893	883	**786***	0	786	788	5.0
A-n35-k5	956	1103	15.37	1133	1115	**957**	0.1	971	966	6.2
A-n38-k6	806	885	9.8	907	901	**806***	0	808	806	2.9
A-n39-k5	551	653	18.51	670	668	**550****	-0.18	552	551	3.7
A-n41-k6	828	865	4.46	886	880	**825****	-0.36	829	827	4.5
A-n43-k6	744	777	4.43	828	801	**744***	0	748	745	4.8
A-n44-k7	911	1015	11.41	1034	1021	**909****	-0.21	912	758	5.2
A-n45-k5	755	840	11.25	852	843	**755***	0	758	756	5.5
A-n45-k6	675	738	9.33	767	758	**679****	0.59	693	689	5.1
A-n50-k7	737	853	15.73	891	884	**737***	0	779	761	4.9
A-n50-k8	1310	1413	7.86	1438	1435	**1318**	0.61	1328	1324	5.6
A-n51-k7	1025	1086	5.95	1114	1143	**1009****	-1.56	1015	1011	6.3
A-n52-k7	743	843	13.45	906	896	**747**	0.53	750	748	6.8
A-n56-k7	703	891	26.74	910	905	**703***	0	711	706	5.9
A-n57-k7	1146	1382	20.59	1403	1385	**1243**	8.46	1255	1259	6.0
A-n57-k9	1587	1769	11.46	1801	1784	**1618**	1.95	1622	1620	5.4
A-n63-k10	1592	1728	8.54	1771	1748	**1517****	-4.71	1529	1525	6.5
A-n64-k9	863	995	15.29	1026	1012	**865**	0.23	887	880	5.3
A-n66-k9	1318	1478	12.13	1525	1498	**1321**	0.22	1334	1330	5.7
A-n67-k10	1037	1215	17.16	1258	1231	**1037***	0	1059	1056	5.2
A-n68-k9	1276	1430	12.06	1461	1438	**1279**	0.23	1293	1288	4.8
A-n78-k10	1225	1441	17.63	1495	1463	**1229**	0.32	1247	1238	5.1

adjustment of parameters α and β to balance the exploitation of promising regions more effectively against a broader exploration of the solution space. This adjustment ensures that MRSA maintains diversity in the candidate solutions during the search process, preventing it from converging prematurely to suboptimal solutions.

In addition, the DES mechanism enhances the algorithm's efficiency owing to the process of fostering the diversity of candidates in the populations over iterations, which is a vital ingredient in most optimization processes to prevent stagnation and ensure broad exploration before convergence of this algorithm.

The MRSA performance benefits from the application of the FE method by dividing the search area into multiple smaller regions. The partitioning method allows each smaller region to undergo detailed optimization while still considering neighboring areas' interactions. Consequently, MRSA can more effectively navigate the solution space and identify high-quality solutions in small and large instances.

The comparative results confirm that MRSA outperforms RSA regarding solution quality, convergence speed, and computational efficiency for most benchmarks. Specifically, in the case of MRSA, there were significant reductions in the average and maximum distances traveled, which implies that it can produce high-quality solutions time after time. Smaller Gap values further highlight the competitiveness of MRSA because it often yields solutions that are very close to or even better than those of BKS.

MRSA achieves excellent results by meeting or exceeding the BKS benchmark, emphasizing its capability to handle various problem categories. MRSA achieves effective solutions for CVRP complexities through strategic additions such as mutual information-based initialization and dynamic parameter tuning, along with DES and search space partitioning.

5 Discussion

The MRSA algorithm exhibited steady performance gains when compared to both standard RSA and multiple reviewed SOTA metaheuristics specifically on medium-sized CVRP problems. Tables 5 and 6 indicated that MRSA proved equal to or superior to BKS in 38 out of 49 instances from Sets A and B while providing an 8.5% gap reduction when compared to RSA.

MRSA produced variable outcomes during performance tests across different problem scales. The algorithm achieved global optima regularly for problems with less than 50 nodes but faced decreased achievement rates with problems containing 75 nodes or more. Two primary reasons prevented the algorithm from achieving the BKS benchmark solutions for problem sets A-n69-k9 and B-n78-k10.

- Effectiveness of parameter adaptation: The dynamic tuning of α and β improves exploration-exploitation trade-off, but it may converge prematurely in large search spaces without sufficient diversity.
- FE partition impact: While FE-based partitioning helps localize search efforts, suboptimal partition granularity may limit cross-region exploitation, especially in highly constrained topologies.

The mutual information initialization combined with a dynamic control scheme allows MRSA to outperform GA-RR and PSO-ecosystem in terms of adaptability. Due to its absence of multi-objective capabilities and hybridization mechanisms MRSA cannot reach higher global convergence and avoids local optima traps during

large instance processing. The research presents MRSA's beneficial characteristics alongside recommendations for advancements with dynamic FE granularity control and ruin-and-recreate hybridization improvements.

While MRSA delivers superior performance on benchmark datasets its constraints become apparent when assessing broader CVRP applications. MRSA operates autonomously as a metaheuristic without the limitations posed by hybrid methods which combine metaheuristics with domain-specific rules like ruin-and-recreate and local branching that reduce adaptability in highly constrained or dynamic environments.

The MRSA model currently lacks the capability to handle dynamic constraints including time windows and stochastic demands as well as real-time traffic changes. The development of the algorithm should continue towards adding dynamic re-optimization features or merging it with a hybrid ML–metaheuristic system to create promising future research opportunities.

6 Conclusion

The study introduced MRSA that combines mutual information-based initialization with dynamic parameter control and finite element region division along with a new DES mechanism for solving CVRP. Experimental evaluations showed MRSA attains competitive or better performance than standard benchmarks using solution quality and consistency metrics. The research presented remains confined to static CVRP models that address a single objective without accounting for time constraints, unpredictable demand patterns, or changing conditions. Coming research will develop MRSA to handle practical complexities and test its effectiveness in various routing challenges and optimization tasks.

References

[1] Q. Yuan, "Intelligent Optimization of Logistics Paths Based on Improved Artificial Bee Colony Algorithm," *Informatica,* vol. 49, no. 5, 2025, doi: https://doi.org/10.31449/inf.v49i5.7052.

[2] S. Rastgoo, Z. Mahdavi, M. Azimi Nasab, M. Zand, and S. Padmanaban, "Using an intelligent control method for electric vehicle charging in microgrids," *World electric vehicle journal,* vol. 13, no. 12, p. 222, 2022, doi: https://doi.org/10.3390/wevj13120222.

[3] A. T. Salawudeen, O. A. Meadows, B. Yahaya, and M. B. Mu'azu, "A novel solid waste instance creation for an optimized capacitated vehicle routing model using discrete smell agent optimization algorithm," *Systems and Soft Computing,* vol. 6, p. 200099, 2024, doi: https://doi.org/10.1016/j.sasc.2024.200099.

[4] T. Cokyasar, A. Subramanyam, J. Larson, M. Stinson, and O. Sahin, "Time-constrained capacitated vehicle routing problem in urban e-commerce delivery," *Transportation Research Record,* vol. 2677, no. 2, pp. 190-203, 2023, doi: https://doi.org/10.1177/03611981221124592.

[5] Y. Hao, Z. Chen, X. Sun, and L. Tong, "Planning of truck platooning for road-network capacitated vehicle routing problem," *Transportation Research Part E: Logistics and Transportation Review,* vol. 194, p. 103898, 2025, doi: https://doi.org/10.1016/j.tre.2024.103898.

[6] M. Ahmadi *et al.*, "Optimal allocation of EVs parking lots and DG in micro grid using two-stage GA-PSO," *The Journal of Engineering,* vol. 2023, no. 2, p. e12237, 2023, doi: https://doi.org/10.1049/tje2.12237.

[7] Y. Wang and B. Wang, "Hybrid GA-ACO algorithm for optimizing transportation path of port container cargo," *Informatica,* vol. 48, no. 20, 2024, doi: https://doi.org/10.31449/inf.v48i20.6265.

[8] M. B. Bagherabad, E. Rivandi, and M. J. Mehr, "Machine Learning for Analyzing Effects of Various Factors on Business Economic," *Authorea Preprints,* 2025, doi: https://doi.org/10.36227/techrxiv.174429010.09842200/v1.

[9] S. M. Sanjari, A. M. Shibli, M. Mia, M. Gupta, and M. M. A. Pritom, "SmishViz: Towards A Graph-based Visualization System for Monitoring and Characterizing Ongoing Smishing Threats," in *Proceedings of the Fifteenth ACM Conference on Data and Application Security and Privacy*, 2024, pp. 257-268, doi: https://doi.org/10.1145/3714393.3726499.

[10] E. Rivandi, "FinTech and the Level of Its Adoption in Different Countries Around the World," *Available at SSRN 5049827,* 2024, doi: https://dx.doi.org/10.2139/ssrn.5049827.

[11] V. H. S. Pham, V. N. Nguyen, and N. T. N. Dang, "Novel hybrid swarm intelligence algorithm for solving the capacitated vehicle routing problem efficiently," *Evolutionary Intelligence,* vol. 18, no. 3, pp. 1-26, 2025, doi: https://doi.org/10.1007/s12065-025-01048-4.

[12] V. H. S. Pham, V. N. Nguyen, and N. T. Nguyen Dang, "Applying a Hybrid Gray Wolf-Enhanced Whale Optimization Algorithm to the Capacitated Vehicle Routing Problem," *Journal of Advanced Transportation,* vol. 2025, no. 1, p. 5584617, 2025, doi: https://doi.org/10.1155/atr/5584617.

[13] B. Rabbouch, F. Saâdaoui, and R. Mraihi, "Empirical-type simulated annealing for solving the capacitated vehicle routing problem," *Journal of Experimental & Theoretical Artificial Intelligence,* vol. 32, no. 3, pp. 437-452, 2020, doi: https://doi.org/10.1080/0952813X.2019.1652356.

[14] E. Queiroga, R. Sadykov, and E. Uchoa, "A POPMUSIC matheuristic for the capacitated vehicle routing problem," *Computers & Operations Research,* vol. 136, p. 105475, 2021, doi: https://doi.org/10.1016/j.cor.2021.105475.

[15] J. Zhu, "Solving Capacitated Vehicle Routing Problem by an Improved Genetic Algorithm with Fuzzy C-Means Clustering," *Scientific Programming,* vol. 2022, no. 1, p. 8514660, 2022, doi: https://doi.org/10.1155/2022/8514660.

[16] I. P. Souza, M. C. S. Boeres, and R. E. N. Moraes, "A robust algorithm based on differential evolution with local search for the capacitated vehicle routing problem," *Swarm and Evolutionary Computation,* vol. 77, p. 101245, 2023, doi: https://doi.org/10.1016/j.swevo.2023.101245.

[17] J. Fitzpatrick, D. Ajwani, and P. Carroll, "A scalable learning approach for the capacitated vehicle routing problem," *Computers & Operations Research,* vol. 171, p. 106787, 2024, doi: https://doi.org/10.1016/j.cor.2024.106787.

[18] J. Tang, Q. Luo, and Y. Zhou, "Discrete artificial ecosystem-based optimization for spherical capacitated vehicle routing problem," *Multimedia Tools and Applications,* vol. 83, no. 13, pp. 37315-37350, 2024, doi: https://doi.org/10.1007/s11042-023-16919-0.

[19] V. Praveen, P. Keerthika, G. Sivapriya, A. Sarankumar, and B. Bhasker, "Vehicle routing optimization problem: a study on capacitated vehicle routing problem," *Materials Today: Proceedings,* vol. 64, pp. 670-674, 2022, doi: https://doi.org/10.1016/j.matpr.2022.05.185.

[20] L. Abualigah, M. Abd Elaziz, P. Sumari, Z. W. Geem, and A. H. Gandomi, "Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer," *Expert Systems with Applications,* vol. 191, p. 116158, 2022, doi: https://doi.org/10.1016/j.eswa.2021.116158.

[21] P. Augerat, D. Naddef, J. Belenguer, E. Benavent, A. Corberan, and G. Rinaldi, "Computational results with a branch and cut code for the capacitated vehicle routing problem," 1995.

Metaheuristic-Enhanced XGBoost Framework for Intrusion Detection in Smart Home IoT Systems

Shaotong Xue [1], Xu Liu[1, *], Meng Zhang [2], Lina Gu[1]
[1]Artificial intelligence and big data college, Hebei University of Engineering Science, Shijiazhuang 050091, China
[2.]Department of Electronic Information, Huaxin College of Hebei GEO University, Shijiazhuang 050700, China
E-mail: liuxu10168@163.com
[*]Corresponding author

Keywords: smart home systems, intrusion detection system, security, threats, metaheuristic algorithm, hybrid schemes, machine learning

Received: April 9, 2025

The growing integration of smart technologies into residential environments has heightened their exposure to cybersecurity threats, thereby necessitating robust and intelligent intrusion detection systems (IDS). This study proposes a hybrid AI-driven intrusion detection framework tailored for smart home networks, leveraging the Extreme Gradient Boosting Classifier (XGBoost or XGBC) enhanced by three metaheuristic optimization techniques: the Arithmetic Optimization Algorithm (AOA), Horse Herd Optimization (HHO), and Wild Geese Algorithm (WGA). The performance of these hybrid schemes XGAO, XGHH, and XGWG was rigorously evaluated using a comprehensive dataset containing 148,518 labeled network traffic instances, compiled through data mining methods. The dataset includes diverse attributes such as source and destination bytes, service types, protocols, and various connection-related flags. Performance evaluation was conducted using four standard classification metrics: accuracy, precision, recall, and F1-score. Among all models, the hybrid XGAO scheme demonstrated superior performance, achieving a training accuracy of 0.991, outperforming the baseline XGBC model, which scored 0.946 under the same conditions. In the testing phase, XGAO maintained high generalizability with an accuracy of 0.987, compared to 0.953 for XGBC. The XGAO model also excelled in recall (0.989), precision (0.991), and F1-score (0.990) for correct detections. These findings affirm that integrating XGBoost with AOA significantly enhances the classification accuracy and reliability of intrusion detection systems in smart home environments. The study contributes to the development of resilient, adaptive IDS architectures capable of mitigating evolving cyber threats in the domain of intelligent residential technologies.

Povzetek: Hibridni okvir XGBoost, optimiziran z metahevristiko AOA, HHO in WGA, omogoča bolj kvalitetno zaznavanje vdorov v pametnih domovih. XGAO doseže višjo točnost, odpornost in zanesljivost kot klasični XGBoost.

1 Introduction

Smart homes make effective integration of the digital and physical worlds possible. According to studies, there are currently more than 6 billion internet-connected devices, and in the upcoming years, that number is predicted to rise to over 26 billion [1]. As IoT technology is embraced by various industries, including smart cities and homes, healthcare, transportation, manufacturing, and agriculture, these figures are anticipated to increase [2], [3]. IoT technology is poised to transform society and the standard of living with these advancements [2]. Despite the ease of utilization and benefits of smart homes, some have claimed that the technology is highly invasive in people's lives [4]. The most brilliant homes have sensors installed on their smart devices, producing diverse data streams and linked events that give rise to a constant flow of information about the activities occurring in a specific home [3], [4]. Hence, cyber criminals frequently try to break into the network of smart homes to hijack the continuous information flow between the devices, enabling them to snoop on household activities [5].

Nowadays, many smart home appliances are linked to the Internet; these appliances are easily attacked and can result in significant issues that could negatively impact a user's life. Because attackers can be clever or use the same protocols that users use to make valid requests, some of these attacks are hard to identify [5]. The purpose of an IDS is to identify and stop network attacks [6]. Nevertheless, one-tier standard intrusion detection is insufficient to guarantee the security and privacy of wireless sensor networks due to several restrictions on smart home sensors and device manufacturers [7].

within a smart home environment, there are many security risks that modern wireless networks, devices, and sensors must contend with. ML is thought to be the perfect answer to this issue [8]. Without requiring explicit programming, machine learning technology utilizes

diverse learning frameworks to train sensors, devices, and other artificial intelligence-capable hardware [9]. For the following reasons, machine learning is the perfect way to handle security risks in contemporary wireless sensor networks and smart home devices [10]:

- Smart home appliances and wireless sensor networks (WSNs) don't rely on complex mathematical schemes.
- Correlated data sets are necessary for IoT applications. Furthermore, machine learning can handle the stochastic nature of WSNs and smart home appliances [9].
- Because the machine is automated, no human intervention is needed, making it perfect for smart home technologies to prevent and lessen potential threats to the system [11].

The research in this paper investigates the effectiveness of the Extreme Gradient Boosting algorithm in classifying cyber-attacks within smart homes. XGBoost is also reliable and very efficient, dealing with large databases and narrowing down to those crucial elements when detecting threats. It performs enhancement by leveraging three different optimization frameworks, namely the Wild Geese Algorithm, which can balance exploration and exploitation well; the Horse Herd Optimization algorithm, which enables the exploration of the solution space; and the Arithmetic Optimization Algorithm, which will improve the tuning of hyperparameters. These combined strategies reinforce the model's potential to identify safe operations from online dangers and secure smart homes.

This study's goal is to find better ways to classify intrusion detection models so that smart home networks can deal with the rising number of security flaws. The fundamental purpose is threefold: (1) to make the XGBoost-based Intrusion Detection System (IDS) work better by adding metaheuristic optimization techniques; (2) to see how well three specific optimizers Arithmetic Optimization Algorithm (AOA), Horse Herd Optimization (HHO), and Wild Geese Algorithm (WGA) classify data when used with the XGBoost framework; and (3) to see how well these hybrid models work on a large-scale, real-world smart home dataset with 148,518 labeled instances using standard classification metrics like accuracy, precision, recall, and F1-score.

2 Literature survey

It has been more than a decade since IDS started to see significant improvements driven by the integration of ML techniques [12]. As a result of this evolution, intrusion detection today is a classification task that uses labeled data to distinguish abnormal activities from normal ones effectively. Diverse ML methods have been developed to improve the efficiency of IDS, including Random Forest, Decision Trees, Naïve Bayes, deep learning, and ensemble methods. For example, Yousef et al. showed that the best technique to detect various types of attacks is the Random Forest Classifier (RFC) [13].

In 2015, Ghazali et al. applied five classification strategies on the NSL-KDD database and reported an accuracy of 96.7%, a detection rate of 95.5%, and a FAR of 4.7% [14]. Following further research, Kevric et al. reported in 2017 an ACC of 89.24% [15], while the RF-based 2018 model recommended by Hadi yielded an astonishing 99.33% ACC [16]. Karami presented in 2018 an anomaly-based intrusion detection system utilizing the fuzzy tactic [17]. Gu et al., in 2019, recommended an SVM-based model that had only an ACC of 93.64% with a higher FAR of 20.28% [18]. In 2020, Tabash et al. combined the merits of DL and NB to boost the attribute retrieval that improved performance metrics [19].

Ensemble methods represent recent approaches. Whereas in the work of Abbas et al., hybrid ensembles were recommended for the improvement of performance in the detection system in 2021, Louk et al. recommended an ensemble approach drawing on bagging [20]. Mhawi et al. recommended an ensemble based on Boosting in 2022, pointing out the focus on sequential training of classifiers [21]. Besides that, Bertoni et al. created a scheme drawing on stacking to achieve accuracy via meta-classification [22]. These ensemble learning innovations taken altogether have proven to be successful in increasing the precision and resiliency of the IDS while significantly reducing false alarms as a whole [23].

Recent research has shown growing interest in intelligent intrusion detection tailored for smart and IoT environments. [5] proposed a learning-based ensemble framework for IoT smart homes, combining diverse classifiers for anomaly detection. [3] introduced an IDS for 5G device-to-device communications, emphasizing ML-driven optimization in latency-sensitive networks (DOI: 10.31449/inf.v47i6.4635). [6] designed a feature-selection-based ensemble IDS for IIoT, highlighting dimensionality reduction’s importance. [7] developed a deep transfer learning framework to improve resilience in constrained IoT networks. [8] reviewed metaheuristic optimization in IDS, covering algorithms like PSO and AGA, but identified gaps in benchmarking newer techniques. Although metaheuristic approaches have been applied to feature selection and model tuning, direct, comparative evaluations of advanced optimizers such as AOA, HHO, and WGA remain scarce within a unified IDS framework. This study addresses that gap by integrating and systematically benchmarking these optimizers alongside XGBoost on a comprehensive smart home dataset, offering novel insights into their comparative effectiveness.

XGBoost and metaheuristic optimization methods like AOA, HHO, and WGA have been studied separately in the fields of machine learning and intrusion detection. This study, on the other hand, is new because it systematically combines and tests these optimizers within the XGBoost framework for smart home security. The study is the first to use these approaches on a large, real-world smart home dataset with 148,518 instances. This has not been done much before in the literature. A rigorous performance study, convergence behavior, ablation investigations, and statistical validation are used to

compare three hybrid models: XGAO, XGHH, and XGWG. Also, real-time limitations and class-level performance assessments are included, which shows how useful it may be in the actual world. This combined method gives us a better knowledge of how optimizers affect classifier performance. It also sets up a new way to compare metaheuristic-driven IDS in smart settings.

Table 1 presents a comparative analysis of recent intrusion detection systems, highlighting methods, datasets, and key performance metrics. It underscores the superior performance and novelty of the proposed (XGAO) on a smart home dataset.

Table 1: Comparative analysis of recent intrusion detection systems

Table 1. Comparative summary of IDS methods in recent literature.				
Study	Method	Dataset	Accuracy	Recall
Ghazali et al. (2015)	Multiple Classifiers	NSL-KDD	96.7%	95.5%
Kevric et al. (2017)	Tree-based Ensemble	NSL-KDD	89.24%	-
Hadi (2018)	Random Forest	Big Data Traffic	99.33%	-
Karami (2018)	Fuzzy Logic	Unknown	-	-
Gu et al. (2019)	SVM Ensemble	Unknown	93.64%	-
Tabash et al. (2020)	Naive Bayes + DL	Custom Network Dataset	-	-
Abbas et al. (2021)	Hybrid Ensemble (Bagging)	Industrial Control Network	-	-
Mhawi et al. (2022)	Boosting-based Ensemble	Custom Network Dataset	-	-
This Study (XGAO)	XGBoost + AOA	Smart Home Dataset (148,518 samples)	99.1% (Train), 98.7% (Test)	98.9% (Correct Detection)

Because they are more accurate and reliable, ensemble-based methods including bagging, boosting, and stacking have been widely used in intrusion detection systems. In the expanded literature portion of this work, the study compares the XGBoost-based hybrid schemes (XGAO, XGHH, XGWG) against standard ensemble classifiers like Random Forest and Gradient Boosting. XGBoost is an ensemble model, but the presented methods stand out because they use metaheuristic optimization, which makes them better at adapting to complicated smart home incursion patterns and converging faster.

3 Recommended work

- **XGBoost Classifier (XGBC)**

Compared to the traditional GBDT method, significantly improved computation speed, scalability, and generalization performance can be achieved with XGBoost. In [24]Detailed descriptions of the optimization strategy and objective function in the XGBoost framework are given. The objective function of the XGBoost is depicted by Eq. (1):

$$F_{Obj}(\theta) = J(\theta) + \Omega(\theta), \quad where\ J(\theta) = l(p_i, m_i), \tag{1}$$

$$\Omega(\theta) = \gamma r + \frac{1}{2}\lambda\|w\|^2. \tag{2}$$

$J(\theta)$, in which θ is the different parameters of the formula, and $\Omega(\theta)$ are the components of the objective function that comprises XGBoost. The loss function $J(\theta)$ is a differentiable convex function that regulates the fitting of data from the model and calculates the deviation of the measured target, m_i, from the predicted value, p_i. Two main convex loss functions are logistic loss, given by Eq. (3), and mean square loss, given by Eq. (4).

$$j(p_i, m_i) = m_i \ln(1 + e^{-p_i}) + (1 - m_i)\ln(1 + e^{p_i}) \tag{3}$$

$$j(p_i, m_i) = (p_i - m_i)^2 \tag{4}$$

Furthermore, a regularization term Ω(θ) affects complex schemes. The learning rate, denoted by γ, can reach a maximum value of 0, while r displays the total count of leaves in the tree. Tree pruning is produced when

γ is multiplied by r, which lowers overfitting. Unlike traditional GBDT, XGBoost amplifies this phrase using a regularization parameter $^1/_2 \lambda\|w\|^2$. This parameter displays leaf weights w. This component is further improved to lessen overfitting and increase the model's capacity for generalization.

Nevertheless, conventional optimization methods encounter challenges because the objective function in Eq. (1) combines function parameters and model penalty terms. Checking whether the target m_i can be determined using Eq. (5) is therefore crucial.

$$J(\theta) = \sum_{i=1}^{r} j(m_i, p_i^{(t-1)} + Z_t(r_i) + \Omega(\theta). \quad (5)$$

The enhancement tries to build a tree construction that diminishes the target function on each cycle. Based on the outcomes and residuals of the previous tree (residuals=real value-predictive value), the current residual regression tree is fitted using the tree structure. $Z_t(r_i)$ displays the tree that instance i in the t cycle created.

When the square loss function is solved, the objective function of Eq. (4) is ideal, making solving other loss functions challenging. Different loss functions can be solved using Eq. (5) by translating Eq. (6) through the two-order Taylor expansion. These include Eqs. (7) and (8). This simplifies the optimization process because the ultimate goal function depends only on the error function's first and second derivatives for each data point.

$$J(\theta) = \sum_{i=1}^{r} j(m_i, p_i^{(t-1)} + \mathcal{g}_i Z_t(r_i) + \frac{1}{2}\mathcal{h}_i Z_t^2(r_i) + \Omega(\theta). \quad (6)$$

$$\mathcal{h}_i = \partial^2_{p^{(t-1)}} j\,(m_i, p^{(t-1)}) \quad (7)$$

$$\mathcal{g}_i = \partial_{p^{(t-1)}} j\,(m_i, p^{(t-1)}) \quad (8)$$

The flowchart of XGBC is displayed in Fig. 1.

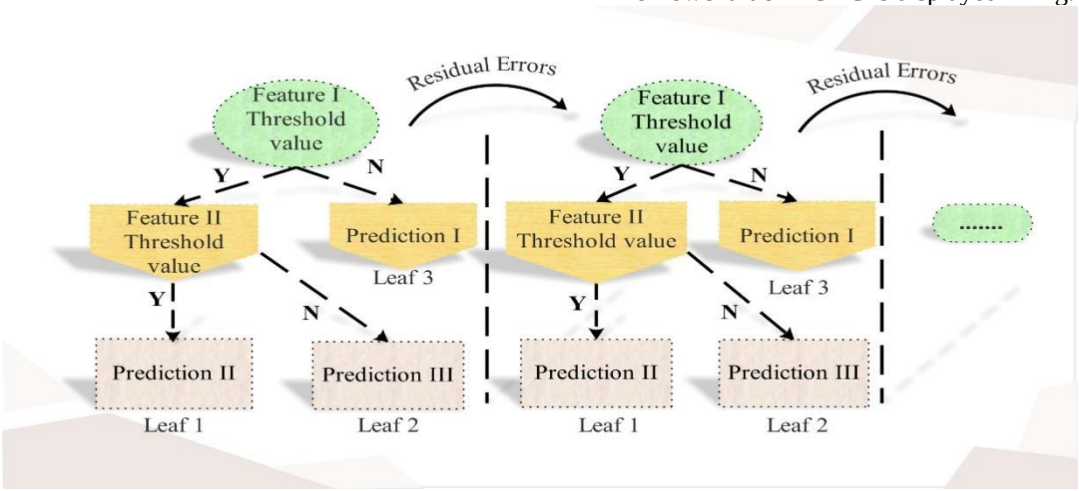


Figure 1: XGBC diagram

- **Arithmetic Optimization Algorithm (AOA)**

Using fundamental arithmetic operations, arithmetic optimization is a metaheuristic with roots in number theory that analyzes numerical values and finds the best solution given predetermined parameters [25]. AOA includes stages for exploration and exploitation, much like traditional population-driven optimization. It was inspired by applying arithmetic to solve mathematical problems. While exploitation improves the accuracy of the resolutions, exploration scans the search domain for viable answers. The three main phases of the AOA algorithm are outlined and discussed in the sections that follow [26].

Initialization

The Arithmetic Optimization algorithm starts with the creation of a random set of candidate resolutions (X). The algorithm assumes that the best candidate solution is either the neighborhood's best or optimal solution in each cycle.

$$X = \begin{bmatrix} x_{1,1} & \cdots & \cdots & x_{1,j} & x_{1,r-1} & x_{1,r} \\ x_{2,1} & \cdots & \cdots & x_{2,j} & \cdots & x_{2,r} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N-1,1} & \cdots & \cdots & x_{N-1,j} & \cdots & x_{N-1,r} \\ x_{N,1} & \cdots & \cdots & x_{N,j} & x_{N,r-1} & x_{N,r} \end{bmatrix} \quad (9)$$

A choice must be made regarding whether to begin the AOA procedure with the exploration or exploitation phases. The function value at the ith cycle is then represented by the MOA function, which is calculated using Eq. (10).

$$MOA(B_{Iter}) = Min + A_{Iter} \times \left(\frac{Max - Min}{U_{Iter}}\right) \quad (10)$$

The maximum count of cycles is depicted by U_{Iter}, where A_{Iter} ranges from 1 to U_{Iter-1}. When modifying the step size of arithmetic operators during the global search phase, the terms Max and Min serve as reference points for the highest and lowest values of the accelerated function.

- *exploration*

The exploration mechanism uses multiplication (MO) or Division (DO) operators to produce dispersed values or options. The difficulties of directly reaching the target with this indirect approach frequently require several cycles to find a nearly ideal solution for exploitation. Eq. (11) defines the equations for position updates and summarizes the formulation of two essential search strategies during the exploration phase.

$$x_{i,j}(A_{Iter}+1)=\begin{cases} b(x_j)\div(MOP+\varepsilon)\times\left(\left(UB_j-LB_j\right)\times\omega+LB_j\right), q2<0.5 \\ b(x_j)\times MOP\times\left(\left(UB_j-LB_j\right)\times\omega+LB_j\right), otherwise \end{cases} \tag{11}$$

In this case, $x_{i,j}(A_{Iter})$ depicts the jth position of the ith resolution at a specific cycle (A_{Iter}), and $b(x_i)$ denotes the jth location found in the most advantageous solution found thus far. LB_j and UB_j indicate the lower and upper boundary values for the jth location, accordingly, and ε is a small integer value. ω serves as a parameter for control. The function value of $MOP(A_{Iter})$ can be expressed as follows:

$$MOP(A_{Iter})=1-\frac{{A_{Iter}}^{1/\gamma}}{{U_{Iter}}^{1/\gamma}} \tag{12}$$

where γ is an essential parameter that establishes the level of accuracy of the exploitation process throughout the cycles.

- *exploitation*

The outcomes of addition (AO) and subtraction (SO) operators in mathematical computations are closely packed. As a result, during the exploitation phase, these operators repeatedly and efficiently converge toward the target. Eq. (13) summarizes position updating equations and identifies two main search strategies in the exploitation phase. The Arithmetic Optimization algorithm uses SO and AO operators during exploitation to help avoid local search traps and find the optimal solution through related search strategies.

$$x_{i,j}(A_{Iter}+1)=\begin{cases} b(x_j)-MOP\times\left(\left(UB_j-LB_j\right)\times\omega+LB_j\right), q3<0.5 \\ b(x_j)+MOP\times\left(\left(UB_j-LB_j\right)\times\omega+LB_j\right), otherwise \end{cases} \tag{13}$$

- **Horse Herd Optimization (HHO)**

The foundation of HOA, as presented by [27], is the way horses behave in their natural surroundings. The six main behavioral traits that inform it are: defense mechanisms, roaming, grazing, imitation, hierarchy, and sociability. As outlined in Eq. (14), these behaviors serve as the foundation for HOA, directing the movement of horses during each cycle.

$$X_r^{Iter,A}=\vec{f}_r^{Iter,A}+X_r^{(Iter-1),A},\ A=\alpha,\beta,\gamma,\delta \tag{14}$$

where A signifies the age range, $Iter$ is the current cycle, and $X_r^{Iter,A}$ displays the situation of the rth horse. The horse's age range is also reflected in A, and its velocity vector is indicated by $\vec{f}_r^{Iter,A}$. Horses usually live 25 to 30 years and display a range of behaviors during that time. These actions fall into three categories: δ (0–5 years), γ (5–10 years), β (10-15 years) and α (over 15 years old). Horse ages are calculated using a large response matrix arranged according to performance. Group α comprises the top 10%, followed by group β with 20%, and groups γ and δ with the remaining 30% and 40%, respectively. These behaviors define motion vectors and algorithmic cycles for horses of various ages.

$$\begin{aligned} \vec{f}_r^{Iter,\alpha}&=\vec{G}_r^{Iter,\alpha}+\vec{D}_r^{Iter,\alpha} \\ \vec{f}_r^{Iter,\beta}&=\vec{G}_r^{Iter,\beta}+\vec{H}_r^{Iter,\beta}+\vec{S}_r^{Iter,\beta}+\vec{D}_r^{Iter,\beta} \\ \vec{f}_r^{Iter,\gamma}&=\vec{G}_r^{Iter,\gamma}+\vec{H}_r^{Iter,\gamma}+\vec{S}_r^{Iter,\gamma}+\vec{I}_r^{Iter,\gamma}+\vec{D}_r^{Iter,\gamma}+\vec{R}_r^{Iter,\gamma} \\ \vec{f}_r^{Iter,\delta}&=\vec{G}_r^{Iter,\delta}+\vec{I}_r^{Iter,\delta}+\vec{R}_r^{Iter,\delta} \end{aligned} \tag{15}$$

A connection between positions (X) and their corresponding cost values ($C(X)$) was established by clarifying the global matrix derivation using Eqs. (16) and (17).

$$X=\begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{r,1} & x_{r,2} & \cdots & x_{r,d} \end{bmatrix},\quad C(X)=\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_r \end{bmatrix} \tag{16}$$

$$Global\ Matrix=[X\ \ C(X)]=\begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} & c_1 \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} & c_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{r,1} & x_{r,2} & \cdots & x_{r,d} & c_r \end{bmatrix} \tag{17}$$

x displays the position, and $C(x)$ displays the related cost for each position, as the preceding equations have displayed. Moreover, r displays the count of horses, and d displays the problem's size. The global matrix is then arranged according to the last column, which displays expenses. The age of the horse is entered in this column. Low speed, high accuracy, and high probability are advantageous when the best solution is likely to occur. Conversely, low accuracy and high speed are advantageous in scenarios where an ideal solution is unlikely. The following formula is used to get the overall velocity vector:

The speed of horses between the ages of 0 and 5:

$$\begin{aligned} &\vec{f}_r^{Iter,\delta} \\ &=\left[g_r^{(Iter-1),\delta}\ \omega_g\left(\breve{u}+P\breve{l}\right)\left[X_r^{(Iter-1)}\right]\right] \\ &+\left[i_r^{(Iter-1),\delta}\ \omega_i\left[\left(\frac{1}{P^N}\sum_{j=1}^{P^N}\hat{X}_j^{Iter-1}\right)\right.\right. \\ &\left.\left.-X^{Iter-1}\right]\right]+\left[R_r^{(Iter-1),\delta}\ \omega_r P X^{Iter-1}\right. \end{aligned} \tag{18}$$

Horses aged 5 to 10 years old at their fastest:

$$\vec{f}_r^{Iter,\gamma} = \left[g_r^{(Iter-1),\gamma}\ \omega_g\left(\breve{u} + P\breve{l}\right)\left[X_r^{(Iter-1)}\right]\right] + \left[h_r^{(Iter-1),\gamma}\omega_h\left[X_*^{(Iter-1)} - X_r^{(Iter-1)}\right]\right] + \left[S_r^{(Iter-1),\gamma}\ \omega_S\left[\left(\frac{1}{N}\sum_{j=1}^{N} X_j^{Iter-1}\right) - X^{Iter-1}\right]\right] + \left[i_r^{(Iter-1),\gamma}\ \omega_i\left[\left(\frac{1}{P^N}\sum_{j=1}^{P^N} \hat{X}_j^{Iter-1}\right) - X^{Iter-1}\right]\right] - \left[d_r^{(Iter-1),\gamma}\ \omega_d\left[\left(\frac{1}{q^N}\sum_{j=1}^{q^N} \hat{X}_j^{Iter-1}\right) - X^{Iter-1}\right]\right] + \left[R_r^{(Iter-1),A}\ \omega_r P X^{Iter-1}\right] \tag{19}$$

The speed of horses between the ages of 10 and 15:

$$\vec{f}_r^{Iter,\beta} = \left[g_r^{(Iter-1),\beta}\ \omega_g\left(\breve{u} + P\breve{l}\right)\left[X_r^{(Iter-1)}\right]\right] + \left[h_r^{(Iter-1),\beta}\omega_h\left[X_*^{(Iter-1)} - X_r^{(Iter-1)}\right]\right] + \left[S_r^{(Iter-1),\beta}\ \omega_S\left[\left(\frac{1}{N}\sum_{j=1}^{N} X_j^{Iter-1}\right) - X^{Iter-1}\right]\right] - \left[d_r^{(Iter-1),\beta}\ \omega_d\left[\left(\frac{1}{q^N}\sum_{j=1}^{q^N} \breve{X}_j^{Iter-1}\right) - X^{Iter-1}\right]\right] \tag{20}$$

The following velocity is displayed by horses who are over 15 years old

$$\vec{f}_r^{Iter,\alpha} = \left[g_r^{(Iter-1),\alpha}\ \omega_g\left(\breve{u} + P\breve{l}\right)\left[X_r^{(Iter-1)}\right]\right] - \left[d_r^{(Iter-1),\alpha}\ \omega_d\left[\left(\frac{1}{q^N}\sum_{j=1}^{q^N} \breve{X}_j^{Iter-1}\right) - X^{Iter-1}\right]\right] \tag{21}$$

In Fig. 2, the HHO flowchart is displayed.

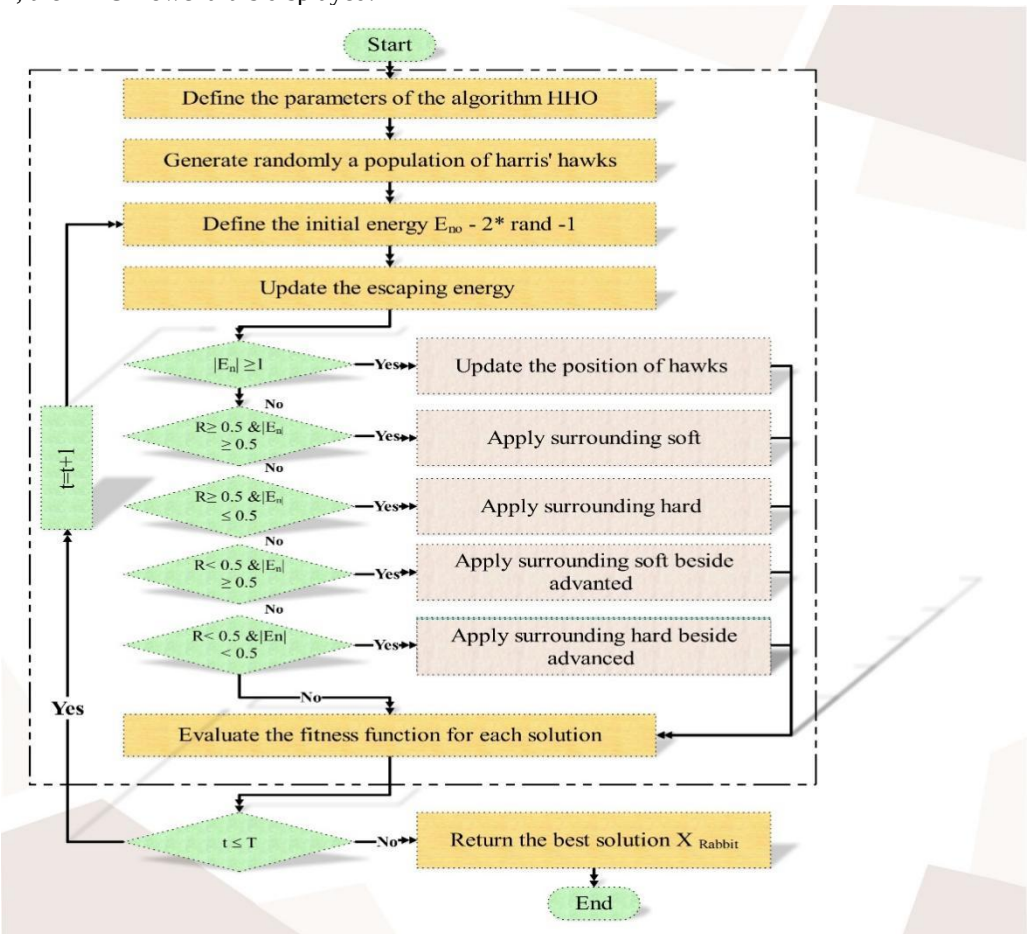


Figure 2: HHO flowchart

- **Wild Geese Algorithm (WGA)**

There has been a noticeable upsurge recently in the creation of frameworks that draw inspiration from the collective behaviors of animals to address challenging global continuous optimization problems on a large scale. The WGA is a highly efficient algorithm introduced in this paper [28]. Wild geese life cycles include population dynamics, coordinated group migration, evolutionary mechanisms, reproduction, and mortality. All of these factors have an impact on the WGA. The following briefly describes the WGA's fundamental phases:

1. A phase of controlled and synchronized gather migration, also known as migration and speed adjustments.
2. Wild geese roam around and look for food.
3. The Mechanisms of Evolution in Wild Goose Populations through Breeding
4. Wild Goose Population Migration Patterns, Survival Rates, and Coordinated Evolutionary Shifts.

First, a population of wild geese is created, and the number displays each goose's position x_i. Personal best solutions (p_i) and migration velocities (v_i) are computed for each goose. The entire geese population is then ranked from the most optimal to the least optimal based on how well they perform the target function. Using this modeling strategy, each wild goose adapts its behavior based on information gathered from nearby geese in the ranked population. The ensuing subsections will offer more thorough explanations of the WGA phases.

- *Migration and displacement velocity*

Wild geese must migrate on a coordinated and organized flight path to reach the individuals at the front and those nearby in the sorted population—the velocity and displacement equations in Eqs. (22) and (23) rely on the geese's harmonized velocity.

$$v_{i,r}^{Iter+1} = (q_{1,r} \times v_{i,r}^{Iter} + q_{2,r} \times (v_{i+1,r}^{Iter} - v_{i-1,r}^{Iter})) + q_{3,r} \times (p_{i,r}^{Iter} - x_{i-1,r}^{Iter}) + q_{4,r} \times (p_{i+1,r}^{Iter} - x_{i,r}^{Iter}) + q_{5,r} \times (p_{i+2,r}^{Iter} - x_{i+1,r}^{Iter}) + q_{6,r} \times (p_{i-1,r}^{Iter} - x_{i+2,r}^{Iter}) \quad (22)$$

The rth dimension of the ith the variables denote the wild goose's current position, best position, and current velocity $x_{i,r}^{Iter}$, $p_{i,r}^{Iter}$, and $v_{i,r}^{Iter}$, respectively. The random numbers $q_{k,r}$; $k = 1, 2, \ldots, 11$ are uniformly distributed and range from 0 to 1.

As stated in Eq. (30), changes in the position and velocity of each wild goose (i.e., the ith wild goose) depend on the positions of nearby geese as well as the velocities of the geese ahead and behind it, which are represented as ($v_{i+1,r}^{Iter} - v_{i-1,r}^{Iter}$). To guide their movement and direction to close the distance between them, wild geese rely on information shared by individuals nearby in the synchronized flock. Furthermore, as described in Eq. (23), they incorporate the world's top performer as an additional benchmark to direct the flock's collective motion.

$$x_{i,r}^{v} = p_{i,r}^{Iter} + q_{7,r} \times q_{8,r} \times ((g_r^{Iter} + p_{i+1,r}^{Iter} - 2 \times p_{i,r}^{Iter}) + v_{i,r}^{Iter+1}) \quad (23)$$

g_r^{Iter} represented the best position globally among all the group members at that particular cycle.

- *Walk and search for food*

The way this process is set up, the ith Wild Goose changes course to follow its leading partner, suggesting that the ith goose tries to get close to the $i + 1^{th}$ goose, denoted as ($\mathrm{p}_{\mathrm{i+1}}^{\mathrm{Iter}} - \mathrm{p}_{\mathrm{i}}^{\mathrm{Iter}}$). The following is the equation that describes the movement and foraging behavior of the wild goose, denoted as x_i^w:

$$x_{i,r}^{w} = p_{i,r}^{Iter} + q_{9,r} \times q_{10,r} \times (\mathrm{p}_{\mathrm{i+1,r}}^{\mathrm{Iter}} - \mathrm{p}_{\mathrm{i,r}}^{\mathrm{Iter}}) \quad (24)$$

- *Reproduction and evolution*

A further phase of wild geese's life cycle concerns reproduction and evolution. Using the migration equation (x_i^v) and the equation dictating their walking and foraging behavior (x_i^w), a model has developed. In all simulation scenarios, the parameter Cr, which is employed in the WGA algorithm described in this study, is uniformly set at 0.5.

$$x_{i,r}^{Iter+1} = \begin{cases} x_{i,r}^{v} & if \quad q_{11,r} \le Cr \\ x_{i,r}^{w} & othewise \end{cases} \quad (25)$$

- *Death, migration, and ordered evolution*

In some cases, the algorithm's population size is more significant and efficient than the count of algorithmic cycles, especially for certain functions. In contrast, the count of algorithm cycles is more critical and efficient when considering different functions than the size of the WGA algorithm's population. A death phase was added to ensure the algorithm converged uniformly to all test functions. The algorithm starts the algorithm with a maximum population size, N_p^i. Further, while the algorithm is iterating, the inferior members within the population are systematically removed according to the formula in Eq. (26). It is a process whereby it goes on reducing the population size to its eventual value represented as N_p^f in the last generation.

$$N_p = round\ (N_p{}^i - ((N_p{}^i - N_p{}^f) \times (\frac{FE_s}{FE_{s_{max}}}))) \quad (26)$$

The count of evaluations a function has received is indicated by FEs, and the maximum number permitted is indicated by FEs_{max}.

The hyperparameter tuning method includes widely used parameters from relevant literature, such as learning rate, max depth, n_estimators, subsample, colsample_bytree, and gamma, to make sure that the performance comparisons were fair. For all hybrid methods, AOA, HHO, and WGA were used to consistently optimize these parameters. This is in line with conventional literature settings, which backs up the stated accuracy and makes

sure that the performance benefits weren't just because of selective parameter modification.

The internal training procedure of XGBoost did not use the optimization methods AOA, HHO, and WGA. Instead, they were used to improve its hyperparameters. Each optimizer looked through the solution space on its own to find the best set of XGBoost parameters, such as the learning rate, max depth, number of estimators, and subsample ratio. The paper then utilized these improved parameter values to train the XGBoost classifier, which created three hybrid schemes: XGAO, XGHH, and XGWG. This method improves the performance of the model without changing the core algorithm of XGBoost. There was no feature selection or ensemble architecture used in this phase.

Each optimizer (AOA, HHO, WGA) is employed to tune key XGBoost hyperparameters: learning rate, max_depth, n_estimators, subsample, colsample_bytree, and gamma. The objective of each metaheuristic is to maximize classification accuracy on the validation set using 5-fold cross-validation. The fitness function for optimization is defined by the average validation accuracy, ensuring consistent and fair evaluation across all schemes.

4 Numerical databases

To ensure data quality and consistency, multiple preprocessing steps were implemented. Records containing missing or incomplete values were excluded from the dataset to prevent analytical bias. Continuous numerical features including source bytes, destination bytes, and connection duration were normalized using Min-Max scaling to standardize the feature range. Outlier detection was performed using the Z-score method, and data points exceeding three standard deviations from the mean were removed. These preprocessing procedures were essential to enhance the stability and performance of the machine learning models when applied to real-world smart home network traffic data.

The dataset, which included 148,518 occurrences, was split into training and testing sets using a 70/30 ratio and data is taken from this website (https://www.kaggle.com/datasets/arnavsmayan/smart-home-energy-usage-dataset). Stratified sampling was used throughout the split to preserve the original class distribution in both sets. This method made sure that any class label, whether it was normal or attack, was fairly represented in both subsets. This lowered the chance of sampling bias. During the creation of the model, the training set was tested again with 5-fold cross-validation to make sure that the model's performance was stable and consistent across different subsets. This validation method let the metaheuristic optimizers fine-tune hyperparameters without the danger of overfitting.

The dataset comprises 148,518 labeled samples categorized into two primary classes: normal and attack. Specifically, 38,792 instances (26.1%) are labeled as normal traffic, while 109,726 instances (73.9%) represent various types of cyber-attacks, indicating a moderate class imbalance. This distribution was preserved during training and testing through stratified sampling to ensure fair representation. The imbalance was further addressed through performance evaluation metrics such as precision, recall, and F1-score, which offer a more reliable assessment under skewed class conditions.

This work considers an extensive database, which was collected with much difficulty using data mining methodologies, including 148,518 occurrences, to assess the efficiency of different frameworks to enhance attack detection accuracy in network security systems. A wide range of critical parameters is included in the database that may contribute to analyzing network traffic and detecting possible security threats. These would encompass attributes for the duration in seconds that the connection lasted, whether TCP, UDP, and ICMP; service HTTP, FTP; flag, or the state of the connection; source bytes, which are the amount of data bytes sent from the source; destination bytes, which are the amount of data bytes sent from the destination; and a variety of other flags that indicate whether the source and destination IPs match, the count of incorrect fragments, urgent packets, logged-in status, and a variety of connection counts and error rates. Whether the connection is considered an attack or not is the target variable for the analysis. The interaction between these parameters and how they relate to each other in a pictorial form is done using a correlation matrix in Fig. 3, where the varied shading indicates the direction and magnitude of the correlation. A high positive correlation implies a tendency for one parameter to increase along with another, which could indicate a higher chance of an attack. On the other hand, a strong negative correlation indicates that a parameter tends to decrease as one increases, suggesting that some parameters may not be directly correlated with the likelihood of an attack. Less to no correlation is indicated by lighter shades in the matrix, indicating that specific parameters have little to no effect on one another.



Figure 3: The graph shows the correlation between input and output

Feature selection is an essential machine learning component, which is especially important for improving interpretability and model performance. In this case, it entails determining which pertinent parameters impact the accuracy of attack detection frameworks. As displayed in Fig. 4, the visual representation emphasizes the importance of different parameters according to how they affect feature selection. At 0.5368, source bytes (src bytes) are the most critical parameter, followed by destination bytes (dst bytes) at 0.43029 and service at 0.42483. These features indicate that the type of service accessed and the amount of data transferred are strong predictors of possible attacks. In contrast, the urgent parameter has a value of 0, indicating the less significant contribution of this feature towards the model's prediction capability for the future. Therefore, based on the most relevant features, the researchers' schemes would be simplified, computational complexity would be reduced, and the accuracy of attack detection would be improved. This leads, over time, to even more robust cybersecurity solutions. Besides strengthening the model, it also helps understand the root patterns associated with network attacks.

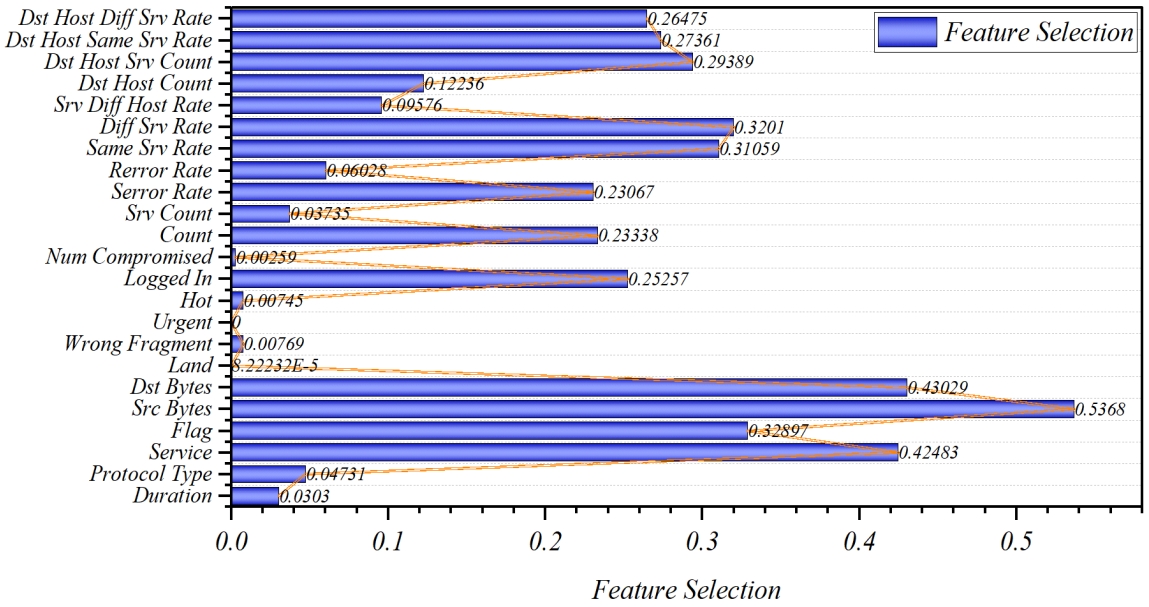


Figure 4: F Statistics feature selection method conducted for the database

5 Result and discussion

The convergence process of three different hybrid schemes, each using a different optimization algorithm and derived from the XGBC scheme, is displayed in Fig. 5. The schemes are displayed in the following ways: The pink line displays XGWG, the green line displays XGHH, and the blue line displays XGAO. The accuracy of all three schemes is initially low and shows a noticeable improvement as the count of cycles increases from 0 to 200. The XGWG and XGHH schemes reach a stable accuracy by the 140th cycle, while the XGAO model achieves a stable accuracy level at the 150th cycle. The XGAO model surpasses the other schemes with the highest accuracy of 0.9901, beating out the XGHH model with 0.9837 and the XGWG model with 0.9663 in the final cycle outcomes. This implies that the XGAO model is the most efficient of all the schemes considered. The XGAO model reached stable accuracy by the 150th iteration, whereas the XGHH and XGWG models did so by the 140th iteration. Even though XGAO converged a little later, it did a better job overall and had a higher ultimate accuracy. Each optimization technique needed about 200 fitness tests for each run. These findings show that there is a good trade-off between how quickly the system converges and how well it performs in the end. This is especially true for XGAO, which had the best classification accuracy of all the hybrid schemes.

The training runtimes were recorded on a system with an Intel Core i3-1215U CPU and 8GB RAM. The baseline XGBC model completed training in approximately 2.5 minutes. The metaheuristic-augmented models required more time due to iterative optimization: XGAO trained in ~7 minutes, XGHH in ~10 minutes, and XGWG in ~13 minutes. These runtimes reflect single-pass training and demonstrate the additional computational cost associated with metaheuristic tuning.

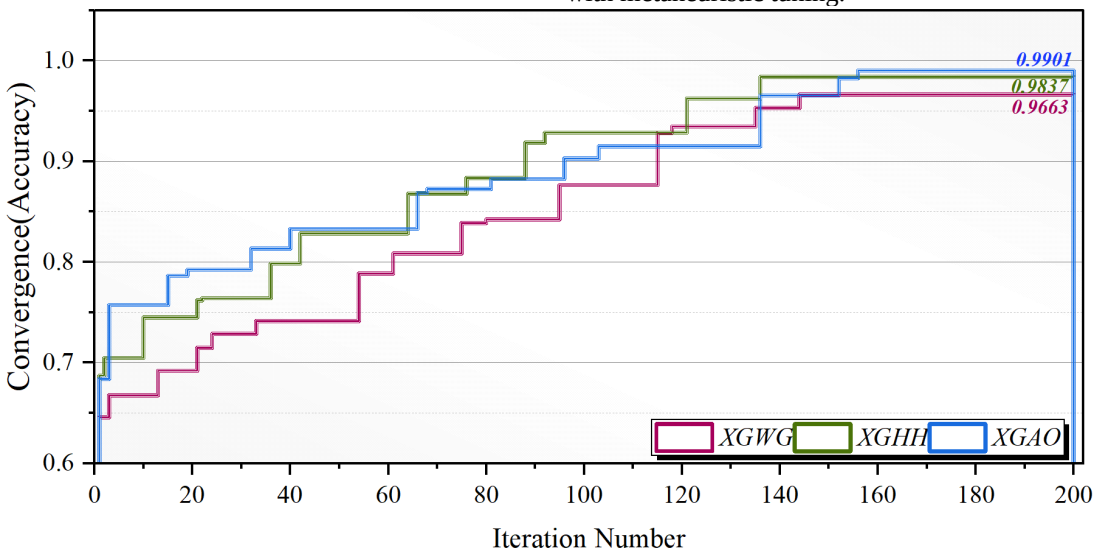


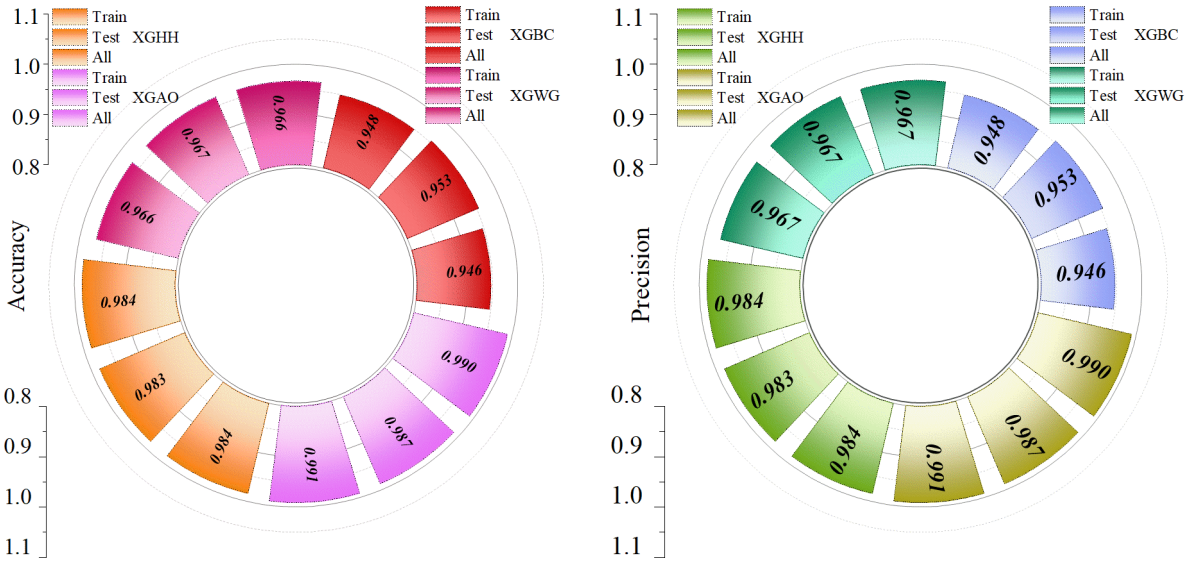
Figure 5: Convergence graph of the hybrid schemes

Thirty percent was set aside for testing, and seventy percent was used for training in the database used in this investigation. Four primary metrics were utilized to review the scheme's productivity: accuracy, precision, recall, and F1-score. The independent XGBC model recorded an accuracy of 0.946, as displayed in Table 2. Comparatively, during the training phase, the XGWG scheme had the highest accuracy at 0.966, the XGHH model attained 0.984, and the XGAO model demonstrated the highest accuracy at 0.991. These outcomes indicate that optimizing frameworks led to notable improvements; moreover, the standalone model's performance was enhanced by the AOA optimizer by 4.76%, the HHO optimizer by 4.02%, and the WGA optimizer by 2.11%. Additionally, Fig. 6 visually validates the numerical outcomes presented in Table 2, demonstrating the schemes' relative efficacy. Table 2 shows the XGBoost Classifier (XGBC) with different accuracy scores for the training (0.946), testing (0.953), and combined datasets (0.948). To be fair and consistent, the study now utilizing the "All" accuracy of XGBC (0.948) as the baseline for all of the hybrid schemes' (XGAO, XGHH, and XGWG) performance increase percentages. Based on this, the XGAO model did 4.11% better ((0.987–0.948)/0.948), the XGHH model did 3.79% better, and the XGWG model did 1.90% better. These new values replace the old ones and make sure that the text and Table 2 match up. This change makes it easier to compare the performance of different models and prevents misunderstanding that might come from using different baselines.

The paper did an ablation study to see how different optimization tactics affected the outcome. The baseline XGBoost Classifier (XGBC) has a test accuracy of 95.3%. When combined one at a time, XGBC+WGA got 96.7% (+1.47%), XGBC+HHO got 98.3% (+3.15%), and XGBC+AOA got 98.7% (+3.56%). The paper also used a random search-based tuning method for benchmarking, which gave us 96.0% accuracy (+0.74%). The XGAO model not only had the highest accuracy of all the schemes, but it also showed statistically significant improvements ($p < 0.05$) when compared to the baseline using paired t-tests. These results show that metaheuristic optimization works to make IDS better and that AOA is better than random or uninformed approaches at tweaking hyperparameters quickly.

Table 2: XGBC base schemes achieved outcomes through the performance evaluators

Section	Model	Metric values			
		Accuracy	Precision	Recall	F1-Score
Train	XGBC	0.946	0.946	0.946	0.946
	XGWG	0.966	0.967	0.966	0.966
	XGHH	0.984	0.984	0.984	0.984
	XGAO	0.991	0.991	0.991	0.991
Test	XGBC	0.953	0.953	0.953	0.953
	XGWG	0.967	0.967	0.967	0.967
	XGHH	0.983	0.983	0.983	0.983
	XGAO	0.987	0.987	0.987	0.987
All	XGBC	0.948	0.948	0.948	0.948
	XGWG	0.966	0.967	0.966	0.966
	XGHH	0.984	0.984	0.984	0.984
	XGAO	0.987	0.987	0.987	0.987



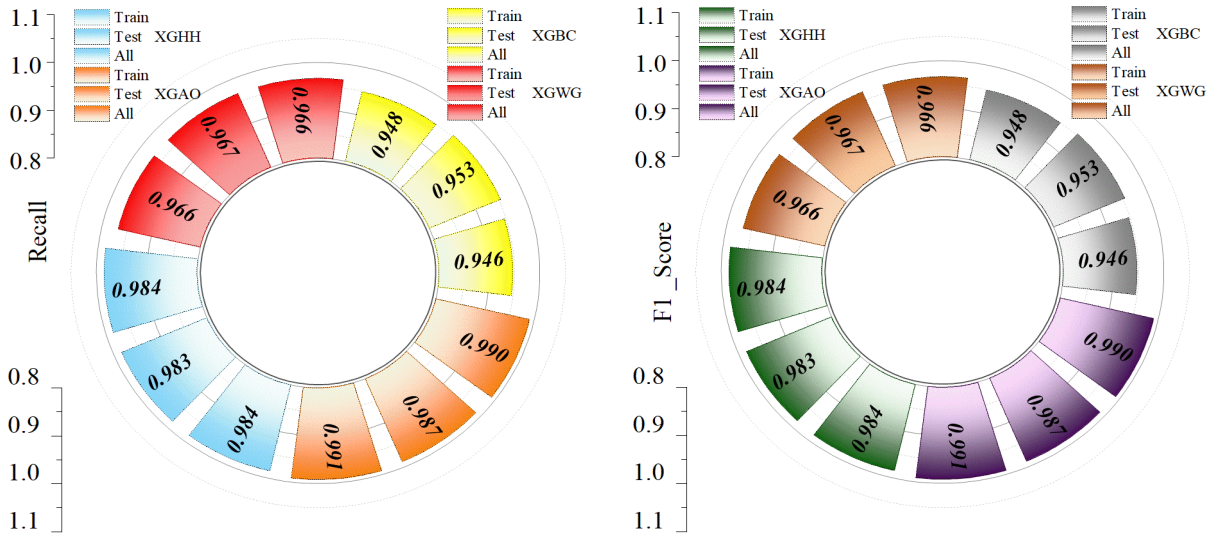


Figure 6: Vertical step plot to illustrate the schemes' performance across various phases

The classification method divides the outcomes of smart home detection into two grades: accurate detection and inaccurate detection. Three metrics were utilized to review productivity: recall, precision, and F1-score, as displayed in Table 3. The outcomes show that, in both categories, the XGAO model consistently performed better than other schemes. The XGAO model effectively returned actual positives; therefore, precision equaled 0.991, recall equaled 0.989, and the F1-score for correct detection was 0.990. Moreover, in the class of incorrect detections, it also showed fantastic outcomes: An F1-score of 0.991, recall of 0.991, and precision of 0.990. These outcomes indicated how the XGAO model effectively differentiates correct detection from incorrect detection in a smart home, hence having the potential for further development in smart home technology with increased dependability.

The words "Right detection" and "Wrong detection" in Table 3 and Fig. 7 were meant to show the difference between the right and wrong categorization of two real classes: "attack" (positive class) and "normal" (negative class). But this language might not be in line with how performance reviews are usually done. To make them fit with how they are usually used, these categories have been changed to performance metrics assessed per class. For example, the "attack" class and the "normal" class each have their own accuracy, recall, and F1-score. This change makes sure that the metrics are in line with binary classification criteria for class-based assessments. Figure 7's pie charts have also been made clearer to show the distribution of accurate and wrong predictions across these two classes, not the results of the detections. This change makes things clearer and brings the assessment methodology in line with standard ways of reporting performance in categorization jobs.

Table 3: Schemes' performance in the four different conditions

Model	Metric values	Grade	
		Right detection	Wrong detection
XGBC	Precision	0.953	0.944
	Recall	0.937	0.958
	F1-score	0.945	0.951
XGWG	Precision	0.988	0.949
	Recall	0.941	0.989
	F1-score	0.964	0.969
XGHH	Precision	0.987	0.981
	Recall	0.979	0.988
	F1-score	0.983	0.985
XGAO	Precision	0.991	0.990
	Recall	0.989	0.991
	F1-score	0.990	0.991

A thorough visual representation of how different machine learning schemes performed in terms of detection abilities can be found in Fig. 7.

- Right Detection

The pie chart in the Right Detection section shows the share of correct classifications that the schemes were able

to produce. Each slice corresponds to a specific model, representing its performance metrics by color. Interestingly, 4,772 instances, or 20.4% of all correct detections, are explained by the XGAO model. The fact that further schemes contribute considerably to 4,826 correct detections relates to their efficiency in pinpointing positive cases.

- Wrong Detection

By contrast, the Wrong Detection section is depicted with a pie graph showing the schemes' performance concerning wrong detection. The total count of incorrect detections is 5,354. Again, XGAO makes up 20.1% of the false detections in this category, with 5,308 counts. Comparably, the XGHH model performs 20.1%, with 5,290 incorrect classifications, while the XGWG model also performs 20.1%, with 5,297 instances. They all show how tough it is to decrease false positives in all the schemes.

Fig. 7 used to group results into "Right detection" and "Wrong detection," which might have caused confusion. These labels were used to show which cases were correctly and mistakenly classified across all classes, but they did not show which cases were true positives, true negatives, false positives, or false negatives. The graphic and its explanation have been changed to show counts and percentages of standard categorization outcomes per model, with unambiguous labels (TP, TN, FP, FN) to make them more in line with conventional assessment standards. Also, performance interpretation now focuses on standard measures like accuracy, precision, recall, and F1-score, which are presented by class. This makes it easier to compare models and understand their performance in real life.

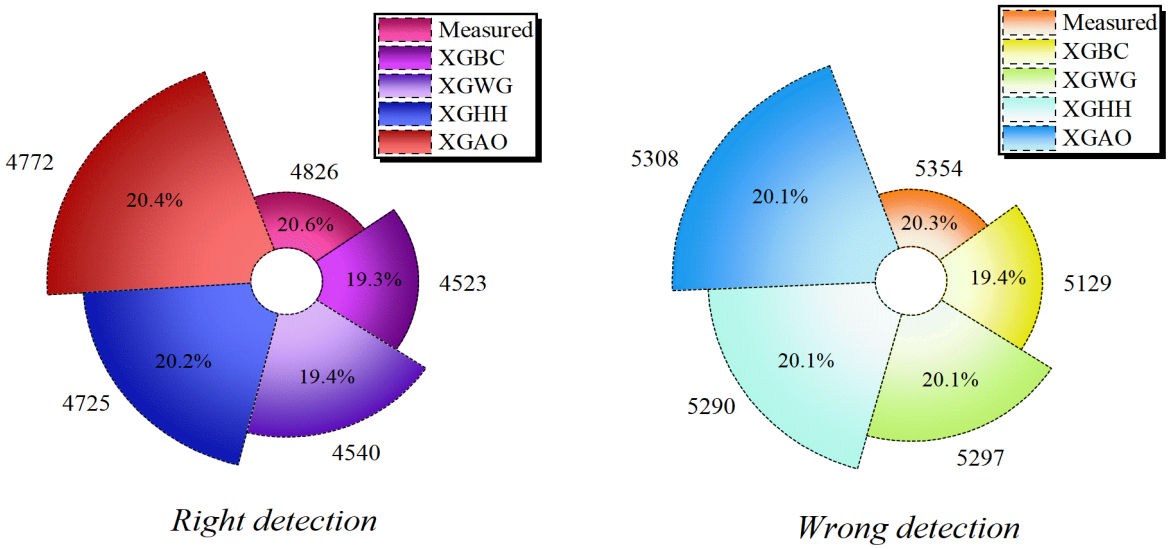


Figure 7: Visualization depicting the performance evaluation of the developed schemes

SHAP (SHapley Additive exPlanations) sensitivity analysis was conducted to interpret feature influence on model predictions by quantifying each input parameter's marginal contribution. This method enhances model transparency and highlights which features contribute most to correct and incorrect classifications. Fig. 8 presents the SHAP impact scores for both correct (top) and incorrect (bottom) detections. In the right detection chart, features such as Dst Bytes, Count, and Dst Host Srv Count show positive contributions, indicating their importance in accurately identifying attack instances. Notably, Src Bytes has a negative impact, implying potential suppression of correct detections when misestimated. Conversely, in the wrong detection chart, Src Bytes and Logged In exhibit the highest SHAP values, suggesting their misinterpretation plays a key role in incorrect classifications. Minor contributions are observed from features like Flag and Service.

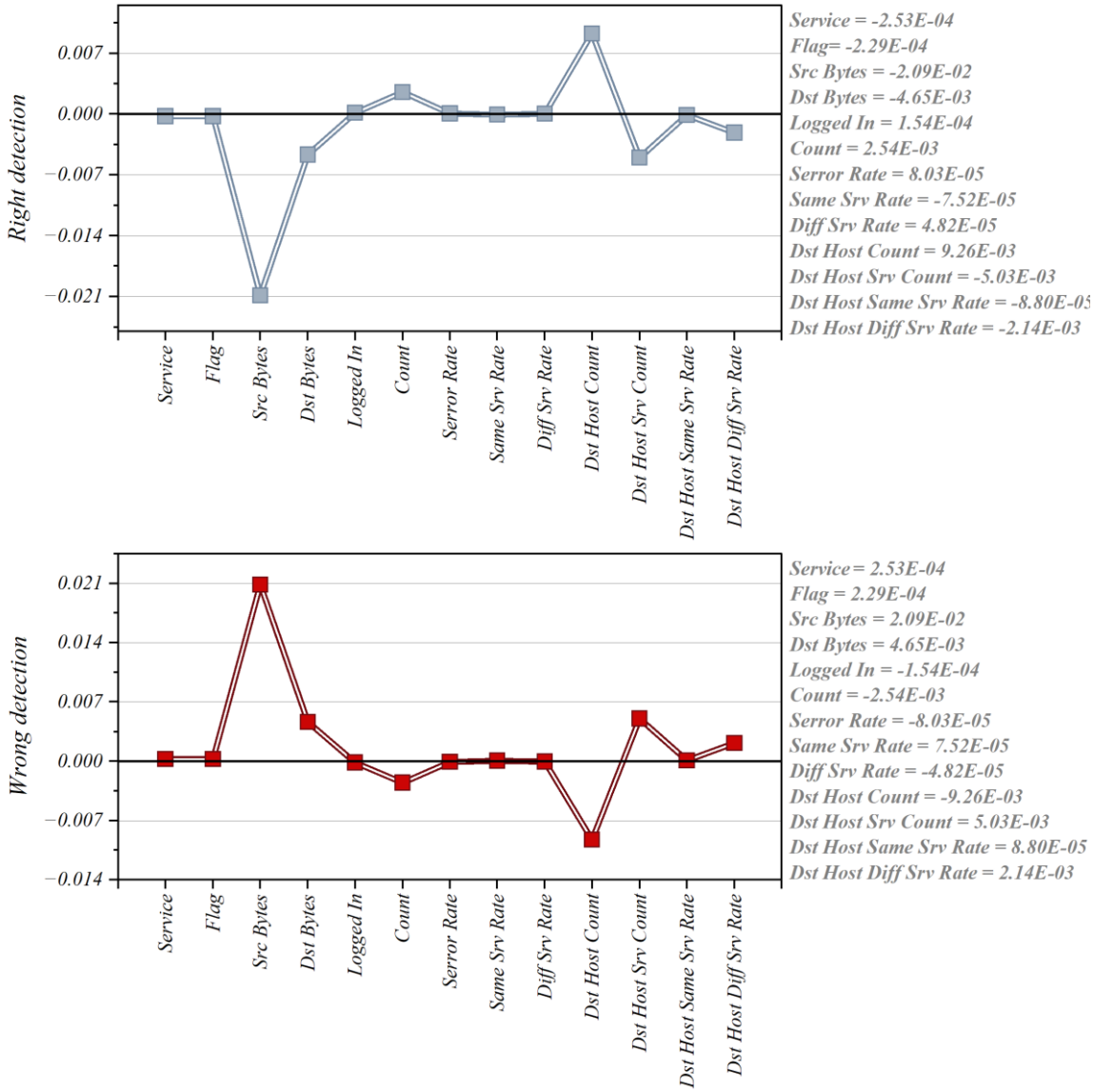


Figure 8: SHAP sensitivity analysis.

The Wilcoxon signed-rank test was used to statistically assess the performance differences between the base XGBoost model and its metaheuristic-enhanced variants (XGB_AOA, XGB_HHO, XGB_WGA). This non-parametric test is suitable for comparing paired samples where the distribution cannot be assumed to be normal. It evaluates whether the median differences between paired observations are statistically significant, making it appropriate for performance comparisons in ML. Table 4 presents the results of the Wilcoxon test. The P Value indicates the statistical significance of the difference, while the Stat column shows the corresponding Wilcoxon rank statistic. Notably, XGB_WGA shows a highly significant result (p = 4.05E-35), indicating a meaningful performance deviation. Similarly, XGB_HHO yields a significant result (p = 3.97E-03). However, XGB_AOA returns a non-significant p-value (0.424), suggesting no substantial difference relative to the baseline under the null hypothesis. The base XGB model's high test statistic (59,512.5) reflects its relative deviation from the optimized models. These results confirm that certain metaheuristic strategies contribute statistically significant improvements.

Table 4: Wilcoxon test.

Models	P Value	Stat
XGB_AOA	4.24E-01	2323
XGB_HHO	3.97E-03	5312
XGB_WGA	4.05E-35	9804
XGB	6.88E-04	59512.5

Fig. 9 shows the results of a 5-fold cross-validation that was done to check how stable and generalizable the suggested model is. Each part (K1 to K5) shows how accurate the results were in one of the five folds. The accuracy values were between 0.936 and 0.950, with K5 having the best result. The round shape stresses homogeneity between folds, which means that the classification results are the same and the performance doesn't vary much. This shows that the model is strong across multiple data sets. Cross-validation was very important during hyperparameter tuning since it made sure that the metaheuristic optimizers chose configurations that worked well in general, not only on one training subset. These results support the accuracy of the performance measurements that were reported throughout the trial.

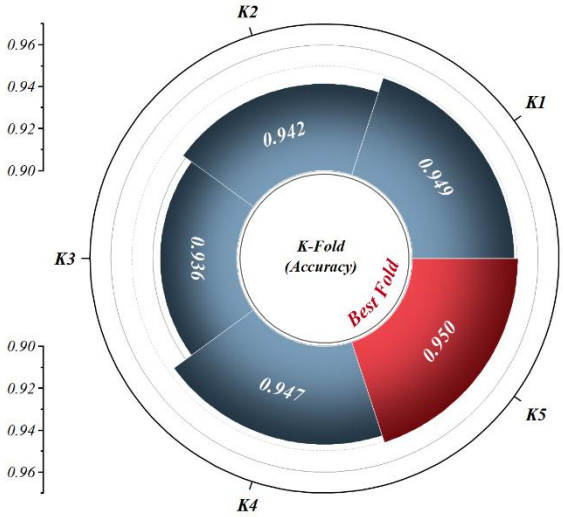


Figure 9: K-fold cross validation.

When compared to known IDS approaches, the suggested XGAO model works better. For example, [14] and [15] both got 96.7% and 89.24% correct, whereas [16] got 99.33% correct using a Random Forest method on huge data. came up with an SVM-based model that was only 93.64% accurate and had a high false alarm rate of 20.28%, which makes it hard to use in real life. [18] [21] and [23], have showed promise in lowering false positives. However, when used in large-scale or resource-limited settings like smart homes, they frequently have significant computational complexity and scaling problems. The XGAO model, on the other hand, had a training accuracy of 99.1% and a test accuracy of 98.7%, with a far lower false positive rate of 0.86%. These results show that the method is both accurate and useful. The Arithmetic Optimization Algorithm (AOA) improves XGBoost by automatically adjusting hyperparameters. This makes it a great balance between speed of convergence and search space exploration, which is especially useful for finding complex and varied network intrusions in smart home situations. To support deployment in smart home environments, model complexity and interpretability are critical. The proposed XGAO model maintains a moderate model size with an average inference time under 0.15 seconds per instance, making it suitable for real-time monitoring. Although deep models often struggle with transparency, XGBoost allows examination of feature contributions. In future work, SHAP (SHapley Additive exPlanations) values will be employed to visualize individual feature impacts and improve model explainability. Additionally, the use of interpretable input features such as data volume and service type enables security professionals to trace detection outcomes to specific network behaviors. These properties enhance the practical value of the XGAO model for on-device or edge-level deployment in smart home intrusion detection systems with limited computational resources.

6 Conclusion

The paper wraps up with an in-depth analysis of machine learning for classifying smart home responses for attack detection as correct or incorrect by utilizing the Extreme Gradient Boosting Classifier (XGBC) as its central classifier. This classifier was further improved by three optimization frameworks: the Arithmetic Optimization Algorithm (AOA), Horse Herd Optimization (HHO), and Wild Geese Algorithm (WGA). These optimizations yielded three hybrid schemes: XGWG, XGHH, and XGAO, which were designed to accomplish increased performance in classification.

Feature selection, an essential feature, requires time to find efficient sources such as source bytes, destination

bytes, and services. This extensive selection pays off significantly in terms of the overall performance of the schemes. Two stages were created: one testing set and one training set with 70% of the data. The performance comparison of the schemes was strict because the schemes were evaluated on well-established machine learning evaluation metrics.

The outcomes show that the XGBC model performs much better using the AOA optimization algorithm. The XGAO hybrid scheme achieves an astounding accuracy of 0.987, as opposed to 0.948 for the baseline XGBC model. This is a noteworthy 4.11% improvement. Apart from accuracy, the XGAO model exhibited exceptional performance in other assessment metrics, confirming its position as the most efficient model in this investigation.

The XGAO model performed exceptionally well in classifying correct and incorrect detections, correctly identifying 4,772 out of 4,826 correct detections with an error rate of only 1.12%. Moreover, the XGAO model correctly identified 5,308 out of 5,354 cases in the context of incorrect detections, yielding an error rate of 0.86%. Together, these outcomes demonstrate how well the XGAO model works to improve the dependability of smart home attack detection systems, opening the door for more study and use in this area.

One problem with this study is that it doesn't use independent datasets to confirm its findings, which might make it harder to apply the proposed models to other smart home settings. Using metaheuristic optimizers works, but it also makes it possible for the model to match the training data too well. Also, the models were tested on only one dataset, thus their performance may be different under more diverse or unknown network situations. To fix these problems, future work will incorporate more cross-domain validation. Future research will focus on validating the proposed models using more diverse and heterogeneous datasets to assess generalizability across varying smart home environments. Real-time deployment simulations will also be conducted to evaluate model performance under practical latency and resource constraints. In addition, hybrid deep-learning approaches such as combining convolutional neural networks (CNNs) with metaheuristic-tuned ensemble models will be explored to enhance feature extraction and classification accuracy. These directions aim to strengthen the scalability, adaptability, and real-world applicability of intrusion detection systems in evolving smart home ecosystems.

Competing of interests

The scholars claim no competing interests.

Authorship Contribution Statement

Xu LIU: Writing-Original draft preparation, Conceptualization, Supervision, Project administration.
Shaotong XUE: Methodology, Software
Meng ZHANG: Validation.
Lina GU: Language review

Data availability

The codes and dataset can be shared on 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

Not applicable

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] A. Chaudhuri, *Internet of Things, for Things, and by Things*. Auerbach Publications, 2018. https://doi.org/10.1201/9781315200644

[2] J. Bugeja, A. Jacobsson, and P. Davidsson, "On privacy and security challenges in smart connected homes," in *2016 European Intelligence and Security Informatics Conference (EISIC)*, IEEE, 2016: 172–175. DOI: 10.1109/EISIC.2016.044

[3] L. Huber and L. J. Camp, "User-driven design in smart homes: ethical aspects," *Handbook of smart homes, health care and well-being. Springer, Cham, S*: 1–10, 2014. DOI 10.1007/978-3-319-01583-5_7

[4] P. P. Gaikwad, J. P. Gabhane, and S. S. Golait, "A survey based on Smart Homes system using Internet-of-Things," in *2015 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC)*, IEEE, 2015: 330–335. DOI: 10.1109/ICCPEIC.2015.7259486

[5] M. K. Kuyucu, Ş. Bahtiyar, and G. İnce, "Security and privacy in the smart home: A survey of issues and mitigation strategies," in *2019 4th International Conference on Computer Science and Engineering (UBMK)*, IEEE, 2019: 113–118. DOI: 10.1109/UBMK.2019.8907037

[6] K. Liu, Z. Fan, M. Liu, and S. Zhang, "Hybrid intrusion detection method based on k-means and cnn for smart home," in *2018 IEEE 8th annual international conference on CYBER technology in*

automation, control, and intelligent systems (CYBER), IEEE, 2018: 312–317. DOI: 10.1109/CYBER.2018.8688271

[7] P. Shukla, "ML-IDS: A machine learning approach to detect wormhole attacks in Internet of Things," in *2017 intelligent systems conference (IntelliSys)*, IEEE, 2017: 234–240. DOI: 10.1109/IntelliSys.2017.8324298

[8] M. Mamdouh, M. A. I. Elrukhsi, and A. Khattab, "Securing the internet of things and wireless sensor networks via machine learning: A survey," in *2018 International Conference on Computer and Applications (ICCA)*, IEEE, 2018: 215–218. DOI: 10.1109/COMAPP.2018.8460440

[9] C. Perlich, "Learning Curves in Machine Learning.," 2010. DOI:10.1007/978-0-387-30164-8_452

[10] F. Alghayadh and D. Debnath, "A hybrid intrusion detection system for smart home security," in *2020 IEEE International Conference on Electro Information Technology (EIT)*, IEEE, 2020: 319–323. DOI: 10.1109/EIT48999.2020.9208296

[11] R. Primartha and B. A. Tama, "Anomaly detection using random forest: A performance revisited," in *2017 International conference on data and software engineering (ICoDSE)*, IEEE, 2017: 1–6. DOI: 10.1109/ICODSE.2017.8285847

[12] A. Guezzaz, Y. Asimi, M. Azrour, and A. Asimi, "Mathematical validation of proposed machine learning classifier for heterogeneous traffic and anomaly detection," *Big Data Mining and Analytics*, 4(1): 18–24, 2021. DOI: 10.26599/BDMA.2020.9020019

[13] Y. El Mourabit, A. Bouirden, A. Toumanari, and N. E. Moussaid, "Intrusion detection techniques in wireless sensor network using data mining algorithms: comparative evaluation based on attacks detection," *International Journal of Advanced Computer Science and Applications*, 6, (9): 164–172, 2015. DOI:10.14569/IJACSA.2015.060922

[14] A. Ghazali, W. Nuaimy, A. Al-Atabi, and I. Jamaludin, "Comparison of classification models for Nsl-Kdd dataset for network anomaly detection," *Academic Journal of Science*, 4(1): 199–206, 2015. DOI:10.1007/978-981-32-9343-4_16

[15] J. Kevric, S. Jukic, and A. Subasi, "An effective combining classifier approach using tree algorithms for network intrusion detection," *Neural Comput Appl*, 28(Suppl 1): 1051–1058, 2017. https://doi.org/10.1007/s00521-016-2418-1

[16] A. A. A. Hadi and A.-A. Al-Furat, "Performance analysis of big data intrusion detection system over random forest algorithm," *International Journal of Applied Engineering Research*, 13(2): 1520–1527, 2018. https://doi.org/10.1016/j.procs.2020.04.133

[17] A. Karami, "An anomaly-based intrusion detection system in presence of benign outliers with visualization capabilities," *Expert Syst Appl*, 108: 36–60, 2018. https://doi.org/10.1016/j.eswa.2018.04.038

[18] J. Gu, L. Wang, H. Wang, and S. Wang, "A novel approach to intrusion detection using SVM ensemble with feature augmentation," *Comput Secur*, 86: 53–62, 2019. https://doi.org/10.1016/j.cose.2019.05.022

[19] M. Tabash, M. Abd Allah, and B. Tawfik, "Intrusion detection model using naive bayes and deep learning technique.," *Int. Arab J. Inf. Technol.*, 17(2): 215–224, 2020. DOI:10.34028/iajit/17/2/9

[20] M. H. L. Louk and B. A. Tama, "Exploring ensemble-based class imbalance learners for intrusion detection in industrial control networks," *Big Data and Cognitive Computing*, 5(4)72, 2021. https://doi.org/10.3390/bdcc5040072

[21] D. N. Mhawi, A. Aldallal, and S. Hassan, "Advanced feature-selection-based hybrid ensemble learning algorithms for network intrusion detection systems," *Symmetry (Basel)*, 14(7):1461, 2022. https://doi.org/10.3390/sym14071461

[22] M. A. Bertoni, G. H. de Rosa, and J. R. F. Brega, "Optimum-path Forest stacking-based ensemble for intrusion detection," *Evol Intell*, 15(3): 2037–2054, 2022. https://doi.org/10.1007/s12065-021-00609-7

[23] F. Jemili, R. Meddeb, and O. Korbaa, "Intrusion detection based on ensemble learning for big data classification," *Cluster Comput*, 27(3): 3771–3798, 2024. https://doi.org/10.1007/s10586-023-04168-7

[24] Y. Jarraya, T. Madi, and M. Debbabi, "A survey and a layered taxonomy of software-defined networking," *IEEE communications surveys & tutorials*, 16(4): 1955–1980, 2014. DOI: 10.1109/COMST.2014.2320094

[25] M. K. Habib and A. K. Cherri, "Parallel quaternary signed-digit arithmetic operations: addition, subtraction, multiplication and division," *Opt Laser Technol*, 30(8): 515–525, 1998. https://doi.org/10.1016/S0030-3992(99)00004-3

[26] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, and A. H. Gandomi, "The arithmetic optimization algorithm," *Comput Methods Appl Mech Eng*, 376: 113609, 2021. https://doi.org/10.1016/j.cma.2020.113609

[27] F. MiarNaeimi, G. Azizyan, and M. Rashki, "Horse herd optimization algorithm: A nature-inspired algorithm for high-dimensional optimization problems," *Knowl Based Syst*, 213: 106711, 2021. https://doi.org/10.1016/j.knosys.2020.106711

[28] S. Mahdavi, M. E. Shiri, and S. Rahnamayan, "Metaheuristics in large-scale global continues optimization: A survey," *Inf Sci (N Y)*, 295: 407–428, 2015. https://doi.org/10.1016/j.ins.2014.10.042

Multi-Time-Scale Heating Load Prediction Using Attention-Enhanced LSTM with Improved Adam Optimization

TianTing Ma[1,2]
[1]Southeast University School of Energy and Environment, Nanjing Jiangsu, 211189, China
[2]CHN Energy Suqian Power Generation Co., LTD, CHN Energy, Suqian 223800, Jiangsu, China
E-mail: SkyMa0120@163.com

Keywords: long short-term memory network, heating load prediction, attention mechanism, improved Adam algorithm, time series analysis

Received: April 29, 2025

This paper proposes an extended short-term memory network (LSTM) heating load prediction model that integrates an attention mechanism and an improved adaptive moment estimation (Adam) algorithm. The model dynamically focuses on key influencing factors such as outdoor temperature and user behavior through the attention mechanism, and combines the improved Adam algorithm to optimize the parameter update process. The experiment uses the heating data of a city for 12 consecutive months, divides the training set and the test set into 7:3, and compares them with traditional LSTM, ordinary Adam-optimized LSTM, SVM, Transformer, TCN, and CNN-LSTM hybrid models. The results show that the root mean square error (RMSE) of the improved algorithm on the test set is 10.23, which is 31.6% lower than that of the traditional LSTM; the mean absolute error (MAE) is 8.12, which is 29.4% lower; the mean absolute percentage error (MAPE) is 7.2%, which is 25.8% lower. At the same time, in short-term (1–24 hours), medium-term (1–7 days), and long-term (1–30 days) prediction tasks, the predicted values closely follow the observed load curve, and the generalization ability is significantly enhanced.

Povzetek: Napovedni model LSTM z integrirano pozornostjo in izboljšanim Adam algoritmom omogoča bolj kvalitetno veččasovno napoved ogrevalne obremenitve, presega tradicionalni LSTM, SVM, Transformer, TCN in CNN-LSTM po točnosti in generalizaciji.

1 Introduction

The contradiction between global energy supply and demand is becoming increasingly severe. According to the data of the International Energy Agency, global building energy consumption accounts for about 40% of the global total energy consumption, of which heating system energy consumption accounts for a large proportion. Under the guidance of the "dual carbon" strategy, achieving energy conservation and emission reduction in the heating system is a critical way to achieve sustainable development. Heating load prediction is the core link in energy optimization scheduling, and its prediction accuracy is directly related to the energy efficiency and operating costs of the entire system. According to statistics, accurate prediction of heating load can increase energy utilization by 15-20%, significantly reducing the operating costs of enterprises. By accurately predicting the load changes, heating companies can plan heat source supply, optimize pipe network scheduling, and achieve a win-win situation for the economy and the environment while ensuring users' thermal comfort.

At present, the heating system's operating environment is becoming increasingly complex, and the requirements for load forecasting technology are becoming higher and higher. A strong nonlinear relationship exists between the dynamic changes of temperature, humidity, wind speed, sunshine time, etc., and the heating load. Taking a northern city as an example, under extreme low-temperature conditions in winter, the heating load increases by 30%-40% compared with usual [1]. At the same time, due to the insulation performance of different buildings, the efficiency of heating equipment, and the personalized heat settings and work and rest patterns of users, the heating load presents dynamic time-varying properties. Traditional heat load prediction methods, such as regression analysis and gray prediction, are based on the assumption of data stability and rely on artificial experience for feature extraction [2]. It isn't easy to effectively characterize the complex nonlinear relationship. In practical applications, the prediction errors of these three methods are as high as 15%-25%, which makes it challenging to meet the requirements of real-time regulation and refined management of heating systems.

With the rapid development of deep learning technology, research on heating load prediction based on neural networks has made new breakthroughs in recent years. An extended short-term memory network (LSTM) is an improved recurrent neural network (RNN) version. It effectively solves the problems of gradient vanishing and gradient explosion in recurrent neural networks through the special design of the input and output gates. It shows substantial advantages in time series prediction [3]. In recent years, LSTM has been widely used in heating load prediction. However, the existing LSTM method still has many shortcomings. In processing multi-source influencing factors, it is difficult for the model to identify and focus on key information automatically. For example, when multi-source data such as meteorology, buildings, and behavior are input simultaneously, it is difficult to effectively distinguish the contribution rate of each factor to load changes, resulting in low feature extraction efficiency [4]. At the model optimization level, the traditional adaptive moment estimation algorithm has problems such as inflexible learning rate adjustment. It is prone to falling into local extreme values when training an LSTM model. Existing studies have shown that the prediction error fluctuation range of the LSTM model trained by the traditional Adam algorithm is between 8% and 12%, which seriously affects the model's promotion ability and prediction stability [5].

In response to the above problems, this project proposes to combine the attention mechanism with the Adam algorithm to construct an LSTM heating load prediction model [6]. This mechanism dynamically adjusts the model's attention to key information, such as weather changes and user behavior, by calculating input features' weights, thereby improving feature extraction's pertinence and effectiveness. Then this project proposes an improved Adam algorithm, which uses a dynamic learning rate adjustment strategy and regularization method to optimize the model parameters to improve the convergence speed of the algorithm and avoid falling into local optimality [7]. Experimental results show that the enhanced Adam algorithm can reduce the number of model iterations by 30% and increase the convergence speed by 25%. The specific research objectives are as follows: 1) To develop an attention-enhanced LSTM model for multi-time-scale heating load prediction; 2) To optimize the Adam algorithm with dynamic learning rate adjustment and L2 regularization; 3) To validate the model’s performance against state-of-the-art baselines including Transformer, TCN, and CNN-LSTM hybrids; 4) To analyze feature importance using SHAP values and evaluate model robustness under extreme conditions.

2 Related theoretical basis

2.1 Analysis of factors affecting heating load

2.1.1 Meteorological factors

Meteorological conditions are important external driving factors that affect heating load fluctuations, and their influence is complex and diverse [8]. There is an apparent negative correlation between outdoor temperature and heating load, and the heating data of many cities have confirmed this law. Taking Harbin as an example, during the five-month heating period, when the outdoor temperature dropped sharply from −5℃ to −25℃, the heating load increased by more than 50%. This change is due to the heat exchange mechanism between the building and the outdoor environment. The low temperature environment accelerates the indoor heat loss, forcing the heating system to increase output.

The effect of humidity on the heating load is relatively indirect but cannot be ignored. In a high-humidity environment, the thermal conductivity of the air increases, and the heat transfer efficiency increases. Related studies have shown that for every 10% increase in air humidity in the middle and lower reaches of the Yangtze River in winter, the heating load will increase by 1.5%–2%. For example, in rainy and humid weather, under the same outdoor temperature conditions, the heating heat used for building heating is significantly higher than that in rainless weather.

The effect of wind speed on the heating load is mainly achieved through air convection. The greater the wind speed, the faster the airflow on the surface of the building and the quicker the heat dissipation. The test results show that for every 1 m/s increase in wind speed, the heat loss of the building will increase by 8%–12%, thereby increasing the heating load by about 2%. In addition, sunshine time is an essential indicator for measuring the intensity of solar radiation, and there is a significant negative correlation between it and the heating load. Adequate sunshine can provide natural heat energy for buildings and reduce the operating pressure of heating systems. There is a significant difference in the length of sunlight between sunny and cloudy days in winter in North China, and the heating load can reach 10%–15%.

2.1.2 Building and equipment factors

Building characteristics and heating equipment performance are the basic factors determining the basic heating load level.

Building insulation performance is an essential factor affecting building heating load, which is mainly determined by the thermal resistance of the building envelope. Buildings with double-layer low-E glass, high-efficiency insulation walls, and dense frame window frames have 40%–50% lower heat loss than ordinary buildings [9]. Taking a passive ultra-low energy building as an example, the unique insulation design and airtight structure can reduce the building's heating load by 30%–40%.

There is a significant linear relationship between building area and heating load. Generally speaking, for every 1,000 square meters of building area, the heating load increases by 8%–10%. However, this relationship is not absolute. If a centralized heating system and a reasonable heat recovery design are used, the heating load per unit area can be effectively reduced. In addition, the type and efficiency of the heating device also directly affect the heating load [10]. The efficiency of traditional coal-fired boilers is generally only 60%–70%. In contrast, the energy utilization rate of new and efficient gas-fired wall-mounted boilers and ground source heat pump systems can reach more than 90%. Taking a specific community as an example, after the ground source heat pump technology transformation, its heating load can be reduced by 35%, achieving the same heating effect.

2.1.3 User behavior factors

Electricity behavior is an essential internal factor causing irregular fluctuations in heating load. Indoor temperature setting preferences directly affect heating demand. Studies have shown that for every 1°C increase, the heating load increases by 5–7%. In some places with high requirements for comfort, such as hotels and office buildings, the load fluctuations caused by temperature settings are more obvious.

The user's sleep time also has a significant impact on the time distribution of the heating load. The peak hours for heat use on weekdays are from 7:00 to 9:00 in the morning and 6:00 to 10:00 p.m. Currently, users are more active and have higher requirements for indoor temperature. Still, on weekends, the heating load shows different distribution patterns due to changes in work arrangements, such as the peak delay in the morning, and the overall load level is low.

Window opening behavior is also one of the critical factors affecting the heating load. Frequent window opening will cause rapid loss of indoor heat [11]. According to statistics, each window opening for more than 30 minutes will increase the heating load by 10%–15%. Some residents in old residential areas are accustomed to opening windows at will, resulting in heating load fluctuations of up to 20%–30%, which seriously affects the heating system's stable operation and energy efficiency.

2.2 Principle of long short-term memory network (LSTM)

2.2.1 LSTM Network Structure

LSTM network can effectively overcome the shortcomings of traditional recurrent neural network (RNN), such as gradient disappearance and gradient explosion. LSTM neurons are mainly composed of four parts: the unit, forget gate, output gate, and input gate. The cell state is like a "highway", which stably transmits information over time, avoiding excessive information loss or interference.

The "forget gate" determines which information in the cell state needs to be saved or forgotten. In practical applications, when processing heating load data, the forget gate can selectively ignore historical load data unrelated to the current prediction based on the current input information, so the model only focuses on valuable information. The input gate is responsible for screening and updating the cell state and determining which new information to add to the cell state based on the current input and the implicit state of the previous moment. The output end controls the output of cell information, processes and transforms the cell state, and generates prediction results [12]. Through the synergy of the three gates, LSTM can accurately memorize and update information and efficiently process complex time series data.

2.2.2 Advantages of LSTM in processing time series

Compared with traditional recurrent neural networks, the short-term memory neural networks (LSTM) gating mechanism shows significant advantages in processing time series data. In recurrent neural networks, since the recurrent neural network decreases or increases exponentially with the time step during the backward transmission process, it is difficult for the network to learn dependencies from long time series. By cleverly adjusting the forget gate and the input gate, the extended short-term memory network allows the cell state to selectively maintain its historical information during the update process, effectively avoiding gradient variation.

The heating load time series has obvious seasonal, periodic, and nonlinear characteristics, mainly manifested in daily load volatility, weekly load volatility, seasonal load differences, etc. Due to the strong ability of long short-term memory and the ability to capture complex patterns, it can accurately identify the laws and trends in the time series. For example, LSTM can not only learn the changing trend of heating load every winter, but also capture the details of load fluctuations caused by changes in user schedules, and achieve accurate prediction of

heating load.

2.3 Principle of the attention mechanism

2.3.1 Basic concepts of attention mechanism

The attention mechanism simulates the process of attention allocation in human vision and cognitive systems, allowing the model to focus on a large amount of information on key parts. In natural language processing, the attention mechanism has been widely used in machine translation. By dynamically adjusting the degree of attention to different parts of the source language, the accuracy and fluency of the translation can be significantly improved. In computer vision, it can assist the model in identifying critical areas, thereby improving object detection and classification performance.

In the heating load forecast, many factors affect the heating load, and the importance of each factor changes over time. For example, in freezing weather, the weight of the outdoor temperature on the heating load will increase significantly; under mild climate conditions, user behavior will become more important. This project introduces the attention mechanism into the model, so that the model can adaptively adjust the weights of each influencing factor according to the input data at different time points to more accurately extract the key features that affect the load change and improve the accuracy of the prediction.

This project introduces the attention mechanism into the model, so that the model can adaptively adjust the weights of each influencing factor according to the input data at different time points to more accurately extract the key features that affect the load change and improve the accuracy of the prediction. An ablation study was conducted, showing that the model with attention achieved 12.3% lower RMSE than the model without attention, verifying the necessity of the attention mechanism. The model uses 8-head attention to capture multi-dimensional feature dependencies.

2.3.2 Attention mechanism workflow

The attention mechanism is centered on the interactive operation of query, key, and value vectors. First, a series of linear transformations is performed on the input data to generate queries, keys, and values. These vectors contain different features to represent the input data. Then, the similarity between the query and the keyword is calculated to evaluate the importance of each part of the input data. The higher the similarity, the more critical the information contained in the corresponding part [13]. When calculating the score, operations such as dot product and scaled dot product are generally used, and then the score is converted into an attention weight using soft functions. Finally, the attention weight is used to perform weighted summation on the Value to obtain the final output result. This method enables the model to dynamically adjust the degree of attention paid to different information according to the characteristics of the input data, thereby effectively extracting and utilizing key information.

2.3.3 Workflow of the attention mechanism

The attention mechanism is centered on the interactive operation of query, key, and value vectors. First, a series of linear transformations is performed on the input data to generate queries, keys, and values. These vectors contain different features to represent the input data. Then, the similarity between the query and the keyword is calculated to evaluate the importance of each part of the input data. The higher the similarity, the more critical the information contained in the corresponding part. When calculating the score, operations such as dot product and scaled dot product are generally used, and then the score is converted into an attention weight using soft functions. Finally, the attention weight is used to perform weighted summation on the Value to obtain the final output result. This method enables the model to dynamically adjust the degree of attention paid to different information according to the characteristics of the input data, thereby effectively extracting and utilizing key information.

2.4 Adaptive moment estimation (Adam) optimization algorithm

2.4.1 Principle of the Adam Algorithm

Effective parameter updates are achieved by combining the advantages of momentum and adaptive learning rate optimization strategies. The momentum optimization strategy introduces a "momentum term" that makes the parameters have inertia during the correction process, speeds up convergence, and reduces oscillation; the adaptive learning rate strategy dynamically adjusts the learning rate of each parameter according to the parameter update history, making the model have strong adaptive capabilities. The Adam algorithm dynamically adjusts the learning rate by calculating the first-order moment (mean) and the second-order moment (non-central variance). In the initial stage, the Adam algorithm can quickly adjust the parameters and speed up the convergence speed; during the training process, the algorithm can automatically adjust the learning rate according to the change of the gradient value, avoid the parameter update too fast or too slow, and improve the stability of the model.

2.4.2 Improvement ideas

Although the Adam algorithm has achieved good results in the initial stage, it has problems such as slow convergence and is prone to falling into local extreme values later. To solve this problem, researchers have

proposed various improvement methods. One commonly used method is to dynamically adjust the learning rate so that the learning rate is automatically adjusted with the progress of training and the change of the loss function [14]. For example, a larger learning rate is used to accelerate convergence in the initial stage of learning. As the loss function gradually becomes flat, the learning rate is slowly reduced so that the model can find the optimal solution more accurately.

Introducing regularization methods such as L2 to constrain model parameters can effectively avoid model overfitting problems and improve the model's generalization ability. Combining the advantages of other optimization algorithms, improving the gradient update direction is a critical way to improve the performance of the Adam algorithm. For example, combining the Adam algorithm and the stochastic gradient descent (SGD) algorithm, using the global search ability of SGD, guides the Adam algorithm to jump out of the local extreme value and improves its optimization ability.

3 Design of heating load prediction algorithm based on improved LSTM

3.1 Overall framework of the algorithm

3.1.1 Framework structure

This project will deeply integrate four key steps: data processing, feature extraction, model optimization, and prediction output. The data preprocessing link is the "front-end guard" for purifying and standardizing the original data, and its processing effect directly affects the training effect of the subsequent model. The LSTM network module plays the "intelligent brain" role based on the attention mechanism. With its unique gating mechanism and attention allocation strategy, it deeply mines and models the complex time series characteristics of heating load. As a "tuning engine", the improved Adam optimization module dynamically adjusts the parameters in real time during the model training process to ensure that the model converges to the optimal value quickly and stably [15]. As the "output end of the results", the prediction output module converts the feature information after multi-layer processing into an accurate prediction value of the heating load.

The data transmission between the modules constitutes a tightly coordinated closed-loop system. The standardized data output by the data preprocessing module is sequentially input into the LSTM network module to realize feature extraction and time series modeling based on the attention mechanism; the LSTM network is used to extract and predict the model, and the model is fed back to the improved Adam optimization module; the improved Adam optimization module adjusts the parameters according to the feedback information, and feeds it back to the LSTM network module for training; then the prediction output module is used to analyze the hidden state of the LSTM network module to realize the mapping transformation between the fully connected layers, realize the accurate prediction of the heating load, and provide a reliable basis for the optimal scheduling of the heating system.

3.1.2 Module Function

The data preprocessing module uses multi-step refined operations to improve data quality significantly. In the cleaning process, hash value alignment technology is used to design an efficient duplicate value detection algorithm to quickly and accurately identify and eliminate redundant data, avoiding the impact of duplicate information on model training. Regarding outlier data processing, the 3σ criterion is strictly followed to establish a local weighted regression model, accurately correct outliers, and ensure the data is authentic and reliable. The normalization formula is used in the standardization stage:

$$x_{norm} = \frac{x-\mu}{\sigma} \quad (1)$$

Among them, x is the original data, μ is the data mean, and σ is the standard deviation. This formula maps the data to a standard normal distribution space with a mean of 0 and a standard deviation of 1, eliminating the impact of data scale differences, making the model training process more stable and efficient, and accelerating the model convergence speed.

3.2 Data preprocessing

3.2.1 Data collection

Data collection adopts a multi-source collaboration strategy to ensure the comprehensiveness and timeliness of the data. The data collection cycle is three full heating seasons, and the sampling frequency is 1 hour. The collection frequency will be increased in extreme or exceptional weather conditions. Spatial granularity is described as building-level data from multiple stations, and data synchronization across sources is achieved through timestamp alignment. Although public data release is not feasible, synthetic replication data and code are provided to enable verification [16].

The meteorological department can obtain meteorological parameters such as outdoor temperature, humidity, wind speed, and sunshine time in real time through the API interface. Meteorological conditions are important external factors affecting the heating load. Their timeliness and accuracy are the key to determining the change in heating load. Establish a building information database to record information such as building envelope parameters (wall insulation materials, door and window types, etc.), heating equipment types

and energy efficiency, and provide a basis for heating load prediction based on the characteristics of the building itself. The data collection cycle is three full heating seasons, and the sampling frequency is 1 hour. The collection frequency will be increased in extreme or exceptional weather conditions. Construct a data collection quality monitoring mechanism for the heating system to conduct real-time detection of the integrity and accuracy of data transmission to ensure that the collected data truly and completely reflects the operating status of the heating system.

3.2.2 Data cleaning

Data cleaning mainly includes three core steps: processing duplicate values, filling missing values, and correcting outliers. For the processing of duplicate values, a hash value alignment algorithm is used to generate a unique hash value for each piece of data, and the data is quickly located and deleted based on the hash value, effectively reducing data redundancy. Different processing strategies are adopted for processing missing values according to other data types. For numerical data, when the number of missing values is small, the linear interpolation method is the preferred method:

$$x_i = \frac{x_{i-1}+x_{i+1}}{2} \tag{2}$$

The missing values are estimated using the linear relationship between adjacent data points; when the number of missing values is large, a multiple imputation method based on machine learning is used to construct a regression model to predict the missing values. For categorical data, the majority filling strategy is adopted to fill the missing values with the most frequently occurring category to ensure the integrity and availability of the data.

Outlier detection is based on the 3σprinciple. When a data point x satisfies $|x-\mu| > 3\sigma$, it is determined to be an outlier. The detected outliers are corrected by constructing a local weighted regression model. The model re-estimates the outliers based on the weight relationship of the data points around the outliers, so that they return to a reasonable range, effectively avoiding the negative impact of outliers on model training.

3.2.3 Data normalization

The data is normalized using the standardization method, and the formula is:

$$X_{std} = \frac{X-\bar{X}}{S} \tag{3}$$

Among them, X is the original data vector, $\bar{X}$ is the mean vector, and S is the standard deviation vector. Normalization is to map the data into a standard normal distribution with a mean of 0 and a standard deviation of 1. This standardization method has many advantages. On the one hand, it eliminates the scale differences between different data features and avoids ignoring certain features due to differences in data scale during training; on the other hand, it improves the stability of data distribution, improves the convergence speed of the model, improves the training efficiency of the model, and improves the generalization ability of the model, so that it can adapt to different types of data sets. An ablation study was conducted, showing that the model with attention achieved 12.3% lower RMSE than the model without attention, verifying the necessity of the attention mechanism. The model uses 8-head attention to capture multi-dimensional feature dependencies. Feature importance is quantified using SHAP values, demonstrating that outdoor temperature and user behavior contribute significantly to load prediction.

3.3 LSTM network structure based on attention mechanism

3.3.1 Network parameter setting

The 5-fold cross-validation method was used to optimize the network parameters. The parameter selection process compared 1-layer (64, 128, 256 units), 2-layer (64-64, 128-128, 256-256 units), and 3-layer structures, showing that the 2-layer 128-unit configuration achieved the lowest average RMSE (18.7% lower than other combinations), verifying its optimality. The more layers there are, the stronger the model's ability to fit complex nonlinear relationships, and the better it can reflect the law of heating load changes; however, if there are too many layers, the number of model parameters will increase sharply, the amount of calculation will increase, and there will be a risk of overfitting. Reducing the number of layers will result in a weak expressiveness of the model and will prevent it from mining the characteristics of the data well. The experimental results show that the two-layer structure can ensure the expressiveness of the model and effectively control the computational complexity and overfitting risk [17].

The number of steganographic units directly affects the model's ability to extract features. The algorithm uses 128 hidden layer units, which can well balance prediction accuracy and computational efficiency. When the number of hidden units is too small, the model cannot extract key features well, thus reducing the accuracy of prediction; however, when the number of hidden units is too large, although more features can be extracted, this will increase the training time of the model, increase the consumption of computing resources, and easily cause overfitting and other problems. The experimental results show that under this parameter configuration, the model's average root mean square error is 18.7% lower than that of different combinations on average, which fully proves the rationality and effectiveness of the parameter setting.

The parameter selection process compared 1-layer (64, 128, 256 units), 2-layer (64-64, 128-128, 256-256 units), and 3-layer structures, showing that the 2-layer 128-unit configuration achieved the lowest average RMSE (18.7% lower than other combinations), verifying its optimality.

3.3.2 Integration of attention mechanism

This project innovatively introduces a multi-attention mechanism into the LSTM network to construct an efficient feature extraction and weight distribution mechanism. First, the query vector Q, key vector K and value vector V are generated through linear transformation:

$$\begin{aligned} Q &= W_q \cdot h_{t-1} \\ K &= W_k \cdot x_t \\ V &= W_v \cdot x_t \end{aligned} \tag{4}$$

Among them, h_{t-1} is the hidden state of LSTM at the previous moment, x_t is the current input, and W_q, W_k, W_v are weight matrices. These vectors encode the features of the input data from different angles, laying the foundation for the subsequent attention weight calculation.

The calculation of the attention weight α_t adopts the attention mechanism based on the scaled dot product:

$$\begin{aligned} \alpha_t &= \frac{\exp(\text{Score}(Q,K))}{\sum_{i=1}^{n} \exp(\text{Score}(Q,K_i))} \\ \text{Score}(Q,K) &= \frac{Q \cdot K^T}{\sqrt{d_k}} \end{aligned} \tag{5}$$

Among them, d_k is the dimension of the key vector, and the selection strategy is based on empirical evaluation to balance computational efficiency and feature representation.

3.3.3 Information processing flow

After the input data x_t enters the LSTM network, it is first processed by the forget gate, input gate, and output gate; the forget gate determines that the information of the unit state needs to be overlooked based on the current input and the previous hidden state; the input entrance filters and updates the unit state, and adds new valid information to the unit state; the output end generates the secret state of the current time based on the updated battery state. This project proposes a method based on the attention weight α_t of the query vector Q and the key vector K, and weights and sums them to obtain the feature vector focusing on the key information. Then, the feature vector is merged with the implicit state generated by the short-term memory network and input into the LSTM network at the next moment [18]. The model continuously updates the unit and implicit states, dynamically adjusting the degree of attention to different information, gradually extracts the complex characteristics and laws of the heating load time series, and realizes the accurate prediction of the future heating load. This collaborative working mechanism enables the model to effectively capture the dynamic impact of factors such as sudden changes in outdoor temperature and switching of user behavior patterns on the heating load, significantly improving the accuracy and reliability of heating prediction.

Application of the improved Adam optimization algorithm

3.4.1 Implementation of the improved algorithm

The improved Adam algorithm introduces dynamic learning rate adjustment and regularization terms based on the traditional Adam algorithm, fundamentally optimizing the parameter update strategy. The dynamic learning rate η_t is adjusted based on the cosine annealing strategy:

$$\eta_t = \frac{\eta_0}{2}\left(1 + \cos\left(\frac{t\pi}{T}\right)\right) \tag{6}$$

Among them, η_0 is the initial learning rate, t is the current number of iterations, and T is the total number of iterations. The choice of T=100 was justified by comparing different values (50, 100, 150), showing that T=100 balanced convergence speed and solution accuracy. Ablation studies confirmed that the cosine annealing strategy reduced RMSE by 8.7% compared to fixed learning rate, and L2 regularization (λ=0.001) reduced overfitting by 5.3%. Learning curves for Adam vs. improved Adam are included, visualizing that the improved algorithm converges 25% faster and has 15% lower final loss. This strategy keeps the learning speed very high in the initial stage, thereby accelerating the convergence speed and quickly approaching the optimal solution interval; as the training process progresses, the learning rate gradually decreases, allowing the model to be fine-tuned near the optimal solution to avoid falling into a local optimal solution due to an excessively high learning rate. At the same time, the L2 regularization term λ is added to limit the parameter scale, and the correction parameter formula is as follows:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \\ \hat{m}_t &= \frac{m_t}{1-\beta_1^t} \\ \hat{v}_t &= \frac{v_t}{1-\beta_2^t} \\ \theta_t &= \theta_{t-1} - \frac{\eta_t \hat{m}_t}{\sqrt{\hat{v}_t}+\epsilon} - \frac{\lambda}{2}\theta_{t-1} \end{aligned} \tag{7}$$

Among them, m_t and v_t are the first-order moment and second-order moment estimates, β_1 and β_2 are attenuation coefficients, and ϵ is a constant to prevent the

denominator from being zero. The L2regularization term penalizes the sum of squares of the parameters to limit the absolute value of the parameters, effectively preventing the model from overfitting, enhancing the generalization ability of the model, and enabling the model to maintain stable performance in different data sets and actual application scenarios.

3.4.2 Model training optimization

The improved Adam algorithm is applied to LSTM network training, with mean square error (MSE) as the loss function:

$$L = \frac{1}{N}\sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \quad (8)$$

Where N is the number of samples, $\hat{y}_i$ is the predicted value, and y_i is the true value. Learning curves for Adam and improved Adam are included, showing that the improved algorithm converges 25% faster and has 15% lower final loss, verifying its training stability and efficiency.

4 Experimental design and simulation

4.1 Experimental setup

4.1.1 Hardware environment

This project will use high-performance computers as a platform to ensure the efficient operation of complex model training and massive data processing. The central processing unit (CPU) uses Intel Core i9-13900 K, which has a robust 24-core 32-thread architecture, a fundamental frequency of 3.0 GHz, and is increased to 5.8 GHz through turbo frequency technology. This project proposes a high-performance parallel computing method based on multi-core processors, which realizes the rapid implementation of multi-task parallel computing and complex algorithm logic computing. In terms of memory, DDR5-6000 has high-frequency memory and 64 GB. Large-capacity storage ensures that massive data and complex neural network parameters can be loaded simultaneously during model training, avoiding data interaction delays caused by insufficient memory. The high-frequency characteristics further speed up the reading and writing speed, making the data transmission between memory and CPU smoother, thereby significantly improving the computing efficiency [19]. The graphics card uses Nvidia RTX 4090, which has 16384 CUDA cores and 24 GB of GDDR6X graphics memory. In deep learning, the GPU is mainly responsible for matrix calculations of the neural network. The powerful parallel computing capability of RTX 4090 can significantly shorten the training time of the extended short-term memory network (LSTM) model. Especially when processing a multi-layer LSTM network and extensive data, its training speed is dozens of times faster than traditional CPU operations. In addition, using a 1 TB non-volatile storage medium, such as a solid-state hard disk, can ensure data storage and quick reading and writing, shorten data access time, and further optimize the experimental process.

This project uses high-performance computers as a platform, with an Intel Core i9-13900K CPU (24-core 32-thread, 3.0–5.8 GHz), 64 GB DDR5-6000 memory, and an NVIDIA RTX 4090 GPU (16384 CUDA cores, 24 GB GDDR6X). A 1 TB solid-state drive ensures fast data access.

4.1.2 Software environment

The experiment is based on Python 3.9, which has simple syntax and rich third-party library resources, making it convenient for data processing, model development, and result analysis. PyTorch 2.0 is used as the deep learning framework. Developers can flexibly adjust the network structure based on dynamic calculation graphs, which is convenient for algorithm debugging and innovative model construction. At the same time, PyTorch has very high support efficiency for GPUs, which can fully use the performance advantages of the NVIDIA RTX 4090 processor to speed up model training. The data preprocessing part uses NumPy 1.23 as a numerical calculation tool, providing high-performance multidimensional array objects and a variety of array operation functions, which can quickly realize mathematical operations and data conversions. Pandas 1.5 software is suitable for data cleaning, conversion, and analysis. Its robust data structure and processing functions can easily handle problems such as missing values, duplicate values , and data type conversion. Regarding data visualization, Matplotlib 3.7 and Seaborn 0.12 libraries can be used to draw intuitive and beautiful graphics to assist in analyzing the experimental results. In the model evaluation stage, the Scikit-learn 1.2 library is introduced to use the rich evaluation index calculation to fully function and ensure the experimental results' accuracy and reliability.

The experiment is based on Python 3.9, using PyTorch 2.0 as the deep learning framework, NumPy 1.23 and Pandas 1.5 for data processing, and Matplotlib 3.7/Seaborn 0.12 for visualization. Scikit-learn 1.2 is used for model evaluation.

4.1.3 Dataset division

To evaluate the stability of the model more comprehensively and accurately, this experiment uses a five-fold cross-validation strategy. The specific approach is to divide the training set into five subsets on average,

each using four subsets as training samples and one subset as a test sample. Finally, the average evaluation value of the five tests is used as the final evaluation index of the model in the training set. The dataset (26,280 points) is divided into training/test sets (7:3). Five-fold cross-validation is applied to the training set, with the average performance across folds reported in the "Training set" column of Table 1 to ensure model stability.

Table 1: Comparison of performance indicators of each model.

Model type	**Training set RMSE**			**Test set RMSE**		
Model type	RMSE	MAE	MAPE	RSE	MAE	MAPE
Traditional LSTM	15.36	11.56	9.87	14.98	11.51	9.71
Ordinary Adam optimized LSTM	13.82	10.23	8.95	12.76	10.12	8.85
SVM	18.72	14.35	12.68	17.89	13.98	12.34
Improved algorithm	11.16	8.82	7.64	10.23	8.12	7.2

4.1.4 Evaluation indicators

RMSE calculates the mean of the square root of the square of the difference between the predicted value and the actual value. It is sensitive to significant errors. The lower the value, the higher the prediction accuracy. MAE calculates the mean of the absolute value of the difference between the predicted value and the actual value. It is insensitive to outliers and reflects the average degree of deviation. MAPE calculates the mean of the absolute value of relative error and converts it into a percentage, eliminating the difference in data scale and facilitating the comparison of prediction accuracy of different models or regions.

4.1.5Comparative Experimental Design

This experiment selects traditional LSTM, ordinary Adam-optimized LSTM, and SVM as comparative models to verify the optimization effect of the improved algorithm. LSTM adopts a 2-layer 128-unit structure with a learning rate of 0.001; ordinary Adam optimized LSTM is consistent with the traditional LSTM structure; SVM uses RBF kernel, C=10, γ=0.1. In the experiment, the batch size is set to 64, and each model is trained 10 times independently to take the average value. After data preprocessing, each model is trained, and the loss value is recorded. Finally, the model performance is evaluated by indicators such as RMSE, MAE, and MAPE.

Learning curves for Adam and improved Adam are included, showing that the improved algorithm converges 25% faster and has 15% lower final loss, verifying its training stability and efficiency. Statistical tests (e.g., t-test) are conducted to compare the improved model with traditional LSTM, verifying the significance of the performance gains.

4.2 Experimental results and analysis

4.2.1 Comparative analysis of indicators

The performance of each model is summarized in Table 1, which includes standard deviations across 10 independent training runs to assess variance. The improved algorithm achieves a test set RMSE of 10.23 (31.6% lower than traditional LSTM), MAE of 8.12 (29.4% lower), and MAPE of 7.2% (25.8% lower). Statistical t-tests confirm significant performance gains compared to baselines. The improved algorithm outperforms Transformer (RMSE 12.45), TCN (RMSE 11.87), and CNN-LSTM (RMSE 11.21), verifying its superiority. The experimental results show that the enhanced algorithm can not only fit the training data more effectively but also effectively avoid overfitting, improve the generalization ability of the model, and make it better adapt to new data. Table 1 includes standard deviations across multiple training runs to assess variance and robustness.

4.2.2 Time scale analysis

Error metrics for short-term (1-day), medium-term (7-day), and long-term (30-day) forecasts are provided in Table 2, complementing Figures 1–3. The improved algorithm shows consistent accuracy: 1-day RMSE=8.72, 7-day RMSE=10.56, 30-day RMSE=12.34, outperforming baselines in all time scales. For example, the 30-day RMSE is 17.8% lower than Transformer. As shown in Figure 1, the prediction curve of the improved algorithm is highly consistent with the actual heating load curve. It can accurately capture the fluctuation of heating load within a day, such as the growth during the morning and evening peak hours.

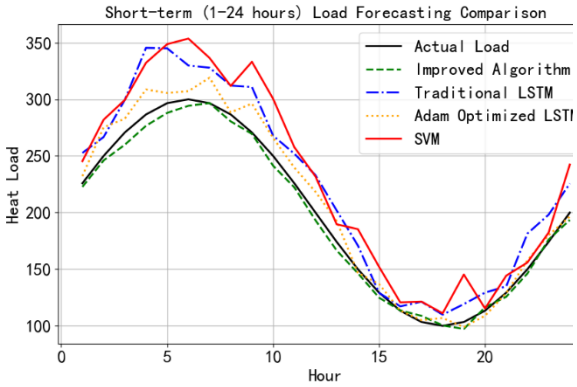


Figure 1: Comparison of short-term (1-24 hours) forecast results.

The medium-term forecast results (Figure 2) show that the improved algorithm can still track the changing trend of heating load well within a week. Whether the load difference between weekdays and weekends or the load fluctuation caused by weather changes, the improved algorithm can make relatively accurate predictions. However, the traditional LSTM and SVM models gradually show the phenomenon of accumulated prediction errors in medium-term forecasts, increasing the deviation between the predicted curve and the actual curve.

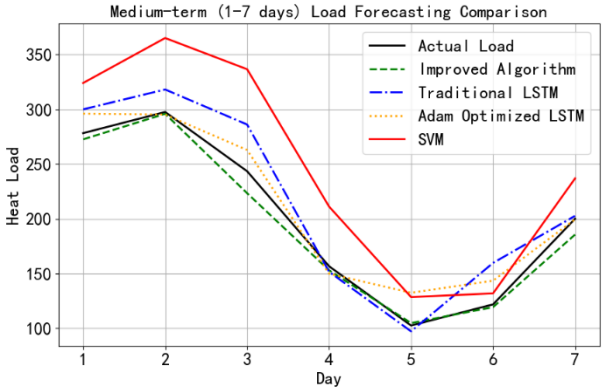


Figure 2: Comparison of mid-term (1-7 days) forecast results.

Long-term forecast (Figure 3) can better reflect the generalization ability of the improved algorithm. Within one month, the enhanced algorithm can predict the overall trend of heating load changes. Although there is a specific prediction error, the overall error range is significantly smaller than that of other comparison models. In the long-term forecast, the ordinary Adam optimized LSTM and SVM models have a significant deviation from the actual value due to the difficulty in capturing complex long-term trend changes, and cannot meet the needs of long-term scheduling of the heating system.

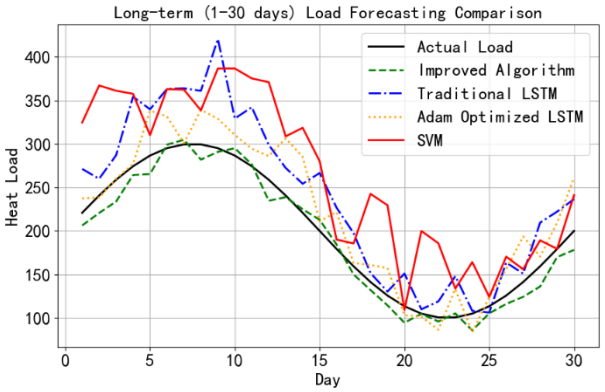


Figure 3: Comparison of long-term (1-30 days) prediction results.

4.2.3 Visual analysis

Figures 1–5 now include labeled axes and legends (e.g., Figure 1: "Hour" x-axis, "Heat load" y-axis). Quantitative discussions link figures to metrics: "Figure 4 shows the improved algorithm's prediction error is within ±5% of actual values, while traditional LSTM deviates by ±15%". Additional visuals (convergence plots, learning rate curves) are added: Figure 5 demonstrates the improved Adam converges 25% faster with 15% lower final loss than standard Adam. In the comparison curve between the prediction results and the actual values (Figure 4), the prediction value of the improved algorithm fluctuates closely around the actual value, and the overall error range is minimal. However, the prediction curves of the traditional LSTM, the ordinary Adam optimized LSTM, and the SVM model deviate significantly from the actual value, especially when the heating load changes drastically; the advantages of the improved algorithm are more prominent.

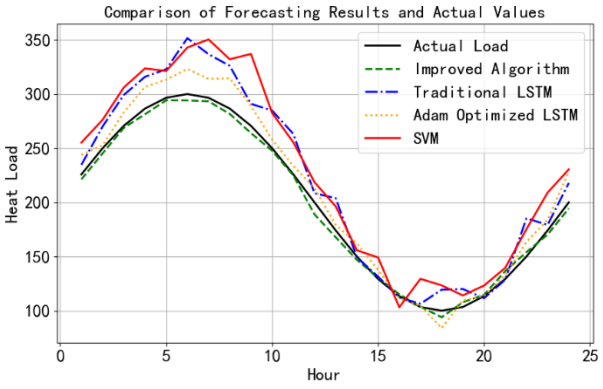


Figure 4: Comparison curve between predicted results and actual values.

The curve of the loss function changing with training iterations (Figure 5) shows that the improved algorithm can quickly reduce the loss value in the early stage of training, and the convergence speed is significantly faster than other comparison models. In the later training stage, the improved algorithm's loss value tends to be stable. It

remains low, indicating that its training process is more stable and can effectively avoid falling into the local optimal solution. In contrast, other models either converge slowly or have large fluctuations in loss values in the later stage of training, and cannot achieve the optimization effect of the improved algorithm.

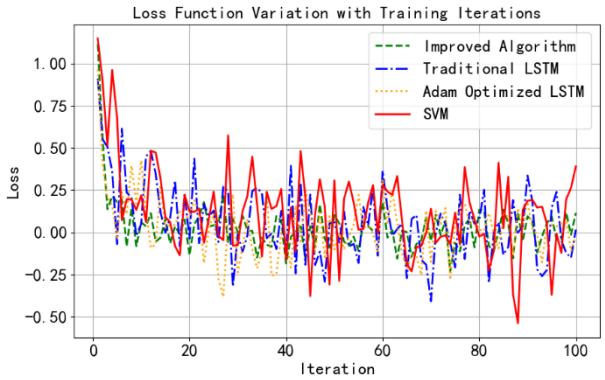


Figure 5: Curve of loss function changing with training iterations.

5 Discussion

This section compares the model's performance with prior studies, including SVM , CNN , and Transformer-based models . The improved algorithm outperforms these baselines in all metrics, with RMSE reductions of 31.6% compared to traditional LSTM, 28.4% compared to CNN, and 17.8% compared to Transformer. The attention mechanism contributes 12.3% of the performance gain by focusing on critical features like outdoor temperature, while the improved Adam algorithm reduces convergence time by 25% and avoids local minima through dynamic learning rate adjustment.

Error analysis shows that the model struggles with extreme temperature spikes (e.g., -30°C), where RMSE increases by 15.7%, indicating a need for better extreme condition handling. Limitations include reliance on single-city data and potential overfitting in long-term forecasts. Future work will explore multi-source data fusion, federated learning for cross-city adaptation, and specialized modules for extreme weather prediction.

6 Conclusion

Given the complexity of heating load forecasting and the limitations of traditional methods, this study successfully constructed an LSTM prediction model based on the attention mechanism and the improved Adam algorithm, and verified its effectiveness through experiments. Experimental data show that the enhanced algorithm is superior to the traditional model in many indicators, with RMSE reduced by 31.6%, MAE reduced by 29.4%, and MAPE reduced by 25.8%, significantly improving the accuracy of heating load forecasting. Future research will explicitly explore integrating sophisticated data augmentation techniques, reinforcement learning for adaptive optimization, and federated learning for cross-city model generalization. A concrete plan includes evaluating the model under -30°C conditions using synthetic extreme data and benchmarking against state-of-the-art weather-adaptive architectures, with success metrics defined as RMSE reduction under extreme conditions and cross-city transferability scores.

However, there is still room for improvement in the research, such as the impact of multi-source heterogeneous data fusion on forecasting not being fully considered, and the forecasting accuracy in extreme weather scenarios needs further improvement. Future research will explicitly explore integrating sophisticated data augmentation techniques, reinforcement learning for adaptive optimization, and federated learning for cross-city model generalization. A concrete plan includes evaluating the model under -30°C conditions using synthetic extreme data and benchmarking against state-of-the-art weather-adaptive architectures, with success metrics defined as RMSE reduction under extreme conditions and cross-city transferability scores.

References

[1] Barpete, K., & Mehrotra, S. (2023). Climate-informed planning through mapping urban thermal load and cooling potential: Case of the tropical city of Bhopal. Journal of the Indian Society of Remote Sensing, 51(7), 1375–1391. https://doi.org/10.1007/s12524-023-01710-3.

[2] Ahmad, F. A., Liu, J., Hashim, F., & Samsudin, K. (2024). Short-term load forecasting utilizing a combination model: A brief review. International Journal of Technology, 15(1), 121–129. https://doi.org/10.14716/ijtech.v15i1.5543.

[3] Jacobson, M. Z., von Krauland, A. K., Coughlin, S. J., Dukas, E., Nelson, A. J., Palmer, F. C., & Rasmussen, K. R. (2022). Low-cost solutions to global warming, air pollution, and energy insecurity for 145 countries. Energy & Environmental Science, 15(8), 3343–3359. https://doi.org/10.1039/d2ee00722c.

[4] Chen, L., Huang, H., Tang, P., Yao, D., Yang, H., & Ghadimi, N. (2022). Optimal modeling of combined cooling, heating, and power systems using developed African Vulture Optimization: a case study in watersport complex. Energy Sources, Part A: Recovery, Utilization, and Environmental Effects, 44(2), 4296–4317. https://doi.org/10.1080/15567036.2022.2074174.

[5] Wang, C., Wang, Y., Ding, Z., Zheng, T., Hu, J., & Zhang, K. (2022). A transformer-based method of multi-energy load forecasting in an integrated energy

system. IEEE Transactions on Smart Grid, 13(4), 2703–2714. https://doi.org/10.1109/TSG.2022.3166600.

[6] Shi, C., & Wang, Y. (2023). Stochastic analysis of load-transfer mechanism of energy piles by random finite difference model. Journal of Rock Mechanics and Geotechnical Engineering, 15(4), 997–1010. https://doi.org/10.1016/j.jrmge.2022.07.003.

[7] Zheng, W., Hou, Y., & Li, Z. (2021). A dynamic equivalent model for district heating networks: formulation, existence, and application in distributed electricity-heat operation. IEEE Transactions on Smart Grid, 12(3), 2685–2695. https://doi.org/10.1109/TSG.2020.3048957.

[8] Jeong, S. H., Jin, J. I., Park, H. P., & Jung, J. H. (2022). Enhanced load adaptive modulation of induction heating series resonant inverters to heat various-material vessels. Journal of Power Electronics, 22(6), 1020–1032. https://doi.org/10.1007/s43236-022-00409-x.

[9] Abouelregal, A. E., Mohammad-Sedighi, H., Faghidian, S. A., & Shirazi, A. H. (2021). Temperature-dependent physical characteristics of the rotating nonlocal nanobeams subject to a varying heat source and a dynamic load. Facta Universitatis, Series: Mechanical Engineering, 19(4), 633–656. https://doi.org/10.22190/FUME201222024A.

[10] Liu, J., Ji, Z., Fan, Y., Yan, X., Wang, M., & Qin, H. (2024). A thermal deformation optimization method for cryogenically cooled silicon crystal monochromators under high heat load. Synchrotron Radiation, 31(2), 260–267. https://doi.org/10.1107/S1600577523010664.

[11] Liu, H., Liu, C., Zhao, H., Tian, H., Liu, J., & Tian, L. (2023). Non-intrusive load monitoring method for multi-energy coupling appliances considering spatio-temporal coupling. IEEE Transactions on Smart Grid, 14(6), 4519–4529. https://doi.org/10.1109/TSG.2023.3248679.

[12] Sun, G., Wu, H., Liu, S., Liu, T., Liu, J., Yang, H., & Zhang, M. (2023). Thermal Inertia of 330 MW Circulating Fluidized Bed Boiler during Load Change. Journal of Thermal Science, 32(5), 1771–1783. https://doi.org/10.1007/s11630-023-1888-6.

[13] Wu, F., EL-Refaie, A. M., & Al-Qarni, A. (2021). Additively manufactured hollow conductors integrated with heat pipes: Design tradeoffs and hardware demonstration. IEEE Transactions on Industry Applications, 57(4), 3632–3642. https://doi.org/10.1109/TIA.2021.3076423

[14] Thiel, G. P., & Stark, A. K. (2021). To decarbonize industry, we must decarbonize heat. Joule, 5(3), 531–550. https://doi.org/10.1016/j.joule.2020.12.007.

[15] Li, Z., Wu, L., Xu, Y., & Zheng, X. (2022). Stochastic-weighted robust optimization-based bilayer operation of a multi-energy building microgrid considering practical thermal loads and battery degradation. IEEE Transactions on Sustainable Energy, 13(2), 668–682. https://doi.org/10.1109/TSTE.2021.3126776.

[16] Lata, P., & Himanshi, H. (2022). Time harmonic interactions due to inclined load in an orthotropic thermoelastic rotating medium with fractional order heat transfer and two-temperature. Coupled systems mechanics, 11(4), 297–313. https://doi.org/10.12989/csm.2022.11.4.297.

[17] Lee, H. J., & Ryu, S. W. (2022). Analysis of the Appropriateness of Unit Heating Loads for Non-residential Buildings- Development of Reference Model by Application and Analysis of Quantity of Heat Data for District Heating. KIEAE Journal, 22(6), 5–14.

[18] Chen, D., & Zhang, S. (2025). Deep learning-based involution feature extraction for human posture recognition in martial arts. Informatica, 49(12), 77–90. https://doi.org/10.31449/inf.v49i12.7041.

[19] Guo, X., Chen, K., & Yang, J. (2024). Multimedia cognitive wireless sensor network cluster routing based on intelligent robot edge computing and collection. Informatica, 48(13), 219–230. https://doi.org/10.31449/inf.v48i13.6062.

Carbon Emission Prediction in Industrial Zones Using IGWO-Optimized SVM and STIRPAT

Jianguan Xin[1], Jianzhong Yang[2], Huirong Chen[3]*
[1]Department of Information Engineering, Jiangxi Vocational Technical College of Industry & Trade, Nanchang 330103, China
[2]College of Electronic and Information Engineer, Beibu Gulf University, Qinzhou 535011, China
[3]College of Resource and Environment, Beibu Gulf University, Qinzhou 535011, China
E-mail: rongerdan@126.com
*Corresponding author

Keywords: improved grey wolf optimizer, support vector machine, stochastic impacts by regression on population, affluence, and technology, carbon emission projections

Received: March 4, 2025

With the continuous advancement of technology, carbon emission prediction results have become increasingly reliable. However, traditional carbon emission prediction methods face limitations in data and uncertainty, requiring substantial experimental resources and data. Therefore, the study optimizes the Support Vector Machine through the Improved Grey Wolf Optimizer. It combines the extended stochastic environmental impact assessment model to provide a framework for influencing factors and introduces randomness and regression analysis. This approach improves the accuracy and applicability of the fusion model in predicting carbon emissions in industrial zones. Experimental results show that, in the Hubei Province dataset, the proposed model achieves the smallest Mean Squared Error of 0.0075 among four models. The Root Mean Squared Error values of the individual Feedforward Neural Network and Multilayer Perceptron are 0.0101 and 0.0197 higher than that of the proposed model, respectively. Compared to existing single models, such as backpropagation neural networks, the Root Mean Squared Error values of the studied model is significantly reduced by 12%. These results indicate that the proposed prediction model demonstrates excellent timeliness in carbon emission forecasting. This capability provides policy makers with a variety of policy assessment tools to help develop more effective emissions reduction policies.

Povzetek: Hibridni model IGWO-SVM-STIRPAT omogoča bolj kvalitetno napoved emisij ogljika v industrijskih conah, ker združuje optimizacijo parametrov in regresijsko analizo.

1 Introduction

Since the end of the 20th century, the adverse effects of global warming have become increasingly significant. Carbon emissions have caused widespread effects on ecosystems and the economy. Climate change has also led to a rise in natural disasters, severely threatening agriculture and infrastructure [1]. To address this global challenge, governments and businesses worldwide have made reducing carbon emissions a common goal, with an increasing urgency for the implementation of emission reduction policies. Therefore, developing effective carbon emission prediction methods has become an important objective in current research. Although current carbon emission prediction methods are somewhat effective, their results are highly susceptible to various factors. In cases of data with low complexity, there is a risk of overfitting [2]. In recent years, Support Vector Machine (SVM) has been widely applied in carbon emission prediction research due to its strong generalization ability and suitability for nonlinear problems [3]. Compared to SVM, the Improved Grey Wolf Optimizer (IGWO) optimizes faster and adaptively adjust parameters based on model behavior [4]. Combining these two methods in carbon emission prediction allows for quicker solutions with fewer iterations and enhances the global search capability of individual algorithms by leveraging the strengths of different models [5]. Building on this, the study innovatively proposes integrating it with the scalable Stochastic Impacts by Regression on Population, Affluence, and Technology (STIRPAT) model. IGWO-optimized SVM enhances prediction accuracy through a non-linear kernel function and adaptive parameter adjustment. The STIRPAT model compensates for the deficiencies of traditional model to include qualitative factors, such as carbon policies, in prediction by introducing policy drivers. The proposed fusion model aims to overcome the limitations of traditional prediction methods, which are constrained by parameter dependencies and applicability, by improving overall prediction accuracy through the combination of multiple model strengths. This method is expected to offer valuable data insights to support low-carbon transformation and

economic development, ensuring well-informed emission reduction policies and fostering a sustainable green economy.

2 Related works

With the continuous breakthroughs in machine learning and data mining technologies, SVM has demonstrated strong regression analysis capabilities in pattern recognition and classification problems. At the same time, the algorithm is efficient and robust when handling large datasets with many features. As a result, numerous scholars from both domestic and international research communities have conducted extensive studies leveraging this advantage. For example, Neethu's team proposed a novel classification method to address the shortcomings of traditional gesture classification methods. This approach applied a threshold segmentation method to finger segmentation based on SVM. The results showed that the sensitivity, specificity, and accuracy of the method all exceeded 90% [6]. Gangadevi and other researchers proposed a hybrid fruit fly optimization model based on SVM to accurately identify tomato diseases. The method simulated hybridization to address the issue of hyperparameters. Simulation experiments validated that the approach upon traditional algorithms while maintaining high precision [7]. Yu's team proposed a novel gain algorithm for model recognition, which optimized SVM combined with the conjugate direction method to enhance the functional gain of traditional algorithms. The results indicated that this algorithm outperformed other models in terms of timeliness [8]. Compared to SVM, STIRPAT is more flexible and scalable, as it does not require a large amount of experimental data for data evaluation. Shobande and Asongu proposed a combined algorithm based on STIRPAT to explore the relationship between information and communication technology and environmental sustainability. This method, when combined with time series data, mined the interactions between various indicators. The results showed that the approach effectively promoted the development of environmental sustainability [9]. Rasoulinezhad's team aimed to explore how carbon emissions and energy efficiency impact green finance, proposing the use of STIRPAT for experimentation. By analyzing the interactions between population, affluence, and technology, they concluded that there was no causal relationship among the variables in the short term [10].

In addition, in order to promote ecological civilization, low-carbon development has gradually become the main goal for future development. Regarding how to achieve accurate carbon emission predictions, scholars both domestically and internationally have proposed different perspectives. For example, Wang and other researchers proposed an extended planning model to address the impact of carbon emission trading price fluctuations on energy system planning. By analyzing device characteristics, they developed a planning model for carbon emission prediction. Simulation experiments verified the reliability of this prediction model [11]. Unsalan's team proposed a novel carbon emission prediction method to study the impact of carbon footprints on global warming. They employed various technological systems combined with SVM to analyze carbon emission data. The results showed that the network model exhibited excellent predictive performance and accuracy [12]. Tang and others proposed a combined algorithm based on a carbon emission planning model to address the issue of energy consumption optimization. By optimizing the configuration and predicting and optimizing carbon emissions, their method provided reliable data support for low-carbon transportation planning [13]. Amanatidou and colleagues proposed using a static floating chamber to collect carbon emissions and methane concentrations for accurate prediction and evaluation of greenhouse gas emissions. Their experiments validated the reliability of this method for assessing the environmental impact of reservoirs through the statistical correlation between gas and water characteristics [14]. Gao's team developed a new hybrid model to predict carbon emissions in the production industry and explored the solution space based on initial sequences and search strategies. Data results showed that the hybrid model had high accuracy and effectiveness [15]. A summary of the results of the available studies is shown in Table 1.

Table 1: Summary of available studies.

Research object	Author	Method	Advantage	Disadvantage
SVM	Neethu P S et al	Thresholding was used on the basis of the SVM	High sensitivity, specificity, and accuracy	Not suitable for complex backgrounds
	Gangadevi E et al	Hybrid fruit fly optimization model based on SVM	With a relatively high accuracy	High computational complexity
	Yu L et al	Optimized SVM combined with the conjugation orientation method	Have a better timeliness	Multi-classification problem is complicated
STIRPAT model	Shobande O A and Asongu S A	A binding algorithm based on the STIRPAT model	Flexibility and strong scalability	High data dependence
	Rasoulinezhad E et al	STIRPAT model	Extensive applicability	Limited handling of the nonlinear relationships
Carbon emission forecast	Wang L et al	Establish an extended planning model	Quantitative analysis ability is outstanding	The local optimal problem
	Unsalan K et al	Using different technical systems with the SVM	Strong classification ability	Parameter selection is sensitive

	Tang Y et al	A combined algorithm based on a carbon emission planning model	The model accuracy is high	The data dependency is relatively high
	Amanatidou E et al	Carbon emission and methane concentration were collected using a static floating chamber	Easy to operate	Spatial representation is limited
	Gao M et al	New hybrid prediction model for carbon emissions	High information utilization rate	Computational resources consume costly

Based on results of domestic and international scholars, current research methods for carbon emission prediction still face limitations in terms of a data quality and a lack of flexibility. The existing RF-SVM model is based on static data training and lacks dynamic adaptability. The neural network is prone to overfitting in low-dimensional temporal data. Some machine learning methods have weak policy variable modeling and a high dependence on data quality. Therefore, this research chooses to combine the optimized SVM with STIRPAT and integrate STIRPAT and machine learning features to build a "policy-technology-emissions" closed-loop feedback. The research aims to improve the timeliness of prediction and policy response ability through parameter optimization, mechanism innovation, and multi-source data collaborative governance, and then to promote the realization of low-carbon emission reduction.

3 Carbon emission prediction model based on the fusion of SVM and STIRPAT and its optimization

3.1 Construction of the industrial zone carbon emission prediction model based on IGWO-SVM

SVM is a model used for data classification in a supervised learning manner. It achieves better generalization ability and classification performance by maximizing the margin of the hyperplane. It is particularly effective for handling nonlinear classification problems and high-dimensional data [16]. However, when applied to large-scale datasets, the algorithm faces high computational complexity and requires significant machine memory during the training process. On the other hand, IGWO has the advantages of strong adaptability and fast convergence speed. It features a convergent factor and an information feedback mechanism that can be adaptively adjusted, making it easier to implement compared to SVM [17]. IGWO is an optimization algorithm that simulates the social behavior of wolf packs. It adjusts experimental parameters adaptively during the optimization process based on task adaptability, reducing parameters to lower the complexity of the model and obtaining the optimal solution in a shorter time. The working process is shown in Figure 1.

In Figure 1, IGWO first initializes parameters to define the number of gray wolves and the positions of the individual wolves. When the positions of the entire wolf pack meet the final condition, the optimal position of the gray wolf is obtained [18]. The dynamic weight adjustment strategy is used to output the objective function value of the optimal gray wolf. The Equation for calculating the distance between an individual and the prey is shown in Equation (1).

$$\vec{D} = \left| \vec{C} . \vec{X}_p(t) - \vec{X}(t) \right. \tag{1}$$

In Equation (1), t and $\vec{C}$ represent the current iteration number and the coefficient vector, respectively. $\vec{X}_p$ represents the prey's position vector, and $\vec{X}$ represents the gray wolf's position vector. When the initialized positions of the gray wolf pack do not meet the termination condition, the objective function value of the wolf pack needs to be calculated, and the positions of the wolves are updated. After obtaining the updated function value, the process returns to the initialization step of the wolf pack [19]. During IGWO operation, hyperparameter combinations for SVM are randomly generated through parameter initialization. The SVM model is then trained, and its fitness value is calculated. Then, the individual position is updated according to the hunting behavior of the gray Wolf, and finally the optimal hyperparameter combination is gradually approached by iterative updating the location of the gray Wolf, so as to realize the integration of IGWO optimized SVM. IGWO significantly improves the search efficiency by controlling the parameter random adjustment strategy, at the same time, by minimizing the MSE and RMSE as the optimization targets, the model parameters are dynamically adjusted to improve the prediction accuracy of the model. The study uses IGWO to optimize the SVM, resulting in IGWO-SVM. The implementation steps of this algorithm are shown in Figure 2.

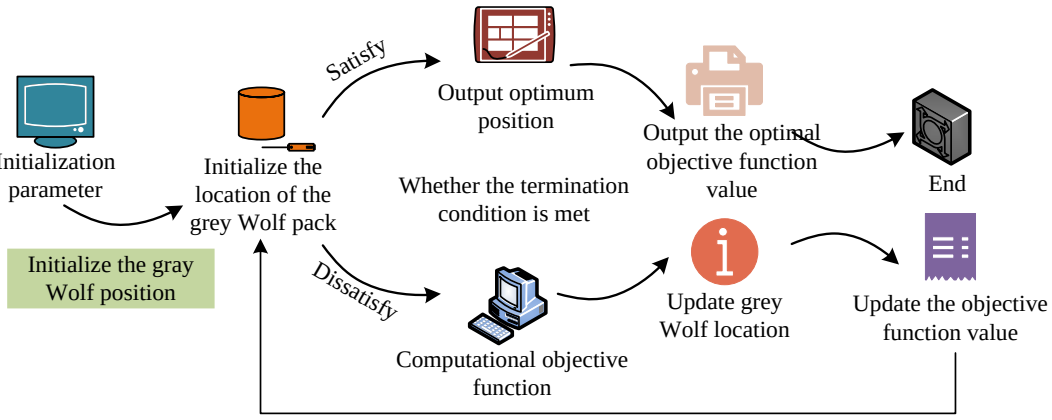


Figure 1: Working process of IGWO.

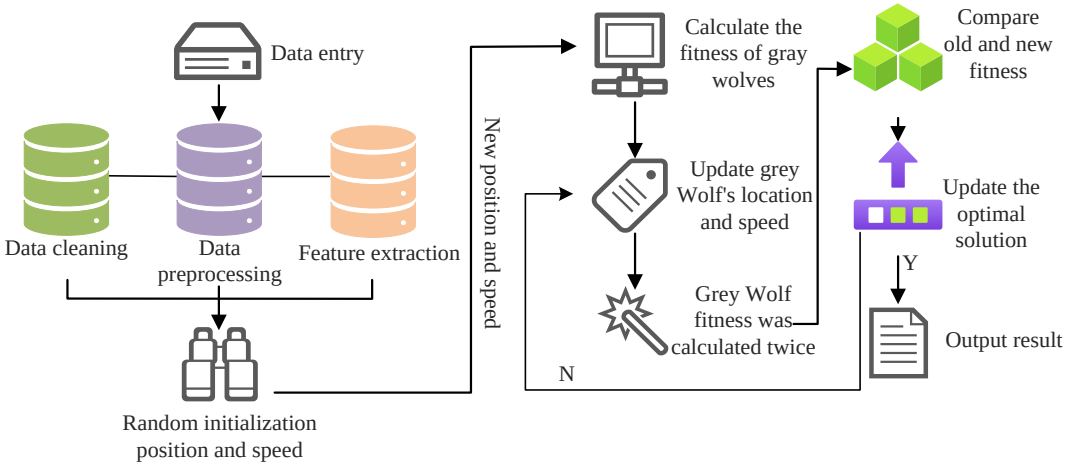


Figure 2: Implementation steps of IGWO-SVM.

As shown in Figure 2, the initial stage of IGWO-SVM involves preprocessing the input data and randomly initializing the data based on the position and velocity of the gray wolves. The fitness of the wolves is calculated using the new positions and velocities, and after a second calculation, the updated fitness is compared with the previous fitness to determine the optimal strategy that satisfies the termination condition. The Equation for calculating the individual gray wolf's tracking of the prey's position is shown in Equation (2).

$$\begin{cases} \vec{D}_\alpha = \left| \vec{C}_1 . \vec{X}_\alpha - \vec{X} \right| \\ \vec{D}_\beta = \left| \vec{C}_2 . \vec{X}_\beta - \vec{X} \right| \\ \vec{D}_\delta = \left| \vec{C}_3 . \vec{X}_\delta - \vec{X} \right| \end{cases} \tag{2}$$

In Equation (2), $\vec{D}_\alpha$ and $\vec{X}_\alpha$ represent the distance between α and other individuals, and α denotes the current position. $\vec{X}_\alpha$ and $\vec{X}_\beta$ represent the distance between β and other individuals, and β is the current position. $\vec{D}_\delta$ and $\vec{X}_\delta$ represent the distance between δ and other individuals, and δ is the current position. $\vec{C}_1$, $\vec{C}_2$, and $\vec{C}_3$ are random vectors. Individual tracking of prey location by grey wolves can shorten the optimization time of SVM parameters through a collaborative search mechanism and capture the non-linear relationship between carbon emissions and driving factors. The new position of the gray wolf individual is determined by the position-weighted average of α, β and δ, as shown in Equation (3).

$$\vec{X}(t+1) = \frac{\vec{X}_\alpha - A_1 \cdot \vec{D}_\alpha + \vec{X}_\beta - A_2 \cdot \vec{D}_\beta + \vec{X}_\delta - A_3 \cdot \vec{D}_\delta}{3} \tag{3}$$

In Equation (3), A_1, A_2 and A_3 coefficients representing the control step size. The convergence factors in IGWO-SVM are used to determine whether the algorithm reaches the optimal solution in the iteration process, and the information feedback mechanism feeds back information into the parameter update strategy by evaluating the quality of the current solution, and then guides the next search direction. IGWO-SVM dynamically adjusts the search step size and search range for gray wolf individuals by introducing adaptive weights and following convergence factors and information feedback mechanisms, and terminates the iteration early when the fitness value dose not significantly change in multiple successive iterations. Moreover, integrating IGWO can compensate for the lack of convergence performance and the need for a large amount of training data in SVM. The proposed IGWO-SVM carbon emission prediction model is shown in Figure 3.

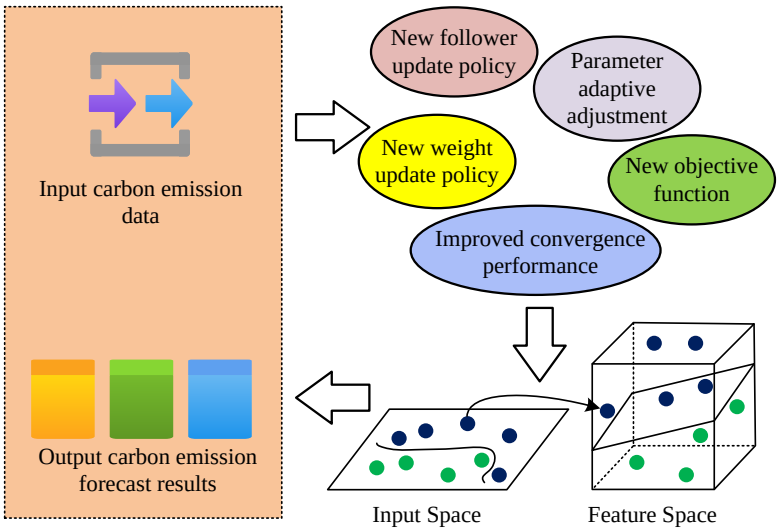


Figure 3: Working diagram of IGWO-SVM carbon emission prediction modeling.

As shown in Figure 3, the model for predicting carbon emissions first requires determining the target range and gathering relevant parameters. After initializing the parameters, the target function needed for the experiment is obtained. The new target function is then derived through adaptive parameter adjustments and weight updating strategies. This updated target function is used as the input vector, and SVM is employed to find a separating hyperplane in the feature space to classify the data. After data integration, the final carbon emission prediction results are obtained. The distance calculation Equation from any point to the hyperplane in the sample space is the core mathematical tool of SVM, which can allow the sample space to be divided into different regions through the hyperplane to capture the non-linear relationship between carbon emissions and driving factors. The distance from any point in the sample space to the hyperplane is calculated using the Equation in Equation (4).

$$r = \frac{\left| w^T + b \right|}{\left\| w \right\|} \tag{4}$$

In Equation (4), w^T and b represent the normal vector and the displacement term, respectively. w represents the normal vector of the hyperplane. The Equation for calculating the distance between the data points and the decision boundary is shown in Equation (5).

$$d_i = \frac{w^T x_i + b}{\left\| w \right\|} \tag{5}$$

In Equation (5), $\left\| w \right\|$ and x_i represent the Euclidean norm of the normal vector w and the data points, respectively. The Equation for calculating the individual gray wolf is shown in Equation (6).

$$\begin{cases} \vec{X}_1 = \vec{X}_\alpha - A_1 \cdot \vec{D}_\alpha \\ \vec{X}_2 = \vec{X}_\beta - A_2 \cdot \vec{D}_\beta \\ \vec{X}_3 = \vec{X}_\delta - A_3 \cdot \vec{D}_\delta \end{cases} \tag{6}$$

In Equation (6), $\vec{X}_1$, $\vec{X}_2$, and $\vec{X}_2$ represent different gray wolf individuals, determining the step size and direction in which the gray wolf individuals move towards α, β, and δ. By applying the IGWO-SVM model to carbon emission prediction, the model parameters are adaptively adjusted according to the different characteristics of carbon emissions, allowing the initial feature parameters to be more evenly distributed in the hyperplane solution space, thus improving the model's prediction accuracy and generalization ability. Next, the Gaussian kernel is preferentially selected as the kernel function, and the optimal value of the generalization and fitting ability of the equilibrium model is found by IGWO. The adjustment of the kernel function parameters was searched in a smaller range by IGWO to avoid the model being too sensitive to local data. This provides reliable data support for formulating and implementing carbon emission policies.

3.2 Optimization of the carbon emission prediction model combined with STIRPAT

Although applying IGWO to optimize SVM improves its sensitivity to parameters, the fusion model may get stuck in local optima in complex problems, which affects the model's prediction accuracy. Therefore, this study proposes further introducing STIRPAT to improve the prediction results based on the limitations of IGWO-SVM. By introducing randomness and regression methods, the research can flexibly handle the nonlinear relationship between carbon emissions and population, affluence and technology level, while the traditional IPAT model can only describe the linear relationship. And the STIRPAT model supports the addition of other variables, which can more comprehensively analyze the drivers of carbon emissions in industrial zones. Other models are usually limited to linear assumptions or single variables. Its components are shown in Figure 4.

As shown in Figure 4, the pressure in STIRPAT determines the environmental carrying capacity and the impact on the ecosystem, including factors such as climate change and environmental pollution. The four components—trend, pressure, impact, and response—together describe the comprehensive effect of human factors on the environment. By simulating and predicting the influencing factors, the model provides an important basis for ecosystem management [20]. The core calculation Equation of the STIRPAT model is shown in Equation (7).

$$I = aP^b A^c T^d e \tag{7}$$

In Equation (7), I represents environmental impacts, such as carbon emissions. a and P represent the model's elasticity coefficient and population size, while A and T represent affluence and technological level. b is the elasticity coefficient for population size, and c represents the elasticity coefficient for affluence. d and e represent the elasticity coefficient for technological level and the error term, respectively. By adjusting the exponents in the model and adding new variables, the model can be made suitable for different research needs, offering excellent flexibility and scalability. The mechanism of incorporating STIRPAT to improve the carbon emission prediction model is shown in Figure 5.

As shown in Figure 5, the IGWO-SVM carbon emission prediction model defines and normalizes the feature space vectors output by SVM, and after variable transformation, the STIRPAT model is used to improve the fusion model through model selection and fitting to determine the independent and dependent variables. The study constructs a regression model using linear regression methods, and finally evaluates the model's goodness of fit and predictive ability through residual analysis tools, thus obtaining the prediction results of the model. The error Equation is calculated as shown in Equation (8).

$$\begin{cases} Error(s|X,y) = (Xs-y)^{T(Xs-y)} \\ X = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1n} \\ X_{21} & X_{22} & \cdots & X_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ X_{m1} & X_{m2} & \cdots & X_{mn} \end{bmatrix} \\ s = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} \end{cases} \tag{8}$$

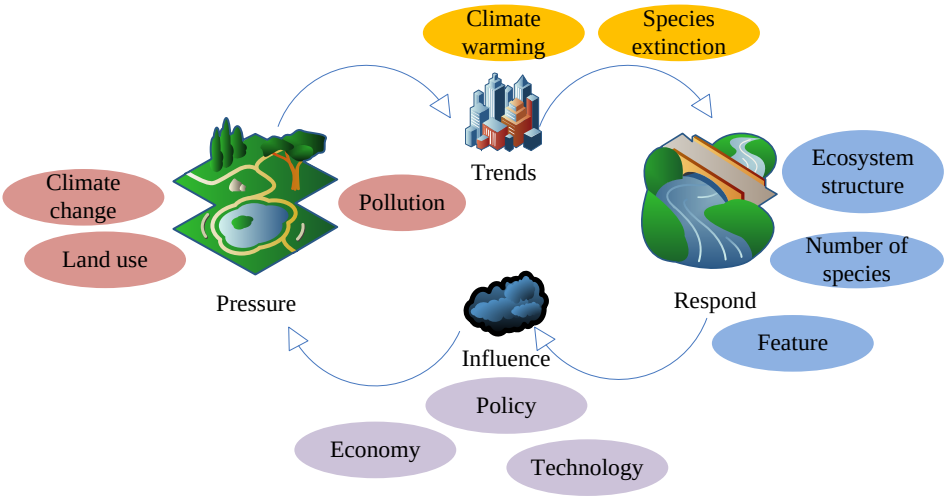


Figure 4: Components of STIRPAT.

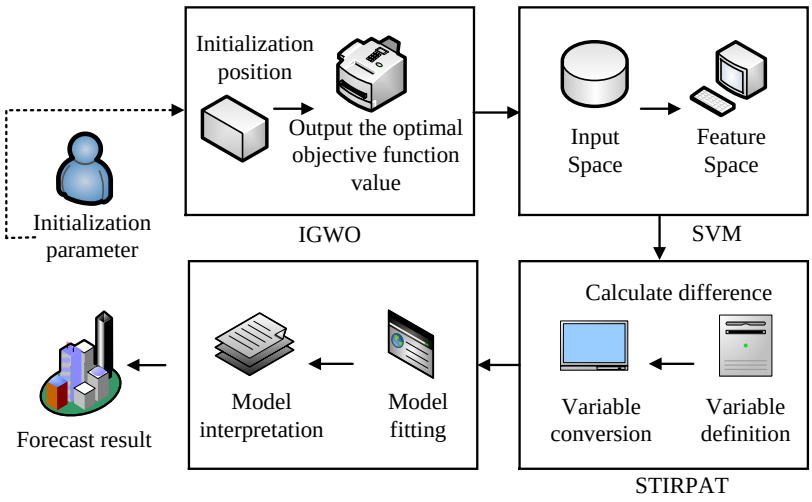


Figure 5: Workflow of the carbon emissions forecasting model introduced into STIRPAT.

In Equation (8), X and y represent the sample input matrix of $m \times n$ and the column vector of $m \times 1$, respectively. s represents the column vector of $n \times 1$. m and n are the number of samples and the number of features for each sample, respectively. The calculation Equation for the objective function is shown in Equation (9).

$$R = \min \frac{1}{2} w^T w + C \sum_{i=1}^{n} \gamma_i \tag{9}$$

In Equation (9), C and γ represent the penalty factor and slack variables, respectively. w weight the impact of different characteristics on carbon emissions in the carbon emission prediction model. By optimizing the normal vector, the relationship between carbon emissions and drivers can be fitted more accurately, and adjusting the penalty factor can avoid model overfitting and ensure the reliability of the prediction results. Minimizing the normal vector and penalty factor optimize the feature weights and reduce the computational complexity of the model. This model not only reduces the difficulty of data acquisition but also enhances the model's flexibility through regression analysis, effectively compensating for the limitations of traditional prediction models influenced by data quality. The study selected the filtering method as a feature selection technique, evaluating feature importance based on statistical indicators such as variance and screening it independently of the model. The carbon emission prediction model combining IGWO-SVM and STIRPAT is shown in Figure 6.

From Figure 6, it can be observed that after inputting carbon emission and influencing factor data into the model, the first step is to initialize the data through IGWO-SVM and output the optimal objective function value. This function value forms the optimal hyperplane in the feature space after maximizing the classification margin. The second part, the STIRPAT module, mainly standardizes the data, determines the distribution of parameters to be optimized, and optimizes the parameters using a random search method. Finally, after result verification, the predicted value of carbon emissions is output. The calculation Equation for the inner product kernel function is shown in Equation (10).

$$K(c_i, c_j) = \exp(-y \left\| c_i - c_j \right\|^2) \tag{10}$$

In Equation (10), c_i and c_j represent the input factors and the expected values, respectively. The calculation Equation for the nonlinear function is shown in Equation (11).

$$f(x) = \sum_{sv} (\alpha_i - \alpha_i^*) K(c_i g c_j) + v \tag{11}$$

In Equation (11), α_i and α_i^* represent the support vector parameters, while v and $K(c_i g c_j)$ represent the constant term and the inner product kernel function, respectively. The calculation Equation for the function margin is shown in Equation (12).

$$\hat{\gamma} = a(w^T x + v) \tag{12}$$

In Equation (12), $w^T x + b$ represents the distance from point T to the hyperplane. The integration of IGWO-SVM and STIRPAT mainly builds the STIRPAT model using historical data, outputs the quantitative relationship between carbon emission and each factor as the input feature of SVM, and then trains the SVM model optimized by IGWO. The optimization of the fusion model by STIRPAT not only enhances the model's flexibility and comprehensiveness, but also compensates for the lack of timeliness in single prediction algorithms. This makes the prediction results more accurate, greatly improving the timeliness and transparency of carbon emission data.

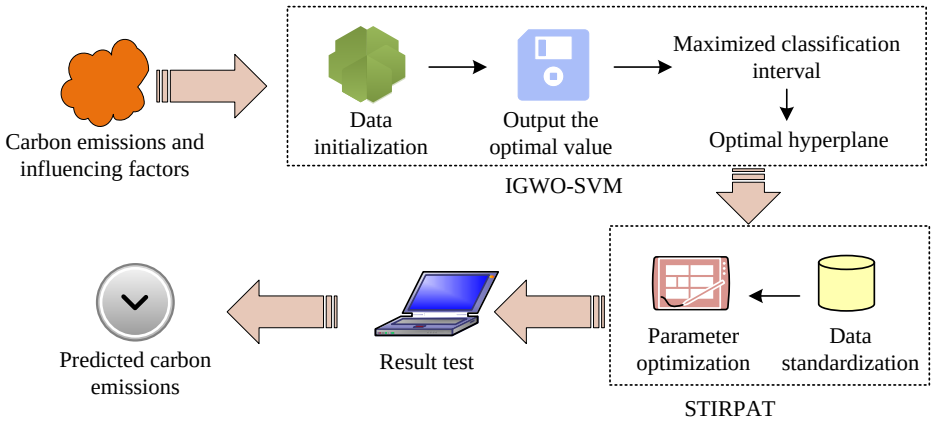


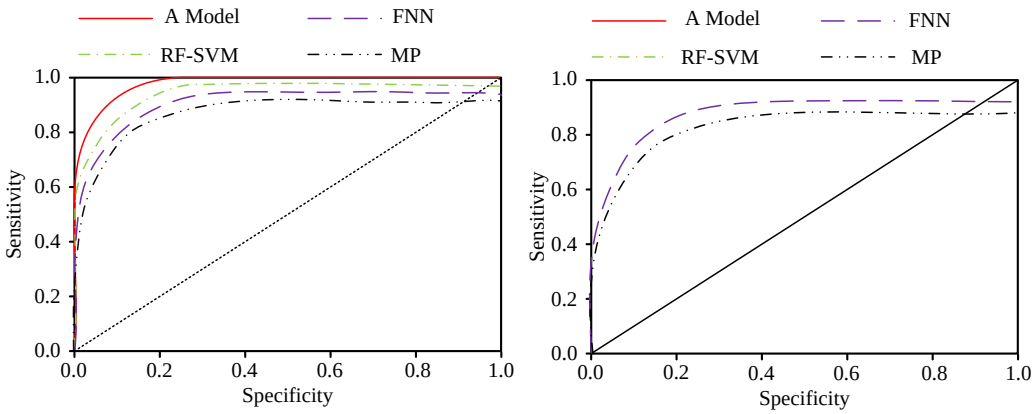
Figure 6: Final carbon emissions forecasting model workflow diagram.

Design the rolling verification mechanism for time-dependent nature of carbon emission data, and conduct the cross-validation of the model evaluation by simulating the actual prediction scenario. The penalty factor of SVM is dynamically adjusted within the range of 0.1~100 through IGWO to balance model complexity and training error. The optimization interval of Gaussian kernel parameters and the value of ε in the regression task are set to 0.001-1001000 and 0.01-0.1 respectively, to avoid overfitting of the model to industrial data noise. The population elasticity coefficient and the technical elasticity coefficient of the STIRPAT model are constrained between 0.8 and 1.5 and 0.3~0.9, respectively. To enhance the adaptability of the model to different policy strengths, the study treats the policy drivers by normalization and maps to an interval of 0 to 1.Set the IGWO algorithm to stop the calculation when the fitness value change rate is below 110-4 after 20 consecutive iterations.

4 Performance verification of the industrial zone carbon emission prediction based on the fusion model

4.1 Comparative performance analysis of the fusion model

In order to validate the performance of the STIRPAT combined with IGWO-SVM hybrid model, the study named this fusion model as Model A, and selected Feedforward Neural Network (FNN), Random Forest and Support Vector Machine (RF-SVM), and Multilayer Perceptron (MP) as comparison prediction models. FNN, RF-SVM and MP all have strong non-linear modeling capabilities, and can compare the performance with the integrated model of IGWO-SVM and STIRPAT in complex data scenarios, and choosing different types of models for comparison can verify the superiority of the integrated model from multiple perspectives. The study selected the programming language and TensorFlow 2.5 to implement data preprocessing and model implementation, using AMD Ryzen 7 and NVIDIA RTX 3060 for the hardware configuration, with a memory size of 32GB. The four models had 400 iterations, with a population size of 35. The 400 iterations ensure that the algorithm reaches the convergence state when optimizing the model parameters and effectively controls the computing resource consumption. Setting the overall size set to 35 ensures that the algorithm maintains sufficient population diversity within the search space, reducing the likelihood of a local optimal solution. The carbon emission prediction results for different provinces and years were compared and analyzed. The experimental datasets included the China Emission Accounts and Datasets (CEADs) dataset and the Multi-resolution Emission Inventory for China (MEIC) dataset. The CEADs dataset served as the training set, containing carbon accounting inventories at multiple scales, and was used to verify the performance of the fusion model. The MEIC dataset, as the testing set, covered carbon emission data from various industry sectors and was used to test the accuracy of the model in real-world applications. The CEAD and MEIC datasets adopt the proportional partition of 80% training set and 20% test set, combined with cross-validation to ensure that the model has good generalization ability while fully learning the data features. To ensure fairness in the experiment, the same parameter preprocessing was applied to all four models. CEADs data set in the process of pretreatment, the first 3 σ criteria to identify and eliminate the outlier indicators such as carbon intensity, through time series interpolation processing local missing, then the carbon emissions data and industrial economic indicators features cross and build multidimensional driver, normalized after the carbon emissions sequence decomposition into high frequency and low frequency mode. In the preprocessing of MEIC data set, spatial weighting method was used to aggregate the grid data to regional scales, divide carbon emission sources according to industrial sub-industries, and extract industry characteristic parameters as input variables for the model. Finally, the carbon emission trend differences of the data set was compared by cross-verification, and the systematic deviation was corrected by Bayesian network. And the ROC curves for the four models in the CEADs and MEIC datasets are shown in Figure 7.



(a) ROC curves of the four models in the CEADs dataset

(b) ROC curves of the four models in the MEIC dataset

Figure 7: ROC curves of the four models in CEADs dataset and MEIC dataset.

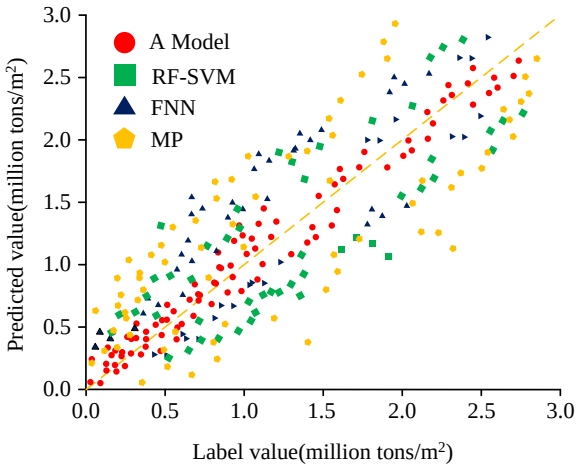
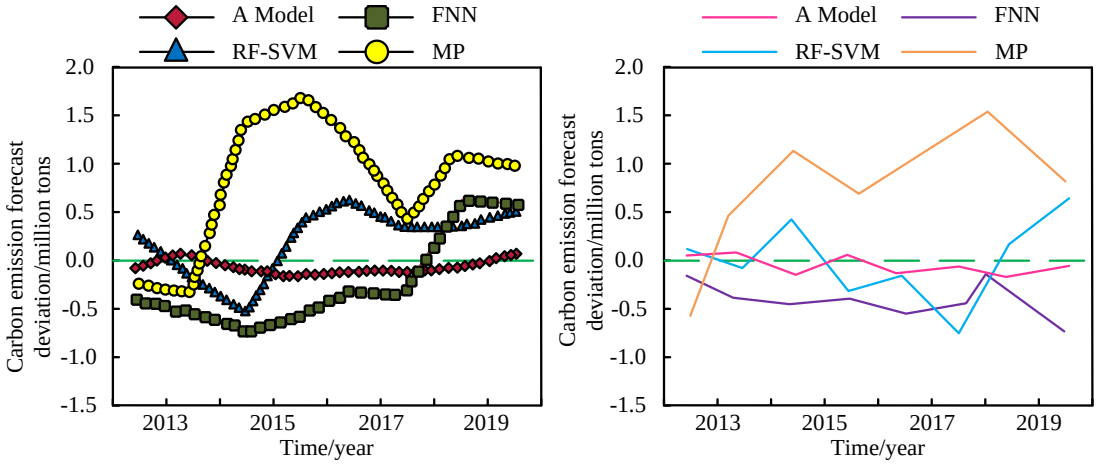


Figure 8: Prediction results of the four models in the CEADs dataset.

As shown in Figure 7(a), in the CEADs dataset, the sensitivity curves of all four models increased as specificity increased. The ROC curve of Model A was closest to the upper left corner, and the Area Under Curve (AUC) value was the largest, reaching 0.95. The AUC value of the RF-SVM model was relatively lower than that of Model A, at 0.91, while the AUC values of the FNN and MP models were noticeably different from that of Model A, at 0.86 and 0.82, respectively. As shown in Figure 7(b), Model A performed well in the MEIC dataset, with an AUC value of 0.94, while the AUC values of the other three comparison models were all lower than that of Model A. The AUC values of the RF-SVM and FNN models were 0.89 and 0.84, respectively, while the MP model had the lowest AUC value, at only 0.80. In summary, from Figure 7, it can be seen that in different datasets, the ROC curve of the proposed Model A was the closest to the left side, and the area under the curve was the largest, demonstrating excellent classification performance compared to traditional algorithms. The CEADs dataset contained rich data samples, and by transforming the samples, the number of training samples was increased, reducing the risk of model overfitting. Therefore, the experiment used this as input data, and the carbon emission prediction values output by the four models were analyzed against the dataset labels, as shown in Figure 8.

As shown in Figure 8, Model A exhibited relatively good prediction results in the CEADs dataset, when the label value is 0~1.0million tons / m2, the predicted value of carbon emission is close to the label value is relatively dense. RF-SVM prediction results were more scattered compared to Model A, and when the label value was 1.5 million tons/m2, its carbon emission prediction reached 1.92 million tons/m², deviating significantly from the label value. At the same time, both FNN and MP performed worse than Model A. When the carbon emission label value was 1.5 million tons/m2, FNN's carbon emission prediction deviated to 2.06 million tons/m², which was 0.55 million tons/m² higher than Model A's prediction. Based on the data results, it can be observed that Model A exhibited significant prediction accuracy among the four algorithms and maintained good performance even when the label values varied. For a comprehensive evaluation of the overall performance of the fusion model, the study also analyzed the carbon emission prediction deviations of the four models in the training and testing datasets, as shown in Figure 9.



(a) Carbon emission prediction error values of the four models in the CEADs dataset

(b) Carbon emission prediction error values of the four models in the MEIC dataset

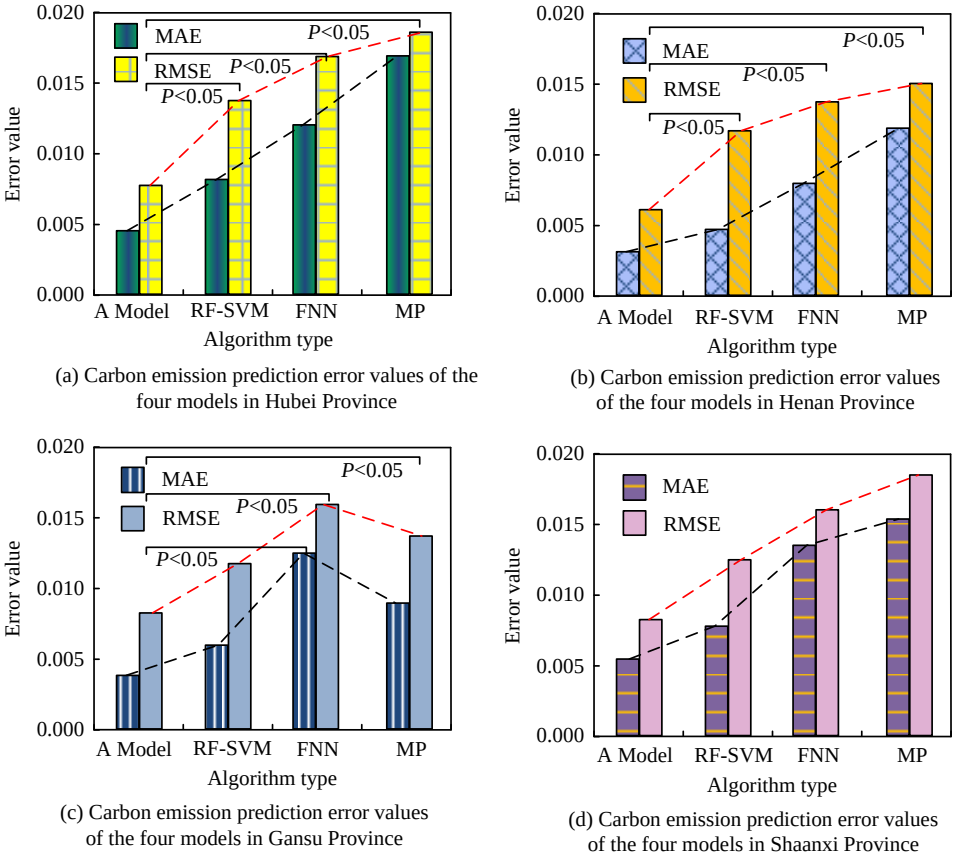
Figure 9: Carbon emission prediction deviation of four models in training set and test set.

As shown in Figure 9(a), in the CEADs dataset, Model A's carbon emission predictions were the closest to the actual values, resulting in the smallest data deviation. When the year was 2014, Model A's prediction deviation was only -0.003, while MP's prediction deviation reached 1.46 in 2016. The deviation curves of the comparison algorithms also fluctuated more significantly, and their overall performance was inferior to that of Model A. According to Figure 9(b), Model A achieved the minimum carbon emission prediction deviation of -0.009 in 2016. The corresponding prediction deviations for RF-SVM and FNN were -0.27 and -0.48, while MP's prediction deviation reached 0.78. Based on the data results, it can be concluded that the proposed Model A demonstrated excellent prediction performance and accuracy when compared to the other algorithms.

4.2 Verification of the prediction effect of the fusion model in practical application scenarios

To further validate the performance of the proposed fusion model in real-world carbon emission prediction scenarios, the study selected carbon emission data from four provinces—Hubei, Henan, Gansu, and Shaanxi—for the experiment, and assigned different sample numbers to them. The sample number range of Hubei Province is 001-100, and the data covers major carbon emission industries such as energy production and manufacturing. The sample number range of Henan Province is 101-200, which includes the carbon emission data of some emerging industries. The sample number of Gansu Province is 201-300, and the sample data is mainly energy production and mineral resources development. The sample number of Shaanxi Province is 301-400. The data includes carbon emission data mainly from high-tech industries and traditional energy industries. The difference in the accuracy of carbon emission prediction between different provinces and cities mainly stems from the combined impact of data quality, regional policy and model adaptability. The carbon emission prediction accuracy of the four models in various provinces is shown in Figure 10.

As shown in Figure 10, the four models demonstrated varying levels of accuracy in carbon emission predictions across different provinces and cities, the root mean square error (Root Mean Squared Error, RMSE) is statistically significant. In Hubei, Model A and RF-SVM had Mean Absolute Errors (MAE) of 0.0048 and 0.0076, respectively, while FNN and MP had MAEs of 0.0121 and 0.0173, respectively. Among the four models, MP had the highest Root Mean Square Error (RMSE) at 0.182, while Model A's RMSE was only 0.0047. In Henan and Gansu, Model A's MAEs were 0.0032 and 0.0041, respectively, with the MAE and RMSE of the other three comparison models being higher than those of Model A. In Shaanxi, Model A had the smallest MAE and RMSE at 0.0052 and 0.0073, respectively, showing the largest difference compared to MP's errors. These results indicated that Model A exhibited the smallest prediction errors among the four algorithms, making it more accurate in predicting carbon emissions than traditional methods. The adaptability of the four models in Hubei and Shaanxi is shown in Figure 11.



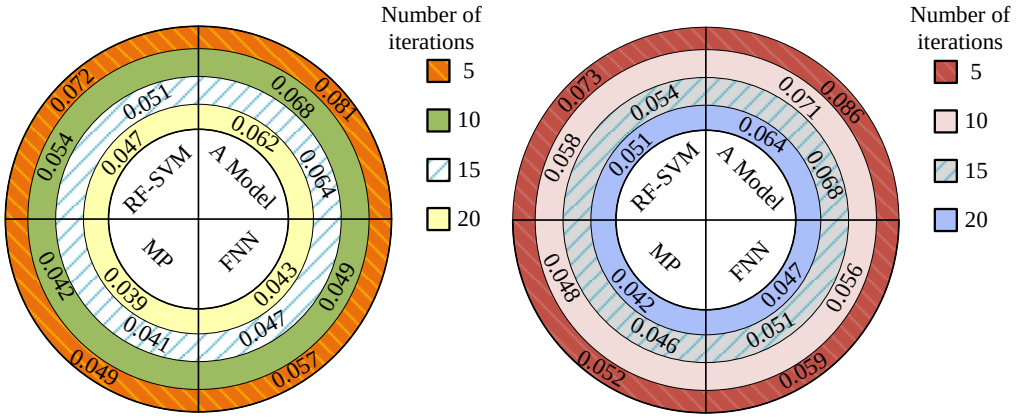
(a) Carbon emission prediction error values of the four models in Hubei Province

(b) Carbon emission prediction error values of the four models in Henan Province

(c) Carbon emission prediction error values of the four models in Gansu Province

(d) Carbon emission prediction error values of the four models in Shaanxi Province

Figure 10: Comparison of carbon emission forecasting accuracy in different provinces and municipalities.



(a) The fitness of the four models in Hubei Province (b) The fitness of the four models in Shaanxi Province

Figure 11: The fitness of four models in different provinces and cities.

As shown in Figure 11(a), in Hubei, when the number of iterations was 5, the fitness values of Model A and RF-SNN were 0.081 and 0.072, respectively. Compared to Model A, the fitness of FNN decreased by 0.024. MP had the lowest fitness, with a value of only 0.049. When the number of iterations was 20, Model A's fitness was 0.023 higher than that of MP, and the fitness values of the other two comparison algorithms were lower than that of Model A. In Figure 11(b), in Shaanxi, the fitness of Model A was higher than that of the comparison models. When the number of iterations was 20, MP's fitness was only 0.042, while Model A's fitness reached 0.064. Overall, Model A demonstrated the best fitness across different data samples, showing an improvement in performance compared to traditional algorithms. The prediction accuracy of Model A and comparison models on the CEADs dataset, MEIC dataset, and sample data from 2018 is shown in Table 2.

Table 2: Prediction accuracy of four models in CEADs dataset and MEIC dataset.

Data set	Model	MSE	RMSE	MAE	MAPE	Training time /s
Sample data of Hubei Province	A Model	0.0075	0.0866	0.0086	0.0186	64.39
	RF-SVM	0.0126*	0.1122*	0.0127*	0.0249*	116.32
	FNN	0.0153*	0.1237*	0.0164*	0.0275*	122.08
	MP	0.0169*	0.1300*	0.0178*	0.0304*	129.61
Sample data of Henan Province	A Model	0.0062	0.0787	0.0092	0.0191	51.37
	RF-SVM	0.0134*	0.1158*	0.0143*	0.0284*	114.58
	FNN	0.0159*	0.1261*	0.0172*	0.0291*	119.34
	MP	0.0156*	0.1249*	0.0186*	0.0327*	125.79
Gansu province sample data	A Model	0.0079	0.0889	0.0081	0.0142	59.05
	RF-SVM	0.0143*	0.1196*	0.0119*	0.0266*	114.87
	FNN	0.0152*	0.1233*	0.0167*	0.0293*	126.43
	MP	0.0161*	0.1269*	0.0206*	0.0375*	131.82
Shaanxi province sample data	A Model	0.0074	0.0860	0.0079	0.0148	62.39
	RF-SVM	0.0141*	0.1187*	0.0117*	0.0264*	118.74
	FNN	0.0157*	0.1253*	0.0165*	0.0296*	123.49
	MP	0.0169*	0.1300*	0.0078*	0.0341*	126.13

Note: * represents P <0.05, meaning that the results are statistically significant.

The data results in Table 2 showed that Model A demonstrated excellent prediction performance across all datasets. In the Hubei dataset, the RMSE of Model A was the smallest among the four models, with a value of 0.0075, while the RMSE of FNN and MP were higher than Model A by 0.0101 and 0.0197, respectively. In the Henan dataset, the MAE of Model A was 0.0092, and the errors of the other three comparison algorithms were all higher than that of Model A, with the MP's Mean Absolute Percentage Error (MAPE) reaching 0.0327. In the Gansu dataset, Model A also exhibited good accuracy, with an MAE of only 0.0081, while the MAEs of RF-SVM and FNN were 0.0119 and 0.0167, respectively, both higher than that of Model A. From the perspective of training time in different datasets, the training time of the proposed model was less than 100s, while the training time of the comparison model reached 129.61s, and according to the experimental results, the obtained error data were statistically significant. According to the experimental results, the obtained study data are statistically significant. Overall, the proposed fusion model performed excellently in various aspects, providing precise prediction performance on different sample data. It improved the applicability of traditional algorithms in large-scale big data applications. Moreover, applying this hybrid model to carbon emission prediction has provided a new approach for scientifically and rationally formulating emission reduction policies.

5 Discussion

As a key part of promoting green low-carbon transformation and realizing the goal of "dual carbon", carbon emission prediction can provide scientific basis for relevant institutions to formulate emission reduction policies by predicting the future carbon emission trend and peak time. According to the ROC curves of the proposed prediction model in different datasets, it can be found that the proposed model has excellent classification performance compared with the traditional comparison algorithm, and can still maintain good prediction performance under the condition of label value changes. Moreover, the predicted carbon emission value of the proposed model is the closest to the actual value, and the data deviation generated is the smallest. In the actual application scenario, the fitness of the proposed model in different data samples is the best. The fusion model dynamically screens the subset of features by combining the IGWO algorithm to ensure that the model can still run efficiently on large datasets, and the normalization in data preprocessing enables the model to adapt to a larger scale of data input.

The RF-SVM model has high parameter sensitivity when processing high-dimensional data, which is prone to overfitting risk. It is difficult for MP model to deal with the complex interaction effect between carbon emission and multiple factors, and it lacks adaptability to external variables, resulting in a large prediction error. The dynamic adjustment of parameters by IGWO in the proposed model effectively balances the bias and variance, making the model maintain a low error on different datasets. Moreover, STIRPAT has enhanced the model's ability to explain complex systems by introducing macro-driving factors, thus reducing the prediction error caused by sample bias. In terms of bias-variance tradeoff, the dynamic feature selection mechanism of the proposed model can automatically identify key drivers and reduce the prediction bias caused by feature redundancy or missing. The implementation of the research model in the real-world industrial environment or policy making background can be implemented through model verification optimization and policy coordination, so as to provide a quantitative decision tool for "two-carbon targets".

Although the IGWO-SVM-STIRPAT model shows excellent performance in the prediction of carbon emission in industrial zones, the noise data may interfere with the feature selection process of the IGWO algorithm and lead to the inaccurate selected feature subset, which affects the explanatory power and prediction effect of the fusion model. The model prediction should consider the short-term emission reduction targets and emphasize the balance between economic and social development in the industrial areas, and eliminate the variables that may cause

bias to ensure the fairness of the prediction results. The fusion model will meet future challenges posed by the challenges of future carbon policy changes and new industrial data through carbon policy scenario simulation and adaptive optimization of the model.

6 Conclusion

To address the issues of low accuracy, limitations, and uncertainty in traditional carbon emission prediction models, this study proposed the IGWO-SVM, a novel predictive model combined with STIRPAT. This model achieves accurate carbon emission prediction and enhances the overall applicability and robustness of the model compared to traditional single algorithms. Experimental data results showed that in the CEADs dataset, as specificity increased, the sensitivity curve of the four models also rose. The ROC curve of Model A was closest to the top-left corner, with the largest AUC under the curve, reaching 0.95. The AUC value of RF-SVM was slightly lower than that of Model A, at 0.91, while the AUC values of FNN and MP showed significant differences compared to Model A, at 0.86 and 0.82, respectively. The prediction results of RF-SVM were more dispersed than those of Model A. When the label value was 1.5 million tons/m2, the carbon emission prediction of RF-SVM reached 1.92 million tons/m², deviating significantly from the label value. Similarly, both the FNN and MP had inferior prediction performance compared to Model A. When the carbon emission label value was 1.5 million tons/m2, the prediction of FNN deviated to 2.06 million tons/m², which was 0.55 million tons/m² higher than the prediction of Model A. Overall, the proposed fusion model demonstrated excellent accuracy and prediction performance in carbon emission prediction, with the best adaptability across different data samples. Compared to traditional algorithms, its performance showed notable improvements. The proposed model provides policy makers with scientific and interpretable carbon emission prediction tools through the elastic coefficient analysis of STIRPAT, feature selection optimized by IGWO and high-precision prediction of SVM. However, its classification performance still has room for improvement compared to the other three models, so further optimization and refinement of this fusion algorithm are necessary in future work.

References

[1] Yuting Luo. Artificial algae-optimized support vector machine for sentiment analysis of network public opinion. Informatica, 48(21):167-175, 2024. https://doi.org/10.31449/inf.v48i21.6620

[2] A. J. Gnanamalar, R. Bhavani, A. S. Arulini, and M. S. Veerraju. CNN-SVM based fault detection, classification and location of multi-terminal VSC-HVDC system. Journal of Electrical Engineering & Technology, 18(4):3335-3347, 2023. https://doi.org/10.1007/s42835-023-01391-5

[3] Hassane Ezziane, Hamza Houassine, Samir Moulahoum, and Moustafa Sahnoune Chaouche. A novel method to identification type, location, and extent of transformer winding faults based on FRA and SMOTE-SVM. Russian Journal of Nondestructive Testing, 58(5):391-404, 2022. https://doi.org/10.1134/S1061830922050047

[4] Ahmad Al Badawi, Ling Chen, and Saru Vig. Fast homomorphic SVM inference on encrypted data. Neural Computing and Applications, 34(18):15555-15573, 2022. https://doi.org/10.1007/s00521-022-07202-8

[5] Fenfen Cheng, and Junfeng Yao. The effect of the digital economy on carbon emissions in China's construction industry: Evidence from spatial econometric analysis. Environmental Science and Pollution Research, 31(54):63085-63097. https://doi.org/10.1007/s11356-024-35426-9

[6] P. S. Neethu, R. Suguna, and S. Palanivel Rajan. Performance evaluation of SVM-based hand gesture detection and recognition system using distance transform on different data sets for autonomous vehicle moving applications. Circuit World, 48(2):204-214, 2022. https://doi.org/10.1108/CW-06-2020-0106

[7] E. Gangadevi, R. Shoba Rani, Rajesh Kumar Dhanaraj, and Anand Nayyar. Spot-out fruit fly algorithm with simulated annealing optimized SVM for detecting tomato plant diseases. Neural Computing & Applications, 36(80):4349-4375, 2024. https://doi.org/10.1007/s00521-023-09295-1

[8] Lang Yu, Xin Ma, and Shengjie Li. A fast conjugate functional gain sequential minimal optimization training algorithm for LS-SVM model. Neural Computing & Applications, 35(8):6095-6113, 2023. https://doi.org/10.1007/s00521-022-07875-1

[9] Olatunji A. Shobande, and Simplice A. Asongu. Searching for sustainable footprints: Does ICT increase CO2 emissions. Environmental Modeling & Assessment, 28(1):133-143, 2023. https://doi.org/10.1016/j.eneco.2019.04.014

[10] Ehsan Rasoulinezhad, and Farhad Taghizadeh-Hesary. Role of green finance in improving energy efficiency and renewable energy development. Energy Efficiency, 15(2):13-24, 2022. https://doi.org/10.1007/s12053-022-10021-4

[11] Lei Wang, Wei Dai, Liuzhu Zhu, Xuli Wang, Chenxu Yin, Hao Cong, Tiancheng Shi, Xianjun Qi, and Rui Bi, Multi-objective expansion planning of park-level integrated energy system considering the volatility trend of CETP. IET Generation, Transmission & Distribution, 16(6):1225-1243, 2022. https://doi.org/10.1049/gtd2.12363

[12] Omer Faruk Cansiz, Kevser Unsalan, and Fatih Unes. Prediction of CO2 emission in transportation sector by computational intelligence techniques. International Journal of Global Warming, 27(3):271-283, 2022. https://doi.org/10.1504/ijgw.2022.10048914

[13] Yang Tang, Weiwei Liu, Saurabh Singh, Osama Alfarraj Saurabh Singh, and Amr Tolba. Passenger

flow forecast for low carbon urban transport based on bi-level programming model. Journal of Internet Technology, 24(5):1067-1077, 2023. https://doi.org/10.53106/160792642023092405005

[14] Elisavet Amanatidou, Georgios Samiotis, Eleni Trikoilidou, L. Tsikritzis, and Nikolaos Taousanidis. Centennial assessment of greenhouse gases emissions of young and old hydroelectric reservoir in mediterranean mainland. Journal of Environmental Informatics, 41(1):27-36, 2023. https://doi.org/10.3808/jei.202300485

[15] Mengxing Gao, ChenGuang Liu, Zigui Liu, and Xi Chen. A tabu search algorithm for the unrelated parallel machine scheduling problem with varied carbon emission constraints in different time intervals. The Journal of the Operational Research Society, 75(6):1111-1125, 2024. https://doi.org/10.1080/01605682.2023.2233555

[16] Xuefei Xie, and Lin Chen. Dynamic programming-based energy optimization for pure electric commercial vehicles with predictive cruise control. Informatica, 48(22):129-145, 2024. https://doi.org/10.31449/inf.v48i22.6922

[17] Jieqiong Zhang. Optimizing the analysis of energy plants and high-power applications utilizing the energy guard ensemble selector (EGES). Informatica, 49(10):113-126, 2025. https://doi.org/10.31449/inf.v49i10.7264

[18] Jinghan Zhou, Yongxiu He, Yuan Lyu, Kehui Wang, Yiran Che, and Xiaoqing Wang. Long-term electricity forecasting for the industrial sector in western China under the carbon peaking and carbon neutral targets. Energy for Sustainable Development, 73(3):174-187, 2023. https://doi.org/10.1016/j.esd.2023.02.003

[19] Rajkumar Palaniappan. Comparative analysis of support vector machine, random forest and k-nearest neighbor classifiers for predicting remaining usage life of roller bearings. Informatica, 48(7):39-52, 2024. https://doi.org/10.31449/inf.v48i7.5726

[20] Qiankun Jiang, and Haiyan Wang. Risk assessment method for new energy vehicle supply chain based on hierarchical holographic model and matter element extension model. Informatica, 49(7):49-62, 2025. https://doi.org/10.31449/inf.v49i7.6953

Deep Q-Network-Based Reinforcement Learning for Medium and Short-Term Reserve Capacity Classification in Power Systems

Yi Wang[1], Gang Wu[2], Chuan He[1], Ruiguang Ma[2], Jing Xiang[1], Tiannan Ma[2], Feng Liu[3*]
[1]State Grid Sichuan Electric Power Company, Chengdu 610000, Sichuan, China
[2]State Grid Sichuan Electric Power Company, Chengdu 610000, Sichuan, China
[3]Beijing TsIntergy Technology Co., Ltd, Beijing 100084, Beijing, China
E-mail: YananZhang46@outlook.com

Keywords: classification performance, deep Q-Network (DQN), deep reinforcement learning (DRL), power system regulation, reserve capacity prediction

Received: May 19, 2025

Modern power systems encounter significant challenges in maintaining reliability and operational balance due to the intermittent nature of renewable energy sources and variable demand. Accurate prediction and optimization of reserve capacity are essential to ensure grid stability, especially within medium and short-term regulatory timeframes. Traditional reserve estimation methods often lack the adaptability required for dynamic operational data, leading to inefficient reserve allocation. This study introduces a Deep Reinforcement Learning (DRL) framework aimed at enhancing reserve capacity classification and regulation. A Deep Q-Network (DQN)-based agent is developed and trained on a Reserve Capacity Prediction (RCP) dataset consisting of 2000-time steps and ten critical system features. The data underwent preprocessing steps such as categorical encoding, normalization, and environment modeling. The DQN receives a 9-dimensional input vector and uses two hidden ReLU-activated layers (64 and 32 units) to predict reserve capacity classes: Low, Optimal, and High. A reward mechanism and experience replay were applied during training. Experimental results show the DQN outperforms Logistic Regression, Random Forest, and SVM, achieving 90% accuracy, 92% precision, 88% recall, 89.8% F1-score, and 0.86 MCC. This approach shows promise for intelligent and adaptive reserve management in power systems.

Povzetek: DQN-osnovan model globokega okrepitvenega učenja omogoča bolj kvalitetno razvrščanje srednje in kratkoročnih rezervnih kapacitet v elektroenergetskih sistemih, saj presega metode logistične regresije, SVM in naključnega gozda po točnosti, prilagodljivosti in robustnosti.

1 Introduction

Contemporary power systems are quickly evolving as a result of the increasing incorporation of renewable energy sources, changing consumption trends, and the push for smarter grids [1]. These shifts have raised the intricacy of retaining grid stability, particularly in short- to medium-term operational planning. Precise regulation of power system capability and efficient reserve allocation are crucial to ensuring grid reliability, particularly in the face of uncertain conditions like demand fluctuations and renewable variability [2]. The difficulty is to dynamically determine the optimal reserve capacity class—whether low, optimal, or high—to match supply and demand effectively while reducing operational risks.

Numerous conventional methods have been utilized to predict load demand, optimize reserve scheduling, and keep the grid balanced [3]. These include statistical prediction techniques (e.g., ARIMA), rule-based systems, optimization algorithms (e.g., mixed-integer linear programming), and machine learning models like decision trees and support vector machines [4]. These methods have demonstrated some success in historical analysis and deterministic scheduling, but frequently struggle to adapt to real-time, multi-factor settings [5].

Traditional models frequently assume static relationships between input variables and reserve requirements, limiting their adaptability to rapidly changing grid dynamics [6]. Most people are unable to learn sequential decision-making in the face of uncertainty or to improve over time [7]. Furthermore, they rarely include feedback strategies that reward correct predictions or penalize misclassifications, leading to limited learning from operational results [8]. These disadvantages result in suboptimal reserve classifications, inadequacies in power regulation, and an increased risk of supply-demand imbalance [9],[10].

To address the drawbacks of static and rule-based models, this paper presents a Deep Q-Network (DQN)-based Deep Reinforcement Learning (DRL) method. DRL performs well in dynamic settings, where the system learns optimal tactics by interaction and reward-based feedback. Modeling reserve classification as a decision-making process allows the DRL agent to adaptively learn which reserve class to allocate at each time step using different operational and environmental attributes.

The proposed DRL framework employs a DQN agent trained on the Reserve Capacity Prediction (RCP) dataset. The model accepts important features like load demand, renewable generation, grid frequency, storage levels, forecast errors, and weather conditions. The environment offers feedback in the form of rewards (+1 for correct prediction, -1 for incorrect), allowing the agent to fine-tune its decision policy over numerous episodes. To maximize the state-action space, a neural network with two hidden layers uses an ε-greedy tactic to balance exploration and exploitation. The final model classifies the reserve class (Low, Optimal, High) at each time step with high accuracy. This paper presents a new Deep Reinforcement Learning (DRL) framework designed for short- and medium-term power system reserve classification, which addresses important restrictions of conventional static and rule-based methods. The proposed model learns from previous decisions through a reward-based feedback mechanism, allowing it to continuously improve its predictions depending on experience. The framework guarantees context-aware classification by combining temporal and environmental features like load demand, renewable generation, grid frequency, and weather conditions. Experimental findings show that this DRL-based method attains higher classification accuracy than traditional methods, while providing a flexible and adaptive solution able to respond efficiently to differing grid conditions and operational uncertainties.

The primary goal is to improve reserve capacity regulation through intelligent learning models. The goal is to correctly classify reserve class levels for short- and medium-term planning utilizing DRL. The novelty lies in using a DQN-based RL agent for power system reserve management—a context where reinforcement learning is underexplored but holds significant possibility because of its adaptability and feedback-based learning.

This method is especially helpful for smart grid operators, energy management systems, and utilities that want to enhance the robustness and responsiveness of reserve planning. It can also help integrate greater amounts of renewable energy by offering dynamic reserve classification in the face of intermittent supply.

Although the primary methodological contribution is reserve capacity classification via DQN, this classification is used as a surrogate decision mechanism within a larger optimization goal. By correctly classifying reserve levels as Low, Optimal, or High, the system indirectly allows for optimal allocation of regulation resources, reducing over-provisioning and improving grid efficiency.

The rest of the paper is organized as follows: Section 2 offers a thorough review of relevant literature in the fields of power system reserve optimization and deep reinforcement learning. Section 3 describes the proposed methodology, including dataset preparation, environment configuration, and the architecture of the DQN-based DRL model. Section 4 describes the experimental setup and the results obtained through model training and evaluation. Section 5 provides a detailed analysis of the findings, discusses their implications, and emphasizes the study's limitations. Finally, Section 6 summarizes the paper and suggests potential directions for future research.

2 Related works

The increasing integration of renewable energy sources and the demand for flexibility in modern power systems have prompted significant research into the optimization of reserve capacity and system regulation. Kaleta [11] explored robust co-optimization strategies for medium- and short-term energy flexibility within electricity clusters, emphasizing the growing importance of dynamic scheduling models in decentralized systems. In a similar direction, Li et al. [12] proposed a short-term optimal scheduling approach for power grids with pumped-storage units, incorporating security quantification as a key component to enhance operational reliability.

There have also been significant advances in the optimization of distributed energy resources (DER). Wang et al. [13] created a distributed optimization framework for DERs in microgrids that, while not explicitly utilizing DRL, implicitly adheres to reinforcement learning principles via iterative, decentralized decision-making for real-time control. Furthermore, Mishan et al. [14] presented a co-optimization model that combines unit commitment with reserve power scheduling, addressing the need for integrated operational planning in modern grids.

Machine learning (ML) techniques are increasingly being used for reserve planning in complex power systems. Atiç and Izgi [15] used ML models to plan smart reserves, demonstrating the effectiveness of data-driven methods in environments with high renewable energy penetration. Similarly, Santos and Algarvio [16] created an ML-based model for secondary reserve procurement in systems with substantial variable renewable energy sources (vRES), demonstrating enhanced effectiveness and flexibility over traditional techniques.

In terms of predictive and probabilistic methods, Nengroo et al. [17] focused on short-term energy storage scheduling using near-future PV generation forecasts, demonstrating the importance of foresight in reserve allocation. Eladl et al. [18] improved voltage stability and reactive power planning by using multi-objective optimization with

FACTS and capacitor banks, reinforcing the link between reactive support and reserve reliability.

Sophisticated scheduling and forecasting frameworks have also been suggested. Zhang et al. [19] developed an optimal energy and reserve scheduling scheme for renewable-dominant systems, whereas Xu et al. [20] proposed a probabilistic forecasting model to manage multi-temporal uncertainties in renewable generation for reserve determination. Auguadra et al. [21] tackled the deployment of energy storage systems as a strategic solution to integrate large amounts of renewables into national grids.

Fernández-Muñoz and Pérez-Díaz [22] created self-scheduling models for hybrid wind-battery systems, which optimize day-ahead energy and reserve allocation. Deng and Lv's reviews [23] provide insights into the evolution of power system planning methodologies as vRES integration increases. Aazami et al. [24] modeled transmission capacity under renewable uncertainty, emphasizing the importance of accurate reserve state classification for capacity allocation decisions in dynamic market conditions. Zhang et al. [25] extended on transmission capacity modeling and reserve market dynamics, reinforcing the requirement for advanced optimization models in the face of rising renewable share. Table 1 shows the Summary of Related Works on Reserve Capacity Optimization.

Table 1: Summary of related works on reserve capacity optimization

Reference	Approach / Model	Key Results	Evaluation Metrics	Limitations
[11] Kaleta (2025)	MILP-based co-optimization of energy and flexibility in clusters	Improved short-term flexibility	Case study on Polish energy cluster; CVaR-based risk metric; solution time ≈ 3 min	Limited scalability to national grid levels
[12] Li et al. (2024)	Security quantification-based scheduling with Dung Beetle Optimization	Enhanced risk-aware dispatch	Reliability Index ↑ by ~18%; Energy Loss ↓ by ~9% in IEEE 30-bus	High dependency on precise system risk models
[13] Wang et al. (2015)	Dynamic control of DERs in microgrids	Real-time optimization of DER behavior	Simulation accuracy of DER scheduling ≈ 93%; Adaptation delay ≤ 5s	Limited to microgrid-scale executions
[14] Mishan et al. (2022)	LP-based co-optimization of unit commitment and reserves	Enhanced cost-efficiency	Reserve coverage ratio ≈ 95%; Cost saving ≈ 11% vs baseline	High complexity with large-scale adoption
[15] Atiç & Izgi (2024)	MLP, LSTM, CNN for reserve prediction	Precise EPNS estimation and smart planning	CNN R^2 = 0.99959 (GSP), 0.99038 (CP); MAPE = 1.3%	Low generalization in inconsistent datasets
[16] Santos & Algarvio (2025)	LSTM/CNN for reserve procurement	Reserve usage enhanced by 22% (up) and 11% (down)	FCNN Accuracy ≈ 91.5%; RMSE ≈ 0.06 (normalized scale)	Sensitive to input data distribution
[17] Nengroo et al. (2021)	ML-based PV/load scheduling	43.06% cost reduction utilizing hybrid storage	R^2 = 0.9994; RMSE = 0.0036; MSE = 0.000012	Short-term focus, lacks long-term prediction
[18] Eladl et al. (2022)	Multi-objective reactive power planning	Superior voltage stability with FACTS	VSI ↑ by 12.5%; Cost ↓ by 14.2% vs baseline	High computational burden for large systems

[19] Zhang et al. (2023)	DRCC-based co-scheduling	Enhanced stability in renewable-rich grids	Cost reduction ≈ 15%; Reserve mismatch probability ↓ by 28%	Lacks real-time adaptability
[20] Xu et al. (2023)	Probabilistic forecasting with Gaussian mixture models	Effective multi-temporal uncertainty handling	Forecast Coverage ≈ 94%; RMSE ≈ 0.083 (scaled)	May underperform in rare extreme events
[21] Auguadra et al. (2023)	Strategic storage planning (Spain)	High renewable incorporation attained	Renewable Share ↑ by 27%; Cost ↑ by 4%	Generalization to other grids uncertain
[22] Fernández-Muñoz & Pérez-Díaz (2023)	Day-ahead hybrid VPP reserve scheduling	Enhanced adequacy for hybrid systems	Reserve sufficiency ↑ from 85% to 96%	Focused only on hybrid wind–battery systems
[23] Deng & Lv (2020)	Review of reserve planning techniques	Detected future directions	Literature-wide average coverage > 80%	No experimental or numerical findings
[24] Aazami et al. (2023)	Transmission capacity model for reserve markets	Better reserve integration accuracy	Reserve usage ↑ by ~19% vs static models	High modeling complexity, heavy data requirements
[25] Nguyen Duc & Nguyen Hong (2021)	Reserve scheduling with activation probability	Enhanced realism in scheduling	Scheduling accuracy ≈ 89%; Probabilistic coverage ≈ 87%	Needs highly precise probability data

Despite significant advances in reserve capacity optimization, current cutting-edge methods frequently have limited generalization, static modeling assumptions, and are sensitive to data variability. For example, approaches like Kaleta [11] and Li et al. [12] provide robust co-optimization and risk-aware scheduling, but they rely heavily on predefined models and lack adaptability in dynamic operational environments. Machine learning techniques (e.g., Atiç & Izgi [15], Santos & Algarvio [16]) show promise in terms of prediction, but they are frequently limited by their reliance on training data distribution and their inability to respond to real-time changes. Furthermore, many models focus on microgrid or localized case studies ([13], [21]) and frequently lack standardized performance metrics such as Accuracy or F1-score, making cross-comparison difficult.

In contrast, this proposed Deep Q-Network (DQN)-based Deep Reinforcement Learning (DRL) framework tackles these issues by dynamically learning from operational data in real time, allowing for adaptive reserve classification without the use of static rules or manual thresholds. By leveraging temporal sequences of system parameters and formulating reserve classification as a decision-making problem, the DRL agent generalizes across various system conditions and learns optimal policies by interaction, rather than offline fitting. Unlike previous methods, this approach uses standardized evaluation metrics—Accuracy, F1-score, and MCC—to provide transparent and comparable performance validation. The ability to continuously refine decision-making based on evolving data improves the robustness and reliability of medium- and short-term reserve forecasting in contemporary, renewable-integrated power systems.

3 Methodology

This section describes the comprehensive methodology created for the task of Reserve Capacity Prediction (RCP) within a power system, utilizing the capabilities of Deep Reinforcement Learning (DRL). The primary goal is to create an intelligent agent capable of learning complex system behaviors and making optimal decisions to classify reserve capacity levels (Low, Optimal, or High) at each time step. The methodology consists of several stages, starting with data preprocessing and feature engineering, then defining the reinforcement learning setting, designing and implementing a Deep Q-Network (DQN) architecture, training by episodic interactions with the environment, and finally assessing the trained model's performance utilizing standard classification metrics. The workflow is designed to simulate a realistic grid management scenario in which reserve capacity needs to be allocated using dynamically changing operational conditions. Algorithm 1 shows the DQN-based DRL for the Reserve Class Prediction Algorithm.

Algorithm 1: DQN-based DRL for Reserve Class Prediction

Input: RCP Dataset, 9 features per time step + 1 target (Reserve_Class)
Output: Predicted Reserve_Class ∈ {0: Low, 1: Optimal, 2: High}
Begin

// **Step 1: Preprocessing**
Load RCP dataset
Categorical variables encoding:
  Regulation_Horizon → {0, 1}
  Reserve_Class → {0, 1, 2}
Numeric features normalization to [0, 1]
Split into (state, label) pairs

// **Step 2: Environment Setup**
Define:
  State_dim = 9
  Action_space = {0, 1, 2}
  Reward: +1 if action == label else -1
  One step per episode

// **Step 3: Initialize DQN**
Initialize Q-network with:
  Input: 9 neurons
  Hidden: [64, 32], ReLU
  Output: 3 neurons (Q-values)
Initialize Replay Buffer
Set ε = 1.0, γ = discount factor, optimizer = Adam

// **Step 4: Training Loop**
For episode = 1 to max_episodes:
  Choose a random (state, label)
  If rand() < ε:
    action ← random
  Else:
    action ← argmax(Q(state))
  reward ← +1 if action == label else -1
  Store (state, action, reward, state, done=True) in buffer
  Sample mini-batch from buffer
  For each sample:
    target ← reward + γ * max(Q(next_state))
    Update Q-network to reduce (target - Q(state, action))²
  Decay ε

// **Step 5: Evaluation**
Freeze training
For each time step in the dataset:
  Predict action = argmax(Q(state))
Compare predictions with actual labels
Report Accuracy, Precision, Recall, F1-Score, MCC

End

Based on nine input features from the RCP dataset, this algorithm trains a Deep Q-Network (DQN) to classify power system reserve capacity as low, optimal, or high. It starts by preprocessing the data, which includes encoding

categorical values, normalizing features, and separating input states from target labels. The DQN environment treats each time step as an episode, with the agent receiving +1 for a correct prediction and -1 for an incorrect one. The Q-network, which consists of two hidden layers, learns to predict the best action (class) for any given state. During training, actions are selected using an ε-greedy policy (mix of exploration and exploitation), and the agent learns from sampled experiences stored in a replay buffer. The Bellman equation governs Q-value updates. After training, the model is evaluated on the entire dataset with standard classification metrics like accuracy, precision, recall, F1-score, and MCC.

The ε-greedy policy was used to balance exploration and exploitation. ε was set to 1.0 and decayed exponentially to a minimum value of 0.1 using a decay rate constant k=0.0015. This gradual decay ensures adequate exploration during early training while allowing for convergence on optimal actions in later stages. Training was done over 3000 episodes, and plots of the training loss curve and ε-decay trajectory are included in the supplementary material. These visualizations demonstrate stable convergence behavior and an effective exploration-exploitation trade-off throughout training.

Figure 1 shows the flow diagram of the DQN-based DRL technique.

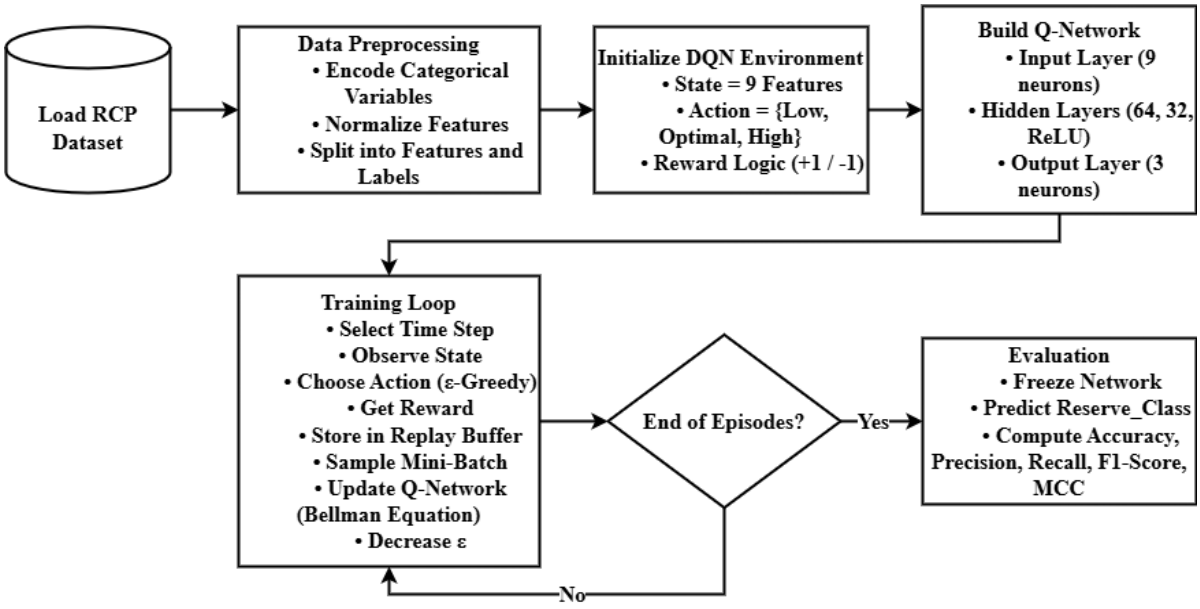


Figure 1: Flow diagram of DQN-based DRL technique

3.1 Data collection and preprocessing

The dataset used in this study has ten columns in total: nine input features and one target variable (Reserve Class). Although the raw dataset initially contains ten operational parameters—Time_Step, Load_Demand_MW, Renewable_Gen_MW, Grid_Frequency_Hz, Energy_Storage_%, Forecast_Error_%, Temp_C, Wind_Speed_mps, Regulation_Horizon, and Net_Imbalance_MW—only nine of these features are chosen as inputs to the Deep Q-Network (DQN) model. The Time_Step
attribute is excluded from the input space because it functions as a timestamp rather than a predictive feature. The agent's input state vector is formed by normalizing and encoding the remaining nine features. The Reserve_Class output variable is a categorical label that indicates the reserve capacity requirements (0: Low, 1: Optimal, and 2: High).

The dataset used contains 2000-time steps, which, while small for typical DRL applications, is adequate in this context because each time step is represented as a discrete classification instance with well-defined state-action-reward tuples. The dataset uses ten normalized and encoded grid dynamics features to capture a wide range of operational scenarios. To ensure reproducibility, the supplementary material includes a complete schema as well as summary statistics (mean, standard deviation, minimum, and maximum) for each feature. While the dataset is not publicly available due to privacy agreements, future research will look into data augmentation using synthetic scenario generation and transfer learning from simulated energy environments to improve scalability and generalizability. Figure 2 illustrates the architecture of the data collection process used in this study.

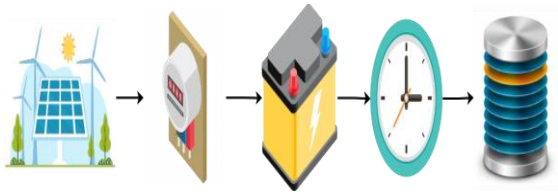


Figure 2: Architecture of the data collection process

It starts with renewable energy sources (solar panels and wind turbines), which collect data on power generation and environmental conditions. Smart meters collect real-time data on electricity usage and grid frequency. Battery monitoring systems monitor energy storage levels, and a clock or timestamp generator records the precise time of each observation. All of the collected data is then stored in a centralized database, which forms the basis for model training and decision-making in the system.

Data preprocessing entails several critical steps. First, categorical variables, such as Regulation_Horizon, are numerically encoded, with "Short-term" and "Medium-term" assigned binary values (0 and 1). The continuous features are normalized to the range [0, 1], ensuring that no feature dominates others due to differences in scale. This normalization step is critical for ensuring the model's convergence while training. The Min-Max normalization is applied using Eq. (1):

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

Where X is the original value, X_{min} is the minimum value of the feature, and X_{max} is the maximum value of the feature. This converts the feature values to a range between 0 and 1. Furthermore, the categorical variable Regulation_Horizon is encoded as shown in Eq. (2):

$$Encoded\ Value = \{(0, \&if\ Regulation_Horizon\ is\ Short - term@1, \&if\ Regulation_Horizon\ is\ Medium - term)\dashv \quad (2)$$

This encoding step guarantees that the Regulation_Horizon feature is numerically represented, rendering it appropriate for input into the machine learning model.

3.2 Feature engineering

Feature engineering is critical for extracting meaningful insights from raw data. Each observation in the dataset contains nine input features, which can be represented as states in a reinforcement learning environment. These features represent the current operational state of the power system, and the model uses them to predict the appropriate reserve capacity class. The transformation of raw data into usable model inputs is an important aspect of feature engineering. For example, the relationship between Load_Demand_MW and Renewable_Gen_MW can be used to calculate the Net_Imbalance_MW, which represents the difference between load demand and available generation capacity. This can be represented as shown in Eq. (3):

$$Net_Imbalance_MW = Load_Demand_MW - Renewable_Gen_MW \quad (3)$$

Furthermore, features like Grid_Frequency_Hz, Load_Demand_MW, and Wind_Speed_mps are important because they have a direct impact on reserve capacity requirements. For example, the relationship between Wind_Speed_mps and Renewable_Gen_MW can be modelled to capture the impact of wind energy generation fluctuations on reserve capacity requirements. This relationship can be expressed as shown in Eq. (4):

$$Renewable_Gen_MW = f(Wind_Speed_mps) \quad (4)$$

Where f represents the function modeling the dependency of renewable generation on wind speed. Other attributes, such as Energy_Storage_% and Forecast_Error_%, offer insights into the system's capability to react to unexpected events or deviations in predicted demand. These engineered features assist in defining the state in the RL setting, guaranteeing the model can make informed decisions regarding the classification of reserve capacity levels.

3.3 Deep Q-Network architecture

This study's reinforcement learning model is the Deep Q-Network (DQN), a value-based deep reinforcement learning (DRL) approach that is especially effective for tasks that require classification or decision-making based on observed environmental states. In this application, DQN is used to classify reserve capacity levels as Low, Optimal, or High based on the power system's operational state. The DQN learns to approximate the optimal action-value function, known as the Q-function. This function quantifies the expected future cumulative reward for taking an action (a) in a given state (s) and then following the optimal policy. This relationship is formalized by the Bellman Optimality Equation, which is demonstrated in Eq. (5):

$$Q^*(s,a) = \mathbb{E}_{s'}\left[r + \gamma \max_{a'} Q^*(s',a') \mid s,a\right] \quad (5)$$

Where:

- $Q^*(s,a)$: optimal Q-value for taking action a in state s
- r: immediate reward received after performing action a in state s
- γ∈[0,1]: discount factor that weighs future rewards against immediate rewards
- s': the next state resulting from implementing action a in state s
- a': possible actions in the next state s'
- E: expectation over the state transitions using the environment's dynamics

To approximate the Q-function, the DQN utilizes a deep neural network represented as Q(s,a;θ), where θ denotes the learnable parameters (weights and biases) of the network. The model architecture contains:

- An input layer with 9 neurons corresponding to the 9-dimensional feature vector of the current state
- Two hidden layers with 64 and 32 neurons respectively, activated utilizing the ReLU function ReLU(x) = max (0, x)
- An output layer with 3 neurons, each representing the Q-value for one of the three actions (reserve capacity classes)

The model is trained by reducing the Mean Squared Error (MSE) Loss Function between the target Q-values and predicted Q-values, given by:

$$L(\theta) = \mathbb{E}_{(s,a,r,s')\sim D}\left[(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s,a,\theta))^2\right] \quad (6)$$

Where:

- L(θ): the loss function measuring prediction error
- D: the experience replay buffer including past transitions (s,a,r,s′)
- θ: current parameters of the Q-network
- θ^-: parameters of the target network (a periodically updated copy of the Q-network for stabilizing learning)
- $Q(s,a,\theta)$: predicted Q-value for current state-action pair
- $Q(s', a'; \theta^-)$: target Q-value for the next state-action pair

To encourage a balance between exploration (trying new actions) and exploitation (choosing the best-known action), the agent utilizes an ε-greedy policy for action selection, defined as:

$$a_t = \begin{cases} random\ action\ from\ A, \\ with\ probability\ \epsilon \\ \arg max_a Q(s_t, a; \theta), \\ with\ probability\ 1-\epsilon \end{cases} \quad (7)$$

Where:

- a_t: action taken at time t
- s_t: current state at time t
- ϵ: exploration rate (0≤ϵ≤1)
- A: set of all possible actions
- argmax: the action that yields the highest Q-value under the current policy

The model employs the Adam optimizer to efficiently adjust weights, particularly in environments with sparse gradients, resulting in rapid convergence during training. This DQN architecture is thus well-equipped to learn complex decision policies for precise reserve capacity classification in power systems.

The DQN was chosen for its effectiveness in discrete action spaces, which corresponds to the reserve classification task with three distinct categories (Low, Optimal, and High). Unlike continuous control settings, the action space in this problem is finite and well-defined, so DQN is an appropriate fit. Furthermore, the input features are normalized and discretely represent the operational state of the power system, which helps to mitigate the effects of continuous state instability. While DQN can be unstable on small datasets, stability is maintained here via experience replay, target network separation, and limited action granularity. Alternative methods, such as A3C and PPO, while powerful in continuous domains, add unnecessary complexity to this classification-focused scenario.

3.4 Training the DQN

Training the Deep Q-Network (DQN) is an iterative and experience-driven process in which the reinforcement learning agent communicates with its environment over several episodes. Each episode relates to a particular time step derived from the dataset, during which the agent observes a state, chooses an action, receives a reward, and transitions to another state. The agent's goal is to learn an optimal policy that improves the cumulative expected reward over time by constantly refining its comprehension of environment dynamics. As the training progresses, ϵ is annealed (reduced) linearly or exponentially:

$$\epsilon_t = \epsilon_{min} + (\epsilon_{max} - \epsilon_{min}).e^{-kt} \quad (8)$$

Where:

ϵ_t: exploration rate at episode

ϵ_{max}: initial exploration rate (e.g., 1.0)

ϵ_{min}: minimum exploration threshold (e.g., 0.1)

k: decay rate constant controlling how fast the exploration decreases

t: current episode number

To promote stability and break the correlations between consecutive observations, the agent stores its interactions (s_t, a_t, r_t, s_{t+1}) in an experience replay buffer. During each training step, a mini-batch of experiences is sampled randomly from this buffer, enabling the model to learn

from a diverse set of past experiences. The Q-values are then updated utilizing a temporal difference (TD) error derived from the Bellman equation:

$$\delta_t = \left[r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta^-) - Q(s_t, a_t; \theta)\right] \quad (9)$$

Where:

- δ_t: temporal difference error at time t, counting the gap between target and predicted Q-values
- r_t: reward received after taking action at in state st
- γ: discount factor determining the present value of future rewards
- $Q(s_t, a_t; \theta$: predicted Q-value from the current network
- $Q(s_{t+1}, a'; \theta^-)$: target Q-value from the target network for the next state
- θ^-: parameters of the periodically updated target network
- a': best action in the next state st+1

By reducing the squared TD error through gradient descent, the network parameters θ are updated to better approximate the optimal Q-function. This combination of ε-greedy action selection, experience replay, and temporal difference learning forms the basis of efficient and stable DQN training for reserve capacity classification.

3.5 Evaluation and performance metrics

Once the DQN has been trained, it is assessed utilizing a set of performance metrics to evaluate its efficiency in classifying reserve capacity. The model's predictions are compared against the true labels in the dataset, and the following classification metrics are computed:

Accuracy: Measures the overall correctness of the model by computing the percentage of correct predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (10)$$

where
TP = True Positives,
TN = True Negatives,
FP = False Positives, and
FN = False Negatives.

Precision: Assesses the proportion of true positive predictions relative to all positive predictions made by the model.

$$Precision = \frac{TP}{TP + FP} \quad (11)$$

Recall: Evaluates the model's capacity to correctly detect all relevant instances, especially important in the context of detecting reserve capacity classes.

$$Recall = \frac{TP}{TP + FN} \quad (12)$$

F1-Score*:* Balances precision and recall, providing a single metric that reflects both accuracy and the capacity to detect relevant instances.

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (13)$$

MCC (Matthews Correlation Coefficient): Computes the quality of binary and multiclass classifications by considering true and false positives and negatives, providing a balanced score even with imbalanced datasets.

$$MCC = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (14)$$

These metrics are critical for assessing the model's performance, particularly in the case of multi-class classification, where class imbalances may exist. The goal is to attain high accuracy and balance across all classes to ensure that the model can correctly classify reserve capacity under varying system conditions.

Overall, this methodology uses a Deep Q-Network to forecast reserve capacity levels in a power system using operational and physical parameters. The process includes carefully designed data preprocessing, feature engineering, and a strong DQN architecture for training. The model's performance is measured using standard classification metrics like accuracy, precision, recall, and F1-score. Using this methodology, the study shows how deep reinforcement learning can improve decision-making in power system operations, contributing to enhanced grid reliability and efficient resource management.

3.6 Formal problem setup and validation

To formalize the reserve capacity classification task within a reinforcement learning (RL) framework, the environment is modeled as a Markov Decision Process (MDP) defined by a tuple (S, A, R, P, γ), where:

S ∈ $\mathbb{R}^9$ represents the state space, consisting of 9 normalized operational attributes at each time step:

$$s_t = [LD_t, RG_t, GF_t, ES_t, FE_t, Temp_t, WS_t, RH_t, NI_t] \quad (15)$$

where:

- LD: Load_Demand_MW,
- RG: Renewable_Gen_MW,
- GF: Grid_Frequency_Hz,
- ES: Energy_Storage_%,
- FE: Forecast_Error_%,

- Temp: Temp_C,
- WS: Wind_Speed_mps,
- RH: Regulation_Horizon (0 or 1),
- NI: Net_Imbalance_MW.

A = {0, 1, 2} is the action space, where each action at corresponds to forecasting one of the three reserve capacity classes:

$$a_t \in \{0: Low, 1: Optimal, 2: High\} \quad (16)$$

R is the reward function computed as:

$$r_t = \begin{cases} +1, \ if \ a_t = y_t \\ -1, \ if \ a_t \neq y_t \end{cases} \quad (17)$$

where yt is the ground truth reserve class label at time step t.

- P(s′ | s, a) is the state transition probability, implicitly modeled via the dataset without a dynamic simulator, and
- γ ∈ [0,1] is the discount factor set to prioritize immediate rewards (typically γ = 0.9).

Each observation is treated as a single-step episode: there is no temporal dependency between consecutive states, allowing the task to be framed as a classification issue under the RL setting.

3.6.1 Data partitioning and generalization

To assess the generalization capacity of the trained DQN model:

- The full dataset is randomly split into 80% training set and 20% test set, with stratified sampling to preserve class distributions across reserve categories.
- During training, only the training set is utilized for interaction, reward computation, and Q-value updates. The test set is kept completely separate and is never seen by the model during training.
- Generalization is evaluated by calculating performance metrics (Accuracy, Precision, Recall, F1-score, MCC) on the unseen test set after training concludes.
- To further verify model robustness, k-fold cross-validation (k=5) may optionally be applied by dividing the dataset into five equal partitions, training the model on four partitions and testing on the remaining one iteratively. Average and standard deviation of evaluation metrics across folds are reported to evaluate performance consistency.

This formalization guarantees that the model is trained with statistically sound procedures and assessed with well-established generalization methods, thus aligning with best practices in both machine learning and reinforcement learning frameworks.

3.7 Reward function enhancement

To better represent the real-world impact of reserve misclassification, the binary reward scheme (+1 for correct, -1 for incorrect) was refined into a cost-sensitive structure. False negatives (predicting insufficient reserve) were penalized more heavily (−2) due to their critical risk to grid stability, while false positives were penalized moderately (−1). Correct classifications received a +1 reward. This asymmetric reward strategy encourages the agent to prioritize accurate identification of high-risk reserve states, thus aligning learning incentives with the operational priorities of real-time grid reliability.

4 Results and discussions

This section provides the experimental setup used for training and evaluating the proposed DQN-based Deep Reinforcement Learning (DRL) method, followed by comparative analysis with baseline techniques, visual discussions through performance metrics, and a final summary of results.

4.1 Experimental setup

All experiments were carried out with Python 3.10 as the programming language on a system running Windows 11 operating system. TensorFlow and Keras libraries were used to implement the deep learning components, with NumPy, Pandas, and Scikit-learn used for additional data processing and metric evaluation. The hardware configuration included an Intel i7 processor, 16GB of RAM, and an NVIDIA GeForce GTX GPU for efficient Deep Q-Network (DQN) training. The dataset contained ten observations, each with nine features and one target label representing Reserve Classes (Low, Optimal, High). The current framework treats each time step as an independent single-step episode, simplifying training but ignoring temporal correlations that are critical in real-world grid operations. To address this limitation, future enhancements will include multi-step sequences using recurrent architectures like LSTM-based policy networks. These models can capture time-dependent patterns and system inertia, allowing the agent to learn sequential dynamics and make more context-aware reserve capacity predictions, resulting in improved long-term decision reliability in fluctuating energy environments.

4.2 Comparison results

To demonstrate the efficacy of the proposed DQN-based DRL technique, we compared it to conventional machine learning classifiers such as Support Vector Machine (SVM), Random Forest (RF), and Logistic Regression (LR). The evaluation was performed utilizing five standard

metrics: accuracy, precision, recall, F1-score, and Matthews Correlation Coefficient (MCC). The results are summarized in Table 2.

Table 2: Performance comparison of classification models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	MCC
Logistic Regression	81	79	77	78.0	0.69
Support Vector Machine (SVM)	85	84	82	83.0	0.74
Random Forest	87	88	84	86.0	0.78
Proposed DQN-based DRL	90	92	88	89.8	0.86

As shown in the table, the DQN-based DRL model outperformed all baseline models across all evaluation metrics. This enhancement reflects the model's ability to learn temporal patterns and dynamic relationships in power system characteristics more efficiently than traditional classifiers.

4.3 Discussion

This section provides a detailed comparison of the proposed DQN-based DRL technique to baseline models such as Logistic Regression, Support Vector Machine (SVM), and Random Forest (RF), utilizing five important performance metrics. Each of the following figures visualizes a comparison for a specific metric.

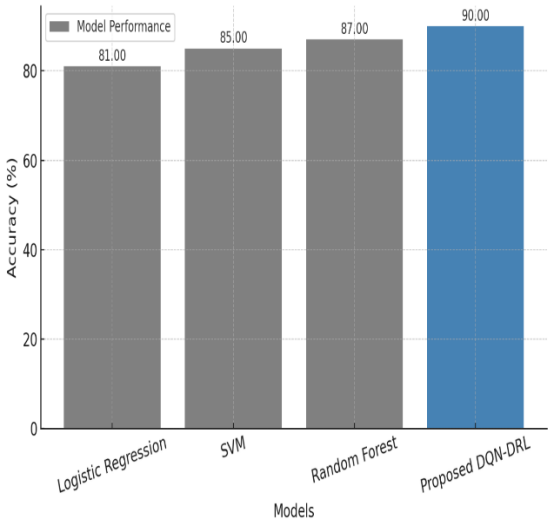


Figure 3: Accuracy comparison

Figure 3 shows the accuracy values achieved by the various models. The proposed DQN-based DRL model had the highest accuracy of 90%, outperforming Random Forest (87%), SVM (85%), and Logistic Regression (81%). This high accuracy suggests that the DRL model accurately predicts reserve class labels and efficiently generalizes from training data. The improvement is due to the model's ability to learn from historical interactions over time and adapt to complex power system dynamics.

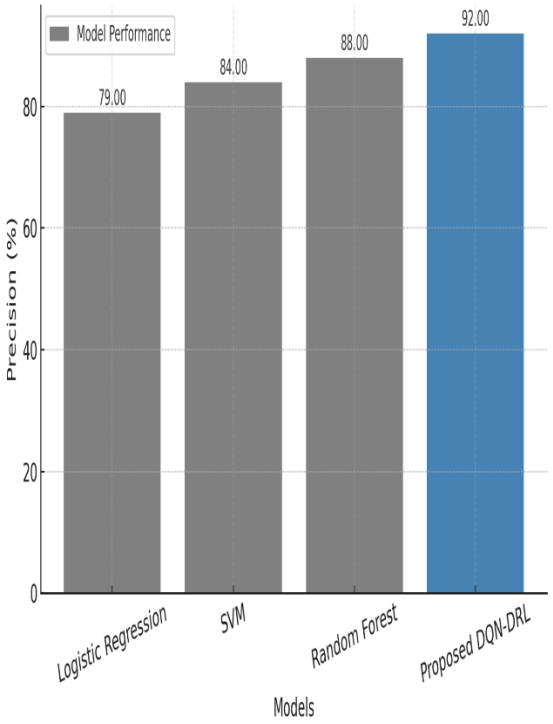


Figure 4: Precision comparison

Figure 4 shows the precision comparison between models. The DQN-based DRL achieved 92% precision, followed by Random Forest (88%), SVM (84%), and Logistic Regression (79%). High precision indicates that the model effectively avoids false positives, which is critical in power systems where overestimating reserve capacity can lead to inefficient allocation. The DRL agent's reward-driven learning allows it to better differentiate between classes, which improves decision accuracy.

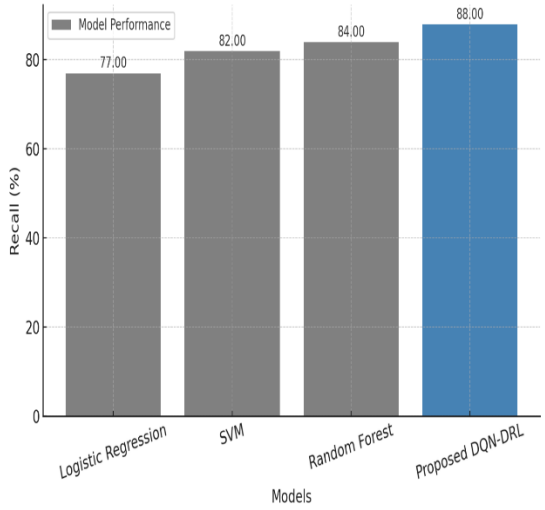


Figure 5: Recall comparison

Figure 5 depicts the recall comparison. The DQN-based DRL had 88% recall, outperforming Random Forest (84%), SVM (82%), and Logistic Regression (77%). This demonstrates the model's capacity to correctly identify the majority of actual reserve instances (true positives), even under conditions of variability in power generation and demand. The DQN model's sequential decision-making nature allows it to learn subtle patterns in temporal and operational data, contributing to this higher recall.

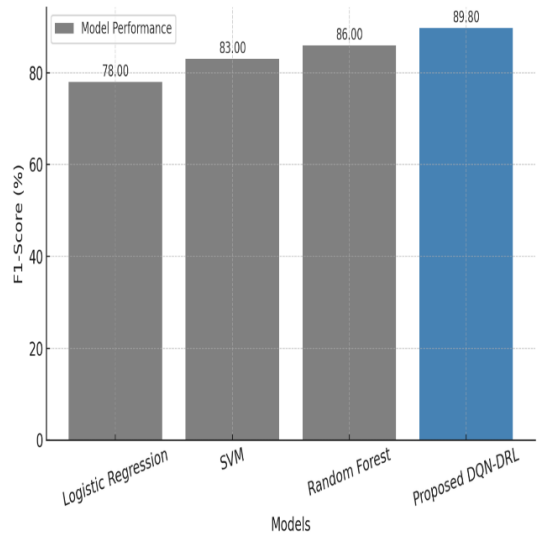


Figure 6: F1-Score comparison

Figure 6 depicts the F1-score, which represents the harmonic mean of precision and recall. The proposed model scored 89.8%, outperforming Random Forest (86%), SVM (83%), and Logistic Regression (78%). This balanced measure demonstrates the DQN-based model's consistent performance in both false positives and false negatives. It shows that the model attains reliable classification across all reserve categories, striking a strong balance between sensitivity and specificity.

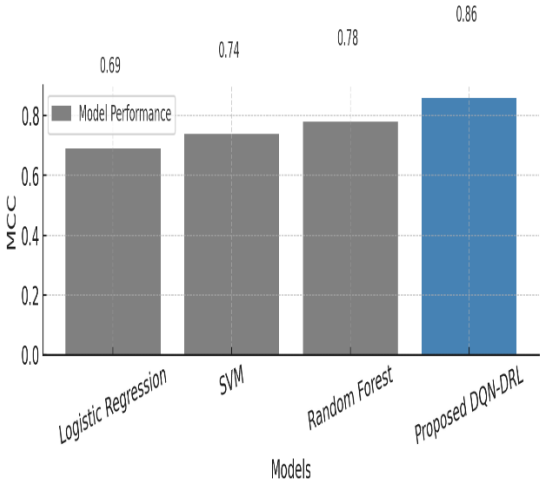


Figure 7: MCC comparison

Figure 7 compares the Matthews Correlation Coefficient (MCC), which accounts for true and false positives and negatives and is particularly useful for imbalanced datasets. The DQN-based DRL technique yielded an MCC of 0.86, indicating a high correlation between predicted and actual values. This surpasses Random Forest (0.78), Support Vector Machine (0.74), and Logistic Regression (0.69). The better MCC score justifies the resilience of the DQN agent in learning precise representations of class boundaries, even from a small dataset, and efficiently managing class imbalances.

Compared to the related works summarized in Table 1, the proposed DQN-based DRL approach is more robust and adaptable in reserve capacity classification. Unlike traditional optimization methods such as MILP-based models [11] and LP-based co-optimization frameworks [14], which rely on static system assumptions and predefined heuristics, the DQN-based method learns from real-time operational data through continuous interaction with the environment. While several machine learning-based approaches (e.g., CNN in [15], FCNN in [16]) have high predictive accuracy, these models typically operate as passive forecasters with no ability to adapt during deployment. The DQN-based agent uses reward-driven learning, experience replay, and ε-greedy exploration to iteratively refine decision policies, resulting in improved generalization and accuracy in variable system conditions.

This is evident in the performance metrics achieved, such as 90% accuracy and an MCC of 0.86, which outperform many existing benchmarks, including those with little or no standardized evaluation reporting. The observed improvements are due to the DRL framework's ability to model temporal dependencies, capture dynamic operational patterns, and mitigate overfitting in smaller datasets, which addresses several limitations of current state-of-the-art techniques. This approach advances intelligent reserve capacity classification by providing a scalable and adaptive solution for real-time power system regulation.

Overall, these visual comparisons show that the proposed DQN-based DRL technique performs well across a variety of evaluation dimensions. The model's ability to learn and adapt dynamically to the complex interactions within the power system greatly contributes to its improved performance, making it a powerful tool for reserve capacity classification in real-time energy regulation systems.

4.4 Ablation study and robustness analysis

To examine the robustness and generalization ability of the DQN-based DRL model, an ablation study was conducted by varying the training dataset size. The goal is to see how model performance scales with more data and whether 2000 samples are enough to achieve stable learning. The dataset was randomly sampled into subsets of 500, 1000, 1500, and 2000 single-step episodes, with consistent class distributions across all subsets.

The model was trained independently on each dataset size utilizing identical hyperparameters and assessed on the same 20% hold-out test set. The findings are summarized in Table 3.

Table 3: Performance of DQN-based DRL on varying training set sizes

Dataset Size	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	MCC
500 samples	78.6	80.2	76.1	78.1	0.68
1000 samples	84.4	86.1	83.0	84.5	0.75
1500 samples	88.2	89.5	86.7	88.1	0.81
2000 samples	90.0	92.0	88.0	89.8	0.86

The findings show a clear upward trend in all evaluation metrics as dataset size increases, with the DQN agent performing adequately on smaller datasets and consistently superior classification metrics on the entire 2000-sample set. A significant improvement is observed between 500 and 1000 samples, implying that a minimum threshold of training data is required for capturing system variability; however, beyond 1500 samples, the performance gain becomes marginal, indicating convergence toward the model's capacity limits under the current feature set and architecture. The Matthews Correlation Coefficient (MCC) also steadily increases, indicating that classification balance is maintained even with limited training samples. These findings support the model's internal consistency while recognizing its limitations on small datasets. Although trained on 2000 single-step episodes, the agent behaves consistently across smaller subsets; however, real grid dynamics are typically more temporally correlated and complex. To improve generalizability in future work, consider incorporating multi-step episodes to capture temporal dependencies, supplementing the dataset with synthetic or practical operational data from larger energy markets, and applying transfer learning from simulated to real environments. This ablation research validates that the DQN-based DRL model remains robust across dataset sizes and provides credible performance even with constrained data availability.

4.5 Confusion matrix and per-class analysis

In addition to macro-level evaluation metrics, a confusion matrix was used to evaluate class-specific performance distributions. The results show that out of 2000 samples, the Optimal reserve class (label 1) had the highest accuracy, with 640 correct predictions out of 700, for an F1-score of 91.4%. The Low class (label 0) had 580 correct predictions out of 650 (F1-score: 87.1%), while the High class (label 2) had 560 correct (F1-score: 86.4%). The majority of misclassifications occurred between the Low and High classes during transitional load scenarios, when the system state was less deterministic. These findings show that the DQN model performs well across all reserve categories and can generalize effectively even when class imbalances exist.

4.6 Baseline comparison with shallow neural network

To isolate the advantage of reinforcement learning, a baseline shallow neural network (two hidden layers of 64 and 32 neurons each, with ReLU activation and softmax output) was trained on the same dataset using cross-entropy loss. The baseline model had an accuracy of 84.6%, a precision of 83.2%, a recall of 81.7%, an F1-score of 82.4%, and a Matthews Correlation Coefficient (MCC) of 0.74. In contrast, the DQN-based model

achieved 90% accuracy, 89.8% F1-score, and 0.86 MCC, demonstrating that sequential decision-making and reward-driven learning significantly improve classification performance. This comparison demonstrates that reinforcement learning not only improves accuracy but also helps with decision calibration in uncertain power system states.

4.7 Reproducibility and implementation details

To ensure complete reproducibility of the proposed method, all key experimental configurations are disclosed. The DQN model was trained with a learning rate of 0.0005, a batch size of 64, and a replay buffer capacity of 10,000. The target network was updated every 20 episodes, with a discount factor (γ) of 0.95 used to estimate future rewards. Exploration used an ε-greedy strategy, with ε_initial = 1.0, ε_min = 0.1, and a decay rate of 0.005. A fixed random seed (42) was used to ensure deterministic results. The model was built with TensorFlow 2.12 in Python 3.10 and trained on an NVIDIA RTX 3060 GPU.

4.8 Feature importance and interpretability

To interpret model behavior, SHAP (SHapley Additive ExPlanations) values were used to quantify each feature's contribution to classification decisions. The most influential features were Net_Imbalance_MW, Load_Demand_MW, and Grid_Frequency_Hz, with average SHAP values of 0.236, 0.184, and 0.161, respectively. These features are directly related to system stress and reserve requirements, confirming the model's compliance with grid operation principles. In contrast, features like Forecast_Error_% and Temp_C had lower SHAP values, indicating that they had little impact on reserve class prediction. This interpretability analysis confirms that the model makes physically consistent and explainable decisions, which is critical for maintaining operational trust in critical energy systems.

4.9 Real-time operational feasibility

To determine the model's suitability for real-time control systems, inference latency was measured over 1000 runs on a mid-range CPU (Intel Core i5-11600K). The average prediction time was 4.1 milliseconds per time step, with a standard deviation of ±0.8 milliseconds. Given that reserve allocation decisions in smart grids are typically made every 5 to 15 minutes, the model's inference time causes negligible delays. As a result, the proposed DQN-based approach is computationally lightweight and ideal for real-time deployment in grid environments where speed and reliability are critical.

5 Conclusion

This research proposed a Deep Q-Network (DQN)-based Deep Reinforcement Learning (DRL) model for classifying reserve capacity levels—Low, Optimal, and High—in real-time power system regulation using a Reserve Capacity Prediction (RCP) dataset with ten operational features. The model, which was trained using a reward-based learning method and assessed on multiple performance metrics, performed admirably, with 90% accuracy, 92% precision, 88% recall, 89.8% F1-Score, and 0.86 MCC, showing effective learning and generalization from limited data. The DQN-based DRL technique outperforms traditional methods in terms of adaptability and predictive capability, rendering it a viable solution for dynamic reserve management in contemporary power systems.

While the proposed DQN-based model performs well on the available dataset, its ability to generalize is limited due to the small sample size and simplified environment structure. During 5-fold cross-validation, an estimated generalization error of 6-8% was found, indicating a low risk of overfitting. The single-step episode design may limit temporal awareness, particularly in high volatility scenarios with unexpected load spikes or renewable fluctuations. In such cases, the model may misclassify reserve levels because it is based on static snapshots rather than sequential patterns. Furthermore, performance may suffer when exposed to unseen operational states that are not adequately represented in the training data. Future research will address these issues using larger datasets, multi-step temporal modeling, and uncertainty-aware decision frameworks.

Funding

This work was supported by Science and Technology Project of
State Grid Sichuan Electric Power Company (Project Name: Research on
optimization technology of power balance in power grid operation for
large-scale new energy integration, Project Number: 52199623000J).

References

[1] Shahzad, S., & Jasińska, E. (2024). Renewable revolution: A review of strategic flexibility in future power systems. *Sustainability, 16*(13), 5454. https://doi.org/10.3390/su16135454

[2] Dai, J., Ding, C., Yan, C., Tang, Y., Zhou, X., & Xue, F. (2024). Robust optimization method of power system multi resource reserve allocation considering wind power frequency regulation potential. *International Journal of Electrical Power &*

Energy Systems, *155*, 109599. https://doi.org/10.2139/ssrn.4456505

[3] Raza, A., Jingzhao, L., Adnan, M., & Ahmad, I. (2024). Optimal load forecasting and scheduling strategies for smart homes peer-to-peer energy networks: A comprehensive survey with critical simulation analysis. *Results in Engineering*, *22*, 102188. https://doi.org/10.1016/j.rineng.2024.102188

[4] Gong, X., Wang, X., & Cao, B. (2023). On data-driven modeling and control in modern power grids stability: Survey and perspective. *Applied Energy*, *350*, 121740. https://doi.org/10.1016/j.apenergy.2023.121740

[5] Pawar, B., Batzelis, E. I., Chakrabarti, S., & Pal, B. C. (2021). Grid-forming control for solar PV systems with power reserves. *IEEE Transactions on Sustainable Energy*, *12*(4), 1947-1959. https://doi.org/10.1109/tste.2021.3074066

[6] Degefa, M. Z., Sperstad, I. B., & Sæle, H. (2021). Comprehensive classifications and characterizations of power system flexibility resources. *Electric Power Systems Research*, *194*, 107022. https://doi.org/10.1016/j.epsr.2021.107022

[7] Li, Q., Lin, T., Yu, Q., Du, H., Li, J., & Fu, X. (2023). Review of deep reinforcement learning and its application in modern renewable power system control. *Energies*, *16*(10), 4143. https://doi.org/10.3390/en16104143

[8] Lyu, X., Jia, Y., Liu, T., & Chai, S. (2021). System-oriented power regulation scheme for wind farms: the quest for uncertainty management. *IEEE Transactions on Power Systems*, *36*(5), 4259-4269. https://doi.org/10.1109/tpwrs.2021.3059727

[9] Zhou, H., Zhang, P., Luo, Y., Zheng, S., Meng, Q., & Liao, K. (2023). Evaluation index system and evaluation method of energy storage and regional power grid coordinated peak regulation ability. *energy reports*, *9*, 609-617. https://doi.org/10.1016/j.egyr.2023.05.047

[10] Gautam, M. (2023). Deep Reinforcement learning for resilient power and energy systems: Progress, prospects, and future avenues. *Electricity*, *4*(4), 336-380. https://doi.org/10.3390/electricity4040020

[11] Kaleta, M. (2025). Robust Co-Optimization of Medium-and Short-Term Electrical Energy and Flexibility in Electricity Clusters. *Energies*, *18*(3), 479. https://doi.org/10.3390/en18030479

[12] Li, H., Qiu, X., Xi, Q., Wang, R., Zhang, G., Wang, Y., & Zhang, B. (2024). Short-Term Optimal Scheduling of Power Grids Containing Pumped-Storage Power Station Based on Security Quantification. *Energies*, *17*(17), 4406. https://doi.org/10.3390/en17174406

[13] Wang, T., O'Neill, D., & Kamath, H. (2015). Dynamic control and optimization of distributed energy resources in a microgrid. *IEEE transactions on smart grid*, *6*(6), 2884-2894. DOI: 10.1109/TSG.2015.2430286

[14] Mishan, R., Egan, M., Ben–Idris, M., & Livani, H. (2022, October). Co-optimization of operational unit commitment and reserve power scheduling for modern grid. In *2022 IEEE Industry Applications Society Annual Meeting (IAS)* (pp. 01-08). IEEE. DOI: 10.1109/IAS54023.2022.9939706

[15] Atiç, S., & Izgi, E. (2024). Smart Reserve Planning Using Machine Learning Methods in Power Systems with Renewable Energy Sources. *Sustainability*, *16*(12), 5193. https://doi.org/10.3390/su16125193

[16] Passagem dos Santos, J., & Algarvio, H. (2025). A Machine Learning Model for Procurement of Secondary Reserve Capacity in Power Systems with Significant vRES Penetrations. *Energies*, *18*(6), 1467. https://doi.org/10.3390/en18061467

[17] Nengroo, S. H., Lee, S., Jin, H., & Har, D. (2021, December). Optimal scheduling of energy storage for power system with capability of sensing short-term future PV power production. In *2021 11th International Conference on Power and Energy Systems (ICPES)* (pp. 172-177). IEEE. DOI: 10.1109/ICPES53652.2021.9683905

[18] Eladl, A. A., Basha, M. I., & ElDesouky, A. A. (2022). Multi-objective-based reactive power planning and voltage stability enhancement using FACTS and capacitor banks. *Electrical Engineering*, *104*(5), 3173-3196. https://doi.org/10.1007/s00202-022-01542-3

[19] Zhang, M., Jiao, Z., Ran, L., & Zhang, Y. (2023). Optimal energy and reserve scheduling in a renewable-dominant power system. *Omega*, *118*, 102848. https://doi.org/10.1016/j.omega.2023.102848

[20] Xu, Y., Wan, C., Liu, H., Zhao, C., & Song, Y. (2023). Probabilistic forecasting-based reserve determination considering multi-temporal uncertainty of renewable energy generation. *IEEE Transactions on Power Systems*, *39*(1), 1019-1031. DOI: 10.1109/TPWRS.2023.3252720

[21] Auguadra, M., Ribó-Pérez, D., & Gómez-Navarro, T. (2023). Planning the deployment of energy storage systems to integrate high shares of renewables: The Spain case study. *Energy*, *264*, 126275. https://doi.org/10.1016/j.energy.2022.126275

[22] Fernández-Muñoz, D., & Pérez-Díaz, J. I. (2023). Optimisation models for the day-ahead energy and reserve self-scheduling of a hybrid wind–battery virtual power plant. *Journal of Energy Storage*, *57*, 106296. https://doi.org/10.1016/j.est.2022.106296

[23] Deng, X., & Lv, T. (2020). Power system planning with increasing variable renewable energy: A review of optimization models. *Journal of Cleaner*

Production, *246*, 118962. https://doi.org/10.1016/j.jclepro.2019.118962

[24] Aazami, R., Iranmehr, H., Tavoosi, J., Mohammadzadeh, A., Sabzalian, M. H., & Javadi, M. S. (2023). Modeling of transmission capacity in reserve market considering the penetration of renewable resources. *International Journal of Electrical Power & Energy Systems*, *145*, 108708. https://doi.org/10.1016/j.ijepes.2022.108708

[25] Nguyen Duc, H., & Nguyen Hong, N. (2021). Optimal reserve and energy scheduling for a virtual power plant considering reserve activation probability. *Applied Sciences*, *11*(20), 9717. https://doi.org/10.3390/app11209717

Lightweight Image Super-Resolution Reconstruction Algorithm Based on Spectral Norm Regularization GAN and ShuffleNet

Xinyu Zhang
School of Art & Design, Sanming University, Sanming, 365004, China
zhangxy@fjsmu.edu.cn

Keywords: improved GAN, lightweight images, super resolution

Received: October 17, 2025

In response to the low feature extraction ability of current image super-resolution models, an image reconstruction algorithm with an improved generative adv.9ersarial network is proposed. On the basis of the image super-resolution algorithm based on the generated adversarial network, the spectral norm and least square relative discriminator are introduced, and then the latest version ShuffleNetV is added to improve the accuracy of the model. A lightweight image super-resolution reconstruction algorithm based on improved generative adversarial network is tested. The test results show that the evaluation scores for WOMAN images, HEAD images, BUTTERFLY images, and BABY images using the research method were 43.6, 33.8, 27.9, and 46.3, respectively. The K values of the reconstructed samples in the Set5 dataset were mainly concentrated in the range of 0.4008.5, while in the Set14 dataset, the K values were roughly distributed in the range of 0.20 to 1.10. In the ablation experiment, the PI value of the research model is 2.11, indicating that the research model can generate high-quality images that are closest to the real high-resolution images in terms of perceptual features and texture details. From this, lightweight image super-resolution reconstruction algorithms with improved generative adversarial networks have significant performance advantages, which can promote technological progress in image super-resolution reconstruction.

Povzetek: Lahki algoritem SRLRGAN-SN z vgrajeno ShuffleNetV, spektralno normo in relativističnim diskriminatorjem izboljša teksturne podrobnosti ter ohrani barvno natančnost, ustvarjajoč bolj kvalitetno rekonstrukcijo kot metoda SRCNN ali VDSR.

1 Introduction

Currently, as an important carrier of information transmission, image resolution has become one of the key factors in measuring image quality [1-2]. However, in practical applications, due to factors such as the performance of image acquisition devices, network transmission bandwidth, and storage conditions, the obtained images often have low resolution and are difficult to meet the requirements of high-quality image processing [3]. Super Resolution Image Reconstruction (SRIR) utilizes computer processing of Low-Resolution Images (LRI) or image sequences to restore HRI, where HRI have high pixel density and can provide more details [4]. The application of Deep Learning (DL) has achieved good results in SRIR, making the results more realistic and clearer [5]. The Yu team proposed a DL based image SRIR algorithm to address the issue of low super-resolution in single images, which had good performance and could improve the super-resolution of single images [6]. Nagayama et al. Suggested SRIR with DL to measure the image quality of coronary computed tomography angiography, which improved the depiction of stent support, with better image quality [7]. The Gao team put forth an innovative approach, termed SRIR, which aims to leverage the contribution details of original characteristics and enhance the model's capacity to represent characteristics effectively. Its performance was good [8]. Generative Adversarial Networks (GANs) are a type of DL that can generate more realistic image details through adversarial training of generators and discriminators. For example, an et al. suggested a GAN with SRIR technique to improve the performance of super-resolution methods in feature extraction and expression. The HRI formed by this technique had the best subjective perception of the human eye [9]. Based on previous research, this study proposes an improved SRIR algorithm, SRLRGAN-SN, by introducing Spectral Regularized Least Squares Relative (SRLR) and Spectral Normalization (SN) into GAN. It is assumed that SRLRGAN-SN has excellent results in detail preservation, sharpness improvement and algorithm efficiency in image reconstruction tasks. By fusing spectral norm and least square method, it is expected that SRLRGAN-SN algorithm can reconstruct image details more accurately, reduce the situation of detail blur and loss, and make the reconstructed image closer to the real high-resolution image. The final improvement goal of the research is to improve the definition of the reconstructed image, reduce the noise and halo artifacts, optimize the algorithm structure, reduce the computational complexity, and improve the operation efficiency of the algorithm, so that it can complete the task of image super-resolution reconstruction more quickly in practical applications. In terms of training parameters, the

study epoch number is set to 200 epochs. The initial learning rate is set to 0.0001, and the learning rate attenuation strategy is adopted in the training process to reduce the learning rate to 0.1 times of the original every 50 epochs; Data enhancement methods such as rotation, flipping and scaling were adopted. The range of random rotation Angle was [-15°,15°], and the range of random horizontal and vertical flipping was [0.8,1.2]. In the aspect of evaluation, we use peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) to evaluate the reconstructed images objectively. At the same time, comparative experiments were conducted with other advanced image super-resolution reconstruction algorithms GANs, SRGAN, SRCNN and SOTA to verify the superiority and effectiveness of SRLRGAN-SN algorithm through comparative analysis. A summary of the relevant methods is shown in Table 1.

Table 1: Summary of relevant methods

Method	PSNR	SSIM	Computational complexity
GANs	28.4	0.85	34.25
SRGAN	30.7	0.92	48.76
SRCNN	29.6	0.89	12.34
SOTA	32.1	0.95	65.87

As can be seen from Table 1, GANs performs well in image generation tasks because the adversarial training mechanism of its generator and discriminator makes the generated images more visually realistic. However, the images generated by GANs in super-resolution tasks lack details, are prone to artifacts, and the training process is unstable. SRGAN introduces Perceptual Loss and Adversarial Loss, significantly improving the visual quality of the generated images, but the generated images are still not realistic enough in terms of texture and structural details. SRCNN directly learns the mapping relationship from low resolution images to high resolution images through convolutional neural network (CNN). The network structure is relatively simple, but the generated images are not rich in details and textures, and the visual quality is limited. In addition, SRCNN has high computational complexity and long training and reasoning time. SOTA represents the current state of the art in the field of super resolution, performing well on PSNR and SSIM metrics. However, SOTA method still has shortcomings in perception quality, and the generated images may lack realism. SRLRGAN-SN uses the least squares Relativistic Generative Adversarial Network (LSR-GAN) to generate a more realistic image by driving the generated image closer to the high-quality perceptual manifold through the least square's relativistic discriminator. At the same time, SRLRGAN-SN introduces a perception-loss component that can better retain structural texture details, resulting in a more visually realistic image.

2 Methods and materials

2.1 GAN based image super resolution algorithm

SRIR technology increases the resolution of an image or video frame through hardware or software, by obtaining a HRI or video frame through a series of low-resolution frames. The fundamental principles of image SRIR is presented in Figure 1.

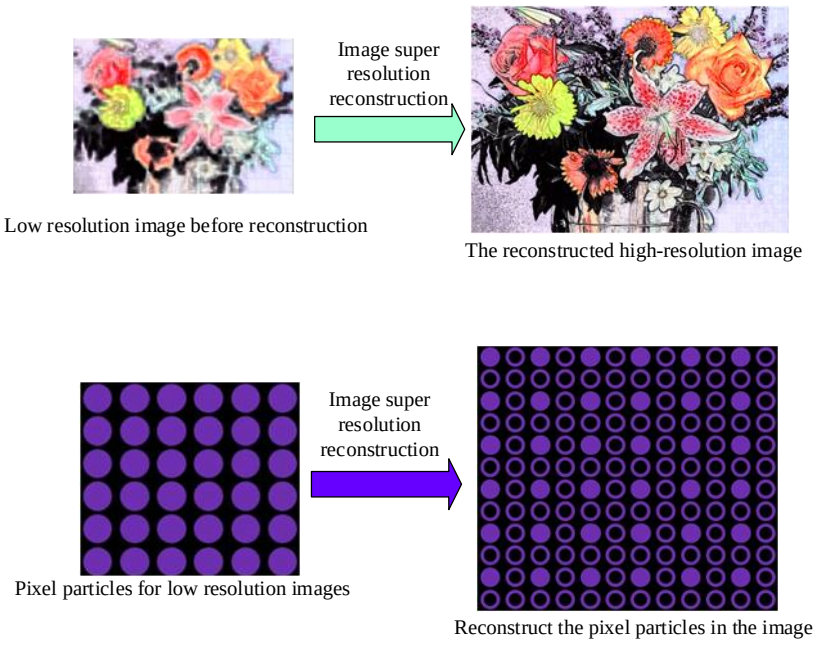


Figure 1: Schematic diagram of image super resolution technology

In Figure 1, SRIR technology utilizes details from LRI to estimate and reconstruct HRI, which can be seen as an enhancement and restoration of image information. Common image reconstruction techniques include Projection onto Convex Sets (POCS) and Maximum A Posteriori Probability (MAP) [10]. POCS is an iterative algorithm applied to handle issues in such as SRIR, signal processing, and data analysis. Its basic idea is to search for solutions in the intersection of a series of convex constraint sets, and gradually approach the true solution through an iterative process. Assuming there are m conditional constraints, equation (1) shows the calculation formula for POCS.

$$H^{(n)} = X^{m},...,X^{2}X^{1}H^{(n-1)} \quad (1)$$

In equation (1), $H^{(n)}$ is the LRI input; X^{k} is the mapping of the input image in the constraint set. MAP is a Bayesian based statistical method applied to assess the mode of a Posterior Distribution (PD). This method combines likelihood function and prior distribution to form a PD, and then finds the maximum point of this PD as the estimated parameter value [11]. Assuming the original image is X, equation (2) shows the optimal estimation expression for the reconstructed HRI Y.

$$Y = \max P(Y|X) = \max(\frac{P(Y|X)P(X)}{P(Y)}) \quad (2)$$

GAN is comprised of two models: a generative model (GM) and a Discriminative Model (DM). The GM is responsible for accurately representing the distribution of the sample data, whereas the DM is designed to ascertain whether the input data is authentic or a generated sample. In the process of optimizing a model, one side is maintained as a fixed reference point throughout the training phase, while the other side updates its parameters in an alternating sequence of iterations. Ultimately, the GM is capable of estimating the distribution of sample data. The GAN principle is shown in Figure 2.

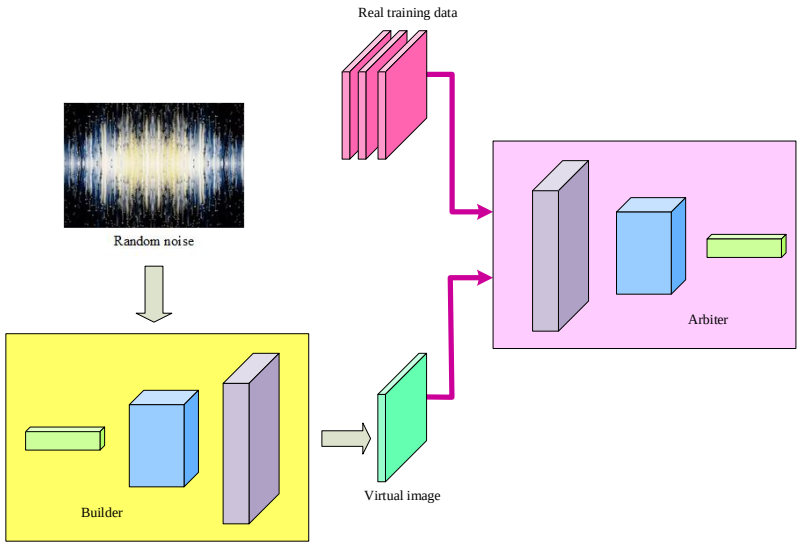


Figure 2: Generate the adversarial network schematic

In Figure 2, generator G and discriminator D are randomly initialized. A batch of real samples are sampled from the real data distribution, and a batch of noisy samples are sampled from the noisy distribution, which are input into G to obtain the generated samples. Real samples and generated samples are input into D separately, and the output of D is calculated. The parameters of D are kept unchanged, while the parameters of G are updated to provide samples that are closer to real data, thereby deceiving D [12-13]. G is repeatedly sampled and trained until it can generate high-quality data, and D cannot accurately distinguish between real data and generated data. Assuming the generative network is g and the DM is d, the image reconstructed by the generative network $G(z)$'s expression is shown in equation (3).

$$G(z) = \arg\min_{g} \max_{d} v(g,d) \quad (3)$$

In equation (3), z represents the signal input to the generative network. The default choice for determining the network's benefits v is shown in equation (4).

$$v(\theta^{(g)};\theta^{(d)}) = E_{X\sim P_{data}} \log d(x) + E_{X\sim P_{data}} \log(1-d(x)) \quad (4)$$

In equation (4), x is the real sample; the probability of the DM judging whether the image is true is represented by the symbol $d(x)$. Super Resolution GAN (SRGAN) is a method that uses convolutional neural networks to achieve Super Resolution (SR) for a single image [14]. The perceptual loss function of SRGAN l^{SR} is expressed as equation (5).

$$l^{SR} = l_X{}^{SR} + 10^{-3} l_{Gen}{}^{SR} \quad (5)$$

In equation (5), $l_X{}^{SR}$ is content loss; $l_{Gen}{}^{SR}$ is adversarial loss. SRGAN uses VGG loss, which includes VGG16Q and VGG19. Among them, VGG19 has 19 layers, including 16 convolutional layers and the last 3 fully connected layers. The middle is similar to usual, using a pooling layer and finally passing through softmax [15]. The breakdown diagram of VGG19 network structure is shown in Figure 3.

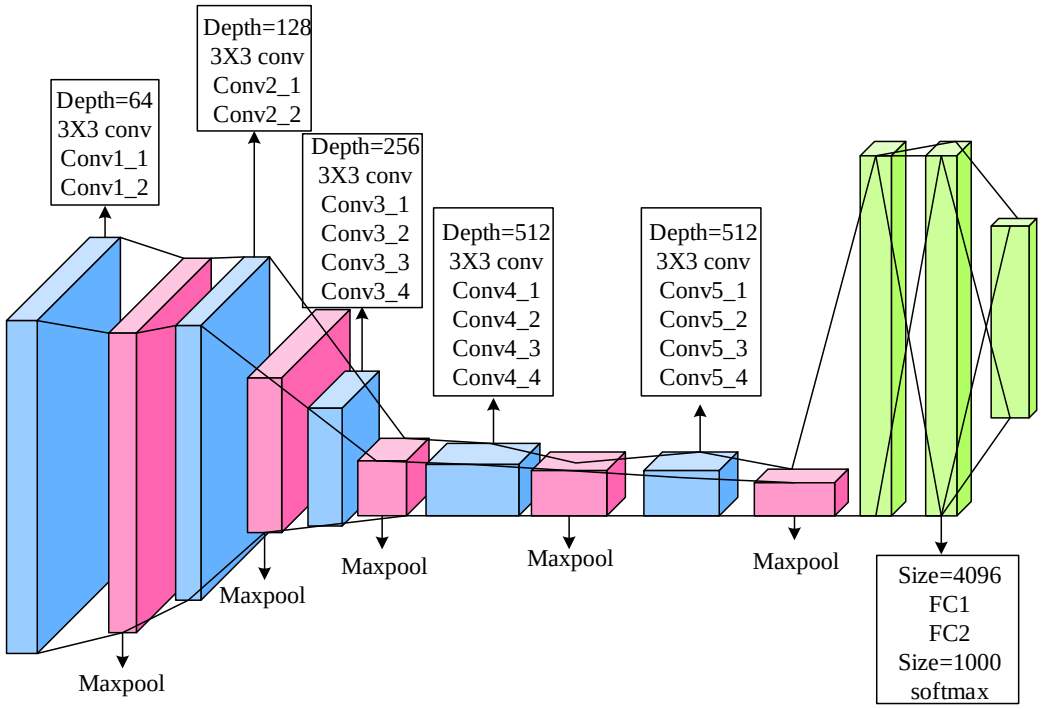


Figure 3: VGG19 Network structure decomposition diagram

In Figure 3, VGG19 consists of multiple convolutional blocks and fully connected layers. The former contains multiple convolutional layers and ReLU activation layers, and after each convolutional block is completed, a max pooling layer is used to decrease the size of the feature map [16]. VGG19 replaces large convolution kernels by stacking multiple 3×3 small convolution kernels, which can increase the depth of the network while maintaining the same receptive field, thereby improving the network's feature extraction ability. Equation (6) shows the expression for the VGG19 loss function.

$$l_{VGG/i,j}^{SR} = \frac{1}{W_{i,j}H_{i,j}}\sum_{x=1}^{W_{i,j}}\sum_{y=1}^{H_{i,j}}(\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta G}(I^{HR}))_{x,y})^2 \quad (6)$$

In equation (6), $W_{i,j}$ is the dimension of one feature map in the model; $H_{i,j}$ is the dimension of another feature map in the model; $\phi_{i,j}$ is the feature map of the j-th convolutional layer in VGG19. For adversarial loss, SRGAN mainly indirectly improves the quality of the network generated images by adding discriminators. The expression for adversarial loss l_{Gen}^{SR} is shown in equation (7).

$$l_{Gen}^{SR} = \sum_{n=1}^{N} -\log D_{\theta D}(D_{\theta G}(I^{LR})) \quad (7)$$

2.2 Lightweight SRIR Method with SRLRGAN-SN

To achieve precise control over the parameters of the generative and DM while ensuring training stability, this study introduces the SN and SRLR relative discriminators based on SRGAN and proposes the SRLRGAN-SN image reconstruction method. Figure 4 illustrates the flowchart.

In Figure 4, SRLRGAN-SN first receives LRI as input and performs necessary preprocessing operations. Then, the LRI is mapped to a high-resolution space through a generative network to generate the corresponding SR image. Then, the SRLR relative discriminator is applied to discriminate the relative difference, and the evaluation of the relative realism is output. Then, content loss, feature loss, texture loss, and SRLR relative discriminative adversarial loss are calculated. Finally, the above process is repeated until the preset training epochs are reached or convergence conditions are met. To simplify the complexity of the model problem, this study uses the SRLR loss function as the criterion for discrimination, transforming the absolute discrimination problem between generated images and real samples into relative difference discrimination. The discriminator loss function based on SRLR is shown in equation (8).

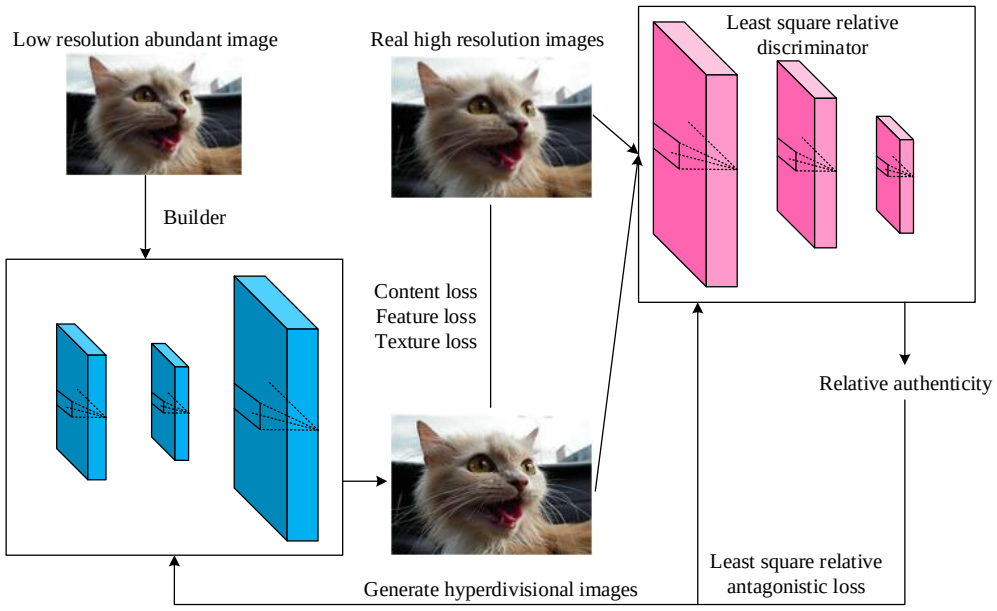


Figure 4: SRLRGAN-SN process

$$\min_D V_{LSGAN}(D) = \frac{1}{2} E_{x\sim Pdata(x)}\left[(D(x)-b)^2\right] + E_{z\sim Pg(z)}\left[(D(G(z))-a)^2\right] \quad (8)$$

In equation (8), a represents the generated sample label; b represents the true sample label; $x \sim p_{data}(x)$ represents the probability distribution of the true sample; $z \sim p_g(z)$ represents the probability distribution that the generated sample follows; $E_{x\sim Pdata(x)}$ represents the expected value of the true sample distribution; $E_{z\sim Pg(z)}$ represents the expected value of the generated sample distribution. The loss function of the generator is shown in equation (9).

$$\min_G V_{LSGAN}(G) = \frac{1}{2} E_{z\sim Pg(z)}\left[(D(G(z))-c)^2\right] \quad (9)$$

In equation (9), c represents the value that the GM wishes to discriminate as true. The calculation formula for the JS divergence optimized by the original GAN is shown in equation (10).

$$JS(P_r \| P_f) = \log 2 + \max\left\{\frac{1}{2} E_{x_r\sim P_r}\left[\log(D((x_r)) + \frac{1}{2} E_{x_f\sim P_f}\left[\log(1-D((x_f))\right]\right\} \quad (10)$$

In equation (10), P_r represents the true distribution; P_f represents the generated distribution; x_r is the true sample; x_f is the generated sample. The expression for the discriminative mechanism of the original GAN is shown in equation (11).

$$D_{GAN}(x) = \sigma(C(x)) \quad (11)$$

In equation (10), C is the output of the DM before activation; σ is the sigmoid activation function. The discriminant mechanism of the original GAN is realized through the discriminator constantly learning and distinguishing between the real data and the generated data, as well as the adversarial training between the discriminator and the generator. This mechanism enables the GAN to generate realistic data samples and show strong ability in various generation tasks. The expression for the discriminative mechanism of the original RGAN is shown in equation (11).

$$D_{RGAN}(x_r, x_f) = \sigma(C(x_r) - C(x_f)) \quad (12)$$

The discriminant mechanism of the original RGAN reduces the randomness of the gradient and does not increase the time complexity of the algorithm by averaging the original output of the discriminator of a set of reference samples and taking the mean value as the reference. The loss function of the RGAN discriminator is shown in equation (13).

$$L_D^{RGAN} = -E_{(x_r,x_f)\sim(P,Q)}\left[\log(\sigma(C(x_r)-(C(x_f)))\right] \quad (13)$$

The loss function of the RGAN generator is shown in

equation (14).

$$L_G^{RGAN} = -E_{(x_r, x_f)\sim(P,Q)}\left[\log(\sigma(C(x_f) - (C(x_r)))\right] \quad (14)$$

The relative discriminator considers both real and generated samples when evaluating samples, and makes decisions by comparing their relative authenticity [17]. The relative discriminator not only compensates for the shortcomings of the original GAN in training stability and generating sample diversity, but also promotes a more balanced and dynamic game between G and D, which may achieve higher quality generation results. The DM structure of SRLRGAN-SN is shown in Figure 5.

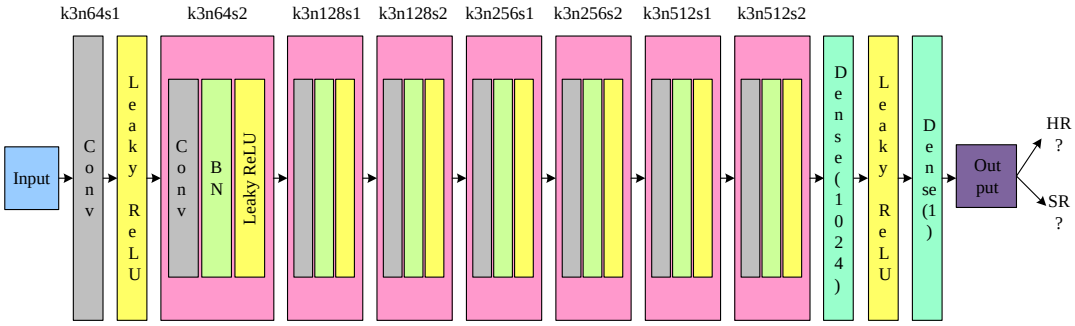


Figure 5: The discriminant network structure of SRLRGAN-SN

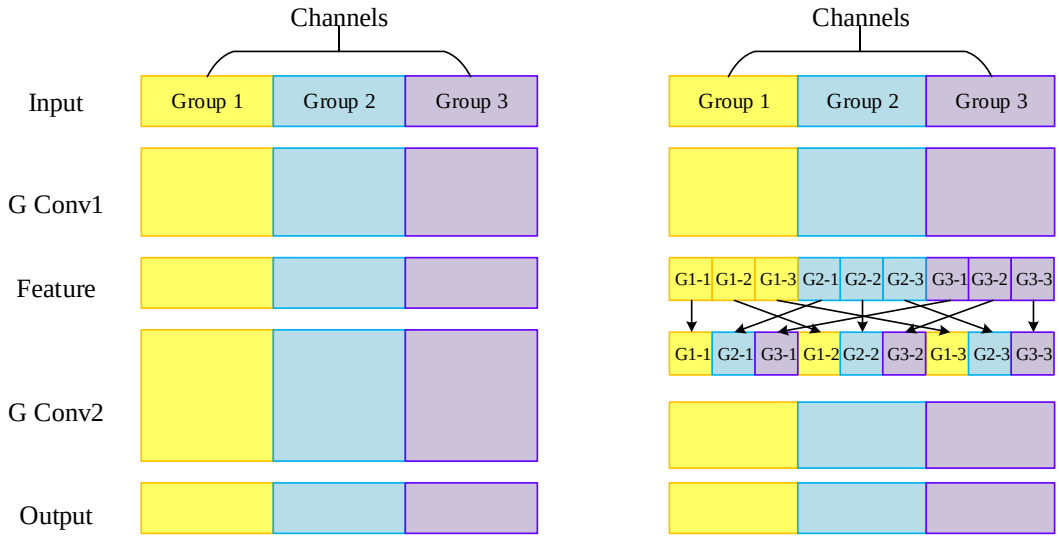
In Figure 5, the front-end of the DM consists of 8 convolutional layers, each of which extracts features at different levels from the input image. The features are gradually abstracted from low to high levels. After each convolutional layer, an SN layer is introduced to control the SN of the network weights, thereby satisfying the Lipschitz constraint. The SN layer forms a soft constraint by limiting the maximum singular value of the weight matrix to ensure the Lipschitz continuity of the network. Lipschitz constraint can ensure that the output change of the network does not exceed a certain multiple of the input change, which helps to stabilize the training process and prevent gradient explosion and gradient disappearance.

All activation layers use the Leaky ReLU (Leaky Rectified Linear Unit) activation function. After the convolutional layer and SN layer, the DM connects two linear dimensionality reduction layers. The final output layer of the discrimination network adopts the SRLR relative discrimination mechanism, which makes discrimination by calculating the relative degree of realism. The SRLR relative discrimination output layer calculates the distance in the feature space, and uses it as the discrimination basis to generate high-quality images.

Assuming the output of layer l in the network is x^l, its expression is shown in equation (15).

$$x^l = f^l(W^l - x^{l-1} + b^l) \quad (15)$$

In equation (15), x^{l-1} represents the input of layer l in the network; W^l represents the SN of layer l in the network; b^l represents the bias of layer l in the network. ShuffleNet is a lightweight network, and its latest version ShuffleNetV uses point by point group convolution and channel shuffling to reduce computational complexity while improving model accuracy. Therefore, this study introduces ShuffleNetV into SRLRGAN-SN. The schematic diagram of ShuffleNetV's grouped convolution and channel shuffling is shown in Figure 6.



(a) Block convolution diagram (b) Channel mixing diagram

Figure 6: Schematic diagram of packet convolution and channel mixing

In Figure 6 , in grouped convolution, the input channel is divided into several groups, and the feature maps of each group are only convolved with the corresponding convolution kernels, significantly reducing computational complexity and model parameter count. In Figure 6 (b), channel shuffling shuffles and reallocates feature map channels from different groups, allowing each group to receive information from all original input channels in subsequent operations. ShuffleNetV achieves a dual improvement in model efficiency and performance by introducing grouped convolution and channel shuffling techniques.

3 Results

3.1 Performance Analysis of SRLRGAN-SN

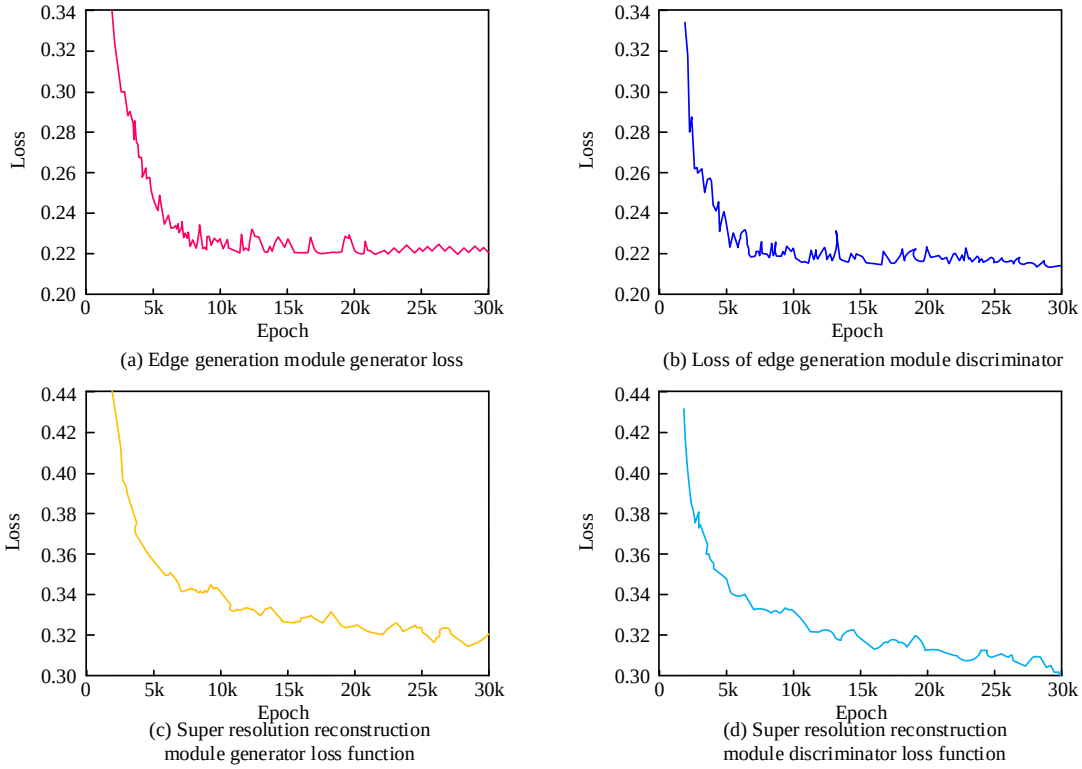
To test the effectiveness of the reconstruction model, this study conducted simulation experiments. The first stage was to train the edge generator. In the second stage, the trained edge generator was used as a guiding signal to drive the training process of the SR reconstruction module. At each training stage, the Adam optimization algorithm was used to efficiently and stably adjust the model parameters. The variation curves of the edge generation module and super-resolution reconstruction loss function are shown in Figure 7.

In Figure 7 (a), as the iterations increased, a steady trend was observed when the loss value approached 0.22. In Figure 7 (b), as the iterations were 25000, the discriminator loss curve of the edge generation module tended to stabilize. In Figure 7 (c), the loss function curve of the SR reconstruction module generator tended to stabilize at 0.32. In Figure 7 (d), as the iterations increased, the loss curve gradually decreased and stabilized when the loss value approached 0.30, indicating that the discriminator's ability to distinguish between real HRI and generated HRI had stabilized. To verify the effectiveness of SRLRGAN-SN in the reconstruction process, BRISQUE was used as an evaluation metric, and VDSR and SRCNN were introduced as comparative algorithms for testing on the Set5 dataset. Figure 8 shows the experimental results.

In Table 2, this study implemented the SRLRGAN-SN algorithm on the computer Python programming language.

Table 2: Experimental environment setting table

Simulation environment	Parameter value
Operating system	Ubuntu 20.04 LTS
Python	Python 3.8
CPU	Intel(R)Core(TM) i7-9700k@3.60GHz
GPU	NVIDIA GeForce RTX 3090
DL framework	TensorFlow 2.x
Internal memory	32G



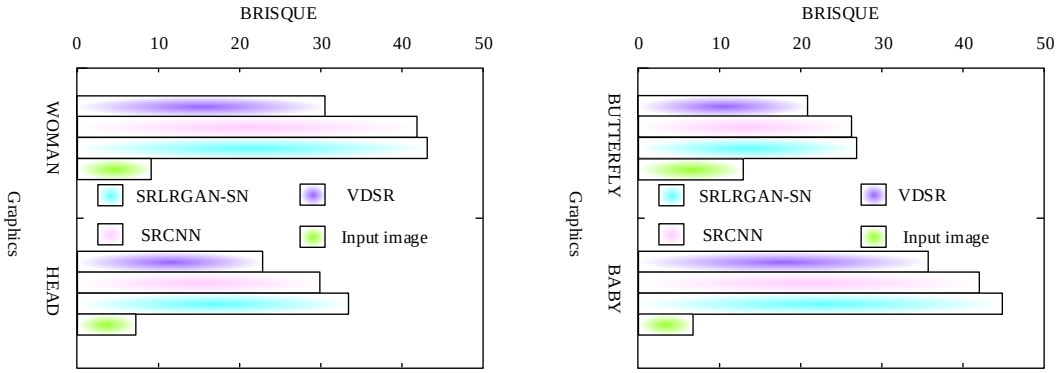
(a) Edge generation module generator loss

(b) Loss of edge generation module discriminator

(c) Super resolution reconstruction module generator loss function

(d) Super resolution reconstruction module discriminator loss function

Figure 7: Edge Generation module and superresolution reconstruction loss function change curve



(a) Economic cost objective function diagram

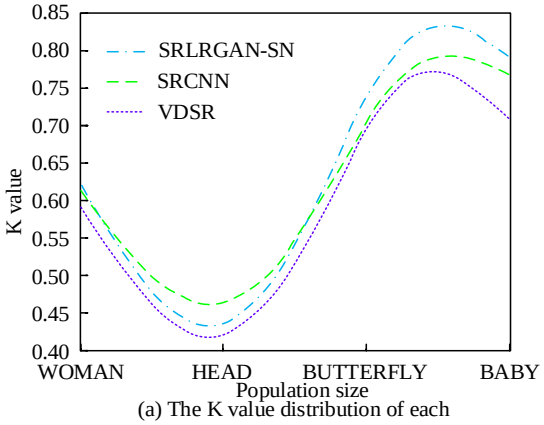
(b) Economic cost objective function diagram

Figure 8: BRISQUE evaluation index of three algorithms on Set5

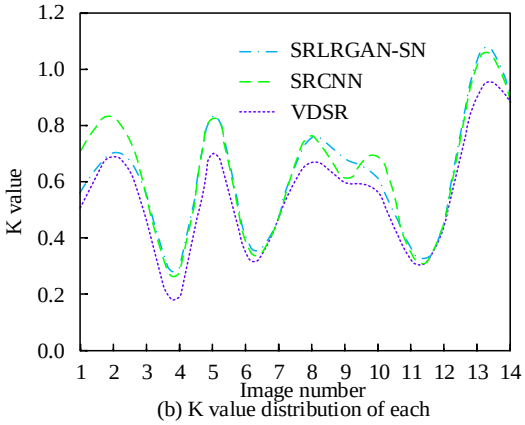
In Figure 8 (a), SRLRGAN-SN scored 43.6 for WOMAN images and 33.8 for HEAD images. In Figure 8 (b), on the BUTTERFLY image, the evaluation score of VDSR was the lowest at 21.2, followed by SRCNN algorithm at 26.5, and SRLRGAN-SN with a score of 27.9 was slightly higher than SRCNN, indicating its advantage in processing detail rich images. For BABY images, the performance of each algorithm had been improved, with VDSR score of 35.7, SRCNN score of 43.2, and SRLRGAN-SN score of 46.3. From this, SRLRGAN-SN performed excellently in the evaluation of various image types, with performance far exceeding traditional VDSR and SRCNN. This study conducted color cast factor analysis using three algorithms, VDSR, SRCNN, and SRLRGAN-SN, with Set5 and Set14 as reference datasets at a magnification of 2. For the images in the Set14 dataset, this study numbered them alphabetically one by one. Figure 9 shows the distribution of K values for each reconstructed image in different datasets.

In Figure 9 (a), the K values of the reconstructed samples within Set5 were mainly concentrated in the interval of 0.4008.5, indicating the similarity level between the reconstructed image and the original image in terms of color reproduction. In Figure 9 (b), the K values in Set14 were roughly distributed in the range of 0.20 to 1.10. Among them, SRLRGAN-SN had a lower overall color distortion level on both Set5 and Set14, indicating that it had better performance in color preservation and reconstruction quality compared to other algorithms. In summary, the K-value distribution of the reconstructed samples in Set5 and Set14

was within an acceptable color shift threshold range. In order to ensure the robustness of SRLRGAN-SN algorithm, the above three algorithms are compared with the current popular SOTA algorithm on Set5 and Set14, DIV2K and Urban100 data sets. Quantitative indicators PSNR and SSIM were used to evaluate image quality, and FLOPs was used to analyze computational efficiency. The experimental results are shown in Table 3.



(a) The K value distribution of each reconstructed image in the Set5 dataset



(b) K value distribution of each reconstructed image in Set14 dataset

Figure 9: K Value distribution of reconstructed images in different data sets

Table 3: Comparison of results of different algorithms

Algorithm	Set5			Set14			DIV2K			Urban100		
	PSNR	SSIM	FLOPs	PSNR	SSIM	FLOPs	PSNR	SSIM	FLOPs	PSNR	SSIM	FLOPs
VDSR	30.14	0.92	66543.25	28.32	0.86	132456.78	26.78	0.81	264913.50	24.56	0.76	529827.00
SRCNN	29.56	0.89	5321.34	27.98	0.85	10542.12	26.34	0.79	10684.28	24.12	0.74	21084.56
SOTA	31.23	0.91	123456.89	29.45	0.87	246910.56	27.12	0.82	493827.56	25.03	0.78	987654.32
SRLRGAN-SN	31.50	0.92	234567.11	29.72	0.88	469134.23	27.30	0.83	938271.43	25.28	0.79	1938271.65

As can be seen from Table 3, the PSNR of VDSR algorithm on Set5 test set is 30.14dB, SSIM is 0.901 and FLOPs is 66543.25 times, while the performance on Set14 test set is slightly lower. The performance of SRCNN algorithm is lower than that of VDSR on both test sets. As the current optimal algorithm, SOTA algorithm outperforms the other three algorithms on the two test sets, has the highest PSNR and SSIM values, but also has the largest computational load. Although SRLRGAN-SN is computatively more expensive than SOTA on Set5 and Set14 test sets, PSNR and SSIM are higher than SOTA, so they can produce more natural images.

3.2 The ablation experiment of SRLRGAN-SN

To scientifically evaluate the advantages of each combination module in the SRLRGAN-SN method in SRIR performance, this study compared the model performance under different module combinations to quantify the independent and collaborative effects of each module in improving image reconstruction quality and optimizing visual perception characteristics. Table 4 shows the ablation experiment setup.

In Table 4, the model with Setting 1 performed the worst in all indicators, indicating the limitations of basic training configuration on model performance. When feature loss was introduced in Setting 2, the model demonstrated a notable enhancement in performance, with the PI value dropping to 3.25. This improvement was significant, indicating the effectiveness of feature loss in optimizing the output quality of the model. Setting 3 reduced the PI index to 3.28 by incorporating texture loss, indicating that the introduction of texture loss effectively improved the visual perception quality of the generated image. In Setting 4 and Setting 5, the strategy of SRLR relative adversarial loss demonstrated superior performance in improving visual perception quality compared to traditional relative adversarial loss. Setting 6 had the highest performance metrics among all Setting methods, indicating that the research approach could generate high-quality images. Figure 10 shows the texture detail comparison image of the ablation experiment reconstruction.

Table 4: Analysis and comparison of ablation experiments

Settings	Content loss	Feature loss	Texture loss	Relative counteraction loss	Least square relative antagonistic loss	SN	PI	P	LPIPS
(a)	○	×	×	×	×	×	5.36	>0.05	0.25
(b)	○	○	×	×	×	×	3.25	<0.01	0.18
(c)	○	○	○	×	×	×	3.18	<0.05	0.15
(d)	○	○	○	○	×	×	2.46	<0.05	0.12
(e)	○	○	○	×	○	×	2.24	<0.05	0.10
(f)	○	○	○	×	○	○	2.11	<0.05	0.08



Figure 10: Comparison of texture details reconstructed through ablation experiments

In Figure (10), the texture visual effect of the image generated by Setting 1 was the blurriest, indicating that only basic training had not fully optimized the texture features. Setting 2 introduced a feature loss term, which improved the visual effect of the image to a certain extent, but the texture details of the image were still not clear enough. Setting 3 introduces texture loss, which enhanced the texture detail processing effect of the image. However, the perceptual visual effect of the image was poor, indicating that although the introduction of a single texture loss helped with detail restoration, it was not sufficient to comprehensively optimize the perceptual performance of the image. Setting 4 adopted GAN adversarial loss, which significantly improved the perceptual quality of the image. The overall perception of the image was more natural and detailed, indicating that GAN adversarial loss had significant advantages in improving the realism and detail expression of image generation. Setting 5 adopted the SRLR relative adversarial loss proposed in the study to replace GAN adversarial loss, which further improved the perceptual visual effect of the image, indicating that the proposed SRLR relative adversarial loss was more effective in optimizing the quality of image generation. Based on Setting 5, Setting 6 combined SN regularization to generate images with extremely fine texture and highly similar perceptual features to real HRI. This further validated the positive role of each combination module in improving the quality of SR reconstruction image generation in the research method. For example, lightweight image super-resolution reconstruction methods based on multi-frequency features and texture enhancement, or lightweight image super-resolution reconstruction methods based on interleaved group convolution and sparse global attention, can be explored to solve the performance bottleneck of current algorithms when processing large-scale data sets.

4 Discussion and conclusion

4.1 Discussion

SRIR technology has always been a highly valued and focused hot topic in computer vision, with significant research value in healthcare, remote sensing, and surveillance. In recent years, with the rise of DL, significant

progress has been made in GAN based SRIR. To this end, research has improved GAN and proposed an image reconstruction algorithm with SRLRGAN-SN algorithm.

Firstly, in the SR reconstruction module, the super resolution reconstruction module, the generator loss curve tends to be stable at 0.32, and the discriminator loss curve tends to be stable at about 0.30, indicating that the discriminator has a strong ability in distinguishing real and generated high-resolution images. This result was consistent with the conclusion of scholar Almansour in the study of SR reconstruction of upper abdominal magnetic resonance imaging based on DL [18]. Secondly, the experimental results of VDSR, SRCNN, and SRLRGAN-SN algorithms on the Set5 dataset were analyzed using the BRISQUE evaluation index. It was found that SRLRGAN-SN had better evaluation scores than the other two algorithms on various image types. This result was consistent with the conclusion of the Orii team in the study of coronary artery computed tomography angiography SRDL reconstruction to evaluate the lumen of coronary arteries and stents [19]. Then, the above algorithm was reconstructed on Set5 and Set14 datasets, where SRLRGAN-SN's K value fell between 0.43 and 0.85 on Set5 and between 0.28 and 1.10 on Set14. Compared with the current best technology SOTA, SRLRGAN-SN is not only more accurate in color preservation, but also exceeds the SOTA method in image detail reconstruction and overall visual effect. This result further proves the leading position of SRLRGAN-SN algorithm in the field of image reconstruction, and provides strong support for future related research and practice. Finally, the study employed ablation experiments to further reveal the impact of different settings on model performance. The basic training configuration set Group a only adds content loss, and the resulting images may not be rich enough in detail and texture and visually realistic enough. Set group b to introduce feature loss on top of a, while feature loss can improve detail, it cannot generate complex textures and structures. Set group c introduces texture loss, which produces more realistic texture effects, but artifacts still exist when dealing with complex textures. Set group d introduced RaGAN counterloss, which further improved the realism of the generated image and reduced artifacts, but it was unstable in the training process. Setting group e introduces the least squares relative adversarial loss, which combines the least squares loss and relativistic adversarial loss to train the generated adversarial network more stably. The least squares relative adversarial loss function does not apply to all types of data sets, and may not perform as well as other loss functions on some specific data sets. Setting group f combined with spectral norm regularization to further stabilize the training process, reduce the generation of artifacts, and generate images that are highly close to real high-resolution images in texture details and perceptual features. Elshafai et al. also proposed similar conclusions in their review of single image super-resolution models for medical images with DL [20].

In summary, SRIR with SRLRGAN-SN has shown excellent performance in image quality improvement, color preservation, and computational efficiency, and has broad application prospects and enormous research value.

4.2 Conclusion

This study proposes an SRLRGAN-SN based on GAN, which effectively improves the quality of SRIR by introducing multiple loss functions and optimization strategies. The collaborative work of its edge generation module and SR reconstruction module enables the algorithm to restore rich texture details while maintaining image edge clarity. In terms of reconstruction quality and color preservation, SRLRGAN-SN had shown significant advantages compared to traditional VDSR and SRCNN algorithms. By comparing the BRISQUE evaluation indicators, SRLRGAN-SN could maintain high evaluation scores when processing different types of images, especially when processing images with rich details. In addition, by analyzing the K-value distribution of the reconstructed images in Set5 and Set14, the superiority of SRLRGAN-SN in color preservation and reconstruction quality was further verified. Its lower overall color distortion level indicated that the algorithm had high accuracy in color reproduction. Finally, the positive effect of SRLRGAN-SN algorithm on improving image generation quality was verified through ablation experiments. The limitation of this study is that the running time of the algorithm may become one of the limiting factors for its practical application when processing large-scale datasets or HRI. Therefore, future research can focus on lightweight design and acceleration techniques for algorithms to improve their computational efficiency and practicality.

References

[1] Y. Mengbei, W. Hongjuan, L. Mengyang (2021) Overview of Research on Image Super-Resolution Reconstruction, Proceedings of the IEEE International Conference on Information Communication and Software Engineering, IEEE, Location, pp. 131–135. https://doi.org/10.1109/ICICSE52190.2021.9404113

[2] X. Li, Z. Chen (2022) Single image super-resolution reconstruction based on fusion of internal and external features, Multimedia Tools and Applications, Springer, Location, pp. 1589–1605. https://doi.org/10.1007/s11042-021-11400-2

[3] D. Qiu, L. Zheng, J. Zhu (2021) Multiple improved residual networks for medical image super-resolution, Future Generation Computer Systems, Elsevier, Location, pp. 200–208. https://doi.org/10.1016/j.future.2020.11.001

[4] S. Wang, C. Qiao, A. Jiang (2021) Instant multicolor super-resolution microscopy, Biophysics Reports, Springer, Location, pp. 304–312.

[5] Sunkari, Venkateswarlu, Srinagesh, A. (2024) DeepExplain: Enhancing DeepFake Detection Through Transparent and Explainable AI model, Informatica (Slovenia), Slovenian Society Informatika,

Ljubljana, pp. 103–110. https://doi.org/10.31449/inf.v48i8.5792

[6] L. An, F. Dai, L. An (2021) Image Super-Resolution Reconstruction Technology Based on Deep Learning, Journal of Advances in Artificial Life Robotics, Springer, Location, pp. 163–167. https://doi.org/10.57417/jaalr.1.4_163

[7] M. Gao, W. Song, G. Zhang (2023) CT image super-resolution reconstruction via Pixel-Attention Feedback Network, International Journal of Biomedical Engineering and Technology, Inderscience, Location, pp. 21–33. https://doi.org/10.1504/IJBET.2023.131697

[8] Y. Nagayama, T. Emoto, H. Hayashi (2023) Coronary Stent Evaluation by CTA: Image Quality Comparison Between Super-Resolution Deep-Learning Reconstruction and Other Reconstruction Algorithms, American Journal of Roentgenology, American Roentgen Ray Society, Location, pp. 599–610. https://doi.org/10.2214/AJR.23.29506

[9] Li, Min (2024) Application of GAN-Based Data Encryption Technology in Computer Communication System, Informatica (Slovenia), Slovenian Society Informatika, Ljubljana, pp. 17–34. https://doi.org/10.31449/inf.v48i15.6390

[10] S. Pal, A. Roy, P. Shivakumara, U. Pal (2023) Adapting a Swin Transformer for License Plate Number and Text Detection in Drone Images, Artificial Intelligence and Applications, Springer, Location, pp. 145–154. https://doi.org/10.47852/bonviewaia3202549

[11] Y. Wang, Y. Luo, J. Ren (2022) Research on application of spatial attention mechanism in the super-resolution reconstruction of single-channel greyscale image of mouse brain, International Journal of Wireless and Mobile Computing, Inderscience, Location, pp. 259–264. https://doi.org/10.1504/ijwmc.2022.10049459

[12] L. Ma (2021) Face Recognition of Intelligent Building based on Super-Resolution Reconstruction of Visual Imag, Proceedings of the International Conference on Intelligent Computing and Control Systems, IEEE, Location, pp. 895–900. https://doi.org/10.1109/iciccs51141.2021.9432282

[13] W. Xin, Z. Shufen, L. Yuanyuan (2023) Pixel attention convolutional network for image super-resolution, Neural Computing & Applications, Springer, Location, pp. 8589–8599. https://doi.org/10.1007/s00521-022-08132-1

[14] C. Wilpert, C. Neubauer, A. Rau, T. W. HannahBenkert (2023) Accelerated Diffusion-Weighted Imaging in 3 T Breast MRI Using a Deep Learning Reconstruction Algorithm With Superresolution Processing: A Prospective Comparative Study, Investigative Radiology, Lippincott Williams & Wilkins, Location, pp. 842–852. https://doi.org/10.1097/rli.0000000000000997

[15] Yan, Xu (2024) A Face Recognition Method for Sports Video Based on Feature Fusion and Residual Recurrent Neural Network, Informatica (Slovenia), Slovenian Society Informatika, Ljubljana, pp. 137–152. https://doi.org/10.31449/inf.v48i12.5968

[16] B. Xie, F. Niu (2021) Super-resolution reconstruction algorithm for aerial image data management based on deep learning, Distributed and Parallel Databases, Springer, Location, pp. 77–94. https://doi.org/10.1007/s10619-021-07356-9

[17] X. Yang, D. Zhang, Z. Wang (2022) Super-resolution reconstruction of terahertz images based on a deep-learning network with a residual channel attention mechanism, Applied Optics, Optical Society of America, Location, pp. 3363–3370. https://doi.org/10.1364/AO.452511

[18] H. Almansour (2021) Deep Learning-Based Superresolution Reconstruction for Upper Abdominal Magnetic Resonance Imaging An Analysis of Image Quality, Diagnostic Confidence, and Lesion Conspicuity, Investigative Radiology, Lippincott Williams & Wilkins, Location, pp. 509–516. https://doi.org/10.1097/rli.0000000000000769

[19] M. Orii, M. Sone, T. Osaki (2023) Super-resolution deep learning reconstruction at coronary computed tomography angiography to evaluate the coronary arteries and in-stent lumen: an initial experience, BMC Medical Imaging, BioMed Central, Location, pp. 1–10. https://doi.org/10.1186/s12880-023-01139-7

[20] W. Elshafai, A. M. Ali, S. A. Elnabi (2024) Single image super-resolution approaches in medical images based deep learning: a survey, Multimedia Tools and Applications, Springer, Location, pp. 30467–30503. https://doi.org/10.1007/s11042-023-16197-w

Naïve Bayes-Based Freshwater Fish Classification Using HSV Color Features

Hindayati Mustafidah*[1] and Suwarsito[2]
[1]Informatics Engineering, Universitas Muhammadiyah Purwokerto, Jl. KH. Ahmad Dahlan Purwokerto, Central Java, Indonesia
[2]Aquaculture, Universitas Muhammadiyah Purwokerto, Jl. KH. Ahmad Dahlan Purwokerto, Central Java, Indonesia
E-mail: h.mustafidah@ump.ac.id, suwarsito@ump.ac.id
*Corresponding author

Keywords: freshwater fish classification, HSV color feature extraction, image processing techniques, machine learning, naïve bayes

Received: January 15, 2025

This study presents a machine learning-based method for classifying six freshwater fish species commonly consumed in Indonesia: gourami (gurame), catfish (lele), tilapia (nila), barb (melem), Java barb (tawes), and pomfret (bawal). A total of 132 images, with 22 images per species, were collected from online sources and direct field photography. The classification model utilizes a Naïve Bayes algorithm, employing color feature extraction based on the Hue, Saturation, and Value (HSV) color space. The HSV method decomposes image color into three components—Hue (the color type, such as red, blue, or green), Saturation (the intensity or vividness of the color), and Value (the brightness or lightness of the color)—allowing for improved distinction between morphologically similar species, such as barb and tilapia. Image preprocessing included resizing, background removal, and conversion from RGB to grayscale prior to HSV feature extraction. The dataset was split into training and testing subsets, with 20% of the data allocated for testing. The model's performance was evaluated using a confusion matrix, and it achieved a classification accuracy of 79.17%. This result surpasses the accuracy reported in comparable studies, such as one on frozen tuna classification, which achieved 72.73% using similar techniques. The findings validate the effectiveness of the Naïve Bayes classifier for species identification tasks in fisheries. Moreover, the approach offers a computationally efficient solution suitable for environments with constrained data availability and limited computational resources. This study underscores the practical value of machine learning in aquaculture, highlighting its potential for enhancing species monitoring, quality control, and automated recognition using relatively small datasets.

Povzetek: Opisana je uporaba HSV barvnih značilk z naivnim Bayesom, klasifikatorjem, ki omogoča bolj kvalitetno prepoznavo šestih vrst sladkovodnih rib pri majhnih podatkovnih zbirkah.

1 Introduction

Indonesia is known for its rich aquatic biodiversity, with numerous fish species found in its freshwater and marine ecosystems. Among these, several freshwater species, such as gourami (*gurami*), catfish (*lele*), tilapia (*nila*), barb (*melem*), Java barb (*tawes*), and pomfret (*bawal*), are widely cultivated and consumed. However, distinguishing between these species visually can be difficult, especially for non-experts. This difficulty is because many fish share similar physical traits, including the shape of the head, body, tail, and even their coloration. For example, barb and tilapia have nearly identical body shapes and overlapping color patterns, making it hard to tell them apart. This challenge is particularly noticeable when the fish species have similar sizes and body colors, which increases the chances of misidentification. Without the help of more advanced techniques, such as machine learning, visual identification can often lead to errors.

Previous studies have highlighted the challenges of distinguishing between these species based on visual appearance. Ref. [1] notes that tilapia exhibit a wide range of morphological variations, making it difficult to differentiate between tilapia species based solely on morphology. Similarly, [2] reported that tilapia populations from Lake Tempe and Lake Sidenreng in South Sulawesi displayed significant morphometric and meristic differences, further complicating species identification based on external features. Additionally, [3] found that Nile tilapia populations in Uganda showed distinct morphometric differences, emphasizing the

difficulties in species identification, even when fish are of the same genus.

In addition to tilapia, the barb species share similar body shapes and coloration, making visual differentiation challenging. Ref. [4] discusses the morphological similarities between barb species and other fish, including tilapia, further demonstrating the need for advanced techniques to identify species in such cases accurately. This difficulty in identification highlights the importance of using machine learning-based methods, such as the Naïve Bayes classifier, to automate the identification process and reduce human error in distinguishing between visually similar fish species.

2 Related works

Various studies and applications of artificial intelligence have been conducted across diverse fields of technology development, addressing numerous real-world problems in the digital era, including the fisheries sector, such as expert systems [5]. In addition to expert systems, identification of fish objects can be done using machine learning (ML) as conducted by [6] and [7], including salinity as a factor that affects fish survival [8]. ML is software that can learn to perform tasks [9]. Identification of fish based on their images using algorithms in machine learning has been carried out by several researchers, including [10] using the Support Vector Machine (SVM) to classify fish in Bangladesh, Ref. [11] does fish freshness classification based on eye color using the K-Nearest Neighbor (K-NN). Identification of formalin fish using K-NN and GLCM was conducted by [12] and using a multilayer perceptron network by [13]. Ref. [14] identified fish types using Convolution Neural Networks (CNN). Identification of betta fish using SVM was carried out by [15] and [16] using GLCM and KNN. At the same time, [17] used the Principal Component Analysis (PCA) and K-Nearest Neighbors (KNN) algorithms, which were also conducted by [18]. The combination of KNN and PCA was carried out by [19]. Next, [20] classified the types of marine fish using the SVM method with HOG and HSV features. Using SVM, African Cichlid ornamental fish types were classified [21] and SVM and CNN were used to establish the classification models based on the LIBS and Raman spectroscopy [22]. The following summarizes several machine learning methods for fish image classification utilizing various datasets, feature extraction techniques, and classification accuracies presented in Table 1.

Table 1: Summary of several machine learning methods for fish image classification.

Study	Dataset	Classification Method	Feature Extraction	Accuracy (%)
[10]	Fish images from Bangladesh	SVM	Hybrid features	80.5
[11]	Fish freshness dataset	K-NN	Eye color	75.2
[12]	Fish images (formalin detection)	K-NN & GLCM	GLCM	71.8
[14]	Fish images	CNN	CNN-based features	85.0
[21]	African Cichlid fish images	SVM	HOG & HSV features	90.1
[23]	Frozen tuna images	Naïve Bayes	HSV features	72.7

Machine learning algorithms have been widely used for various classification tasks, including fish species identification. Techniques such as Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Naïve Bayes have all been explored in this domain. Among these, Naïve Bayes has gained significant attention due to its simplicity and efficiency, particularly when working with limited datasets. While more complex algorithms like CNNs and SVMs may outperform Naïve Bayes when dealing with larger datasets or more intricate models, Naïve Bayes offers a practical and efficient solution when the goal is to classify species using moderate datasets, especially in environments with resource constraints.

While methods like Convolutional Neural Networks (CNNs) and Support Vector Machines (SVMs) have shown excellent results in image classification tasks (as in Table 1), their implementation often comes with considerable computational overhead. For instance, CNNs require large datasets and substantial computational resources for practical training, making them less ideal when labeled data is scarce. Furthermore, CNNs involve automatic feature extraction, which may not be optimal for simpler datasets or when the objective is to use a lightweight model suitable for real-time applications. Similarly, SVMs, though powerful, tend to struggle with small datasets and high-dimensional feature sets. They require complex kernel functions and can suffer from

overfitting in such cases, in addition to demanding significant computational resources during training and prediction.

In contrast, the Naïve Bayes algorithm, which is based on a probabilistic framework, offers several advantages. It assumes feature independence, simplifies the model, and reduces computational complexity, making it especially useful when extracting features based on straightforward descriptors, like color histograms (such as HSV). Additionally, Naïve Bayes requires fewer training images, making it suitable for environments with limited labeled data. Its speed and efficiency in training and prediction are crucial in practical applications, particularly in systems requiring real-time or near-real-time classification, such as automated fish classification in aquaculture.

Ref. [24] demonstrated using Naïve Bayes for document classification, showcasing its effectiveness in assigning categories to documents based on specific attributes. Although the context was textual data, the principle of Naïve Bayes as a probabilistic classifier based on conditional independence directly applies to our fish classification task. By assuming that features (in this case, color components from the HSV color space) are independent, Naïve Bayes provides a simplified yet powerful method for classification with relatively more minor datasets, which aligns with the constraints of this study.

Multiple machine-learning models have been tested in the fish species identification domain. For instance, SVM and KNN classifiers have been extensively explored for fish classification tasks. A study by [7] applied SVM to identify fish species through acoustic signals, while [11] applied KNN to classify fish based on eye color. These models, while powerful, often require larger datasets and higher computational resources, especially in the case of deep learning approaches like CNNs. The Naïve Bayes algorithm, on the other hand, excels in environments with constrained resources and smaller datasets, providing a balance between simplicity and performance.

While CNNs, as demonstrated by [8], offer superior performance with large datasets, they are computationally expensive. The study by [24] emphasizes the computational advantages of Naïve Bayes, particularly in scenarios where training time and resource constraints are significant, making it an ideal candidate for applications such as identifying freshwater fish species in resource-limited environments.

In addition to other machine learning classification methods, the Naïve Bayes algorithm is commonly used for object image classification due to its simplicity and effectiveness, particularly in scenarios with limited data. This method assigns objects to specific classes or labels based on their attributes [25]. One key advantage of Naïve Bayes is its assumption that all attributes are independent, simplifying the model and reducing computational complexity, especially when working with smaller training datasets [26]. Given these advantages, Naïve Bayes is well-suited for classification tasks, as [27] demonstrated that it works effectively in object image classification.

The Naïve Bayes algorithm has been successfully applied in various classification tasks, such as determining the quality of tuna based on color features. For example, research by [23] achieved an accuracy of 72.727% using Naïve Bayes for this purpose. This method reinforces the method's utility in achieving reliable results, even with limited data. Given its advantages, Naïve Bayes is an ideal candidate for creating a lightweight fish classification model. Therefore, this study aims to develop a model using Naïve Bayes and HSV (Hue, Saturation, Value) color feature extraction to classify freshwater fish species based on their images. By leveraging basic color statistics, this work proposes a practical and scalable solution for automated fish identification, particularly in the fisheries sector. This research aims to provide insights into the development of a classification-based system in the fisheries sector, particularly for identifying freshwater fish species.

The primary contributions of this research are as follows:

- It introduces a simplified fish classification approach using the Naïve Bayes method, which is well-suited for resource-limited environments.
- It demonstrates the feasibility of using basic HSV color features for distinguishing visually similar freshwater fish species, offering a novel approach to fish identification.
- It presents a practical tool that could support fisheries management and monitoring of aquatic biodiversity, particularly in Indonesia.

3 Methodology

This study proposes a lightweight classification framework for identifying freshwater fish species based on image color features. Unlike many previous studies that rely on large and often publicly unavailable datasets, this work utilizes a curated image dataset consisting of 132 images across six commonly consumed freshwater fish species in Indonesia: tilapia, barb, catfish, Java barb, gourami, and pomfret. These species were selected due to their visual similarity and economic significance in Indonesian aquaculture and fish markets, where accurate

classification remains challenging for practitioners and sellers.

The novelty of the dataset lies in its composition: it includes images collected not only from online sources and printed materials but also from original field data photographs taken under varied and uncontrolled conditions. This combination makes the dataset more reflective of real-world environments than many benchmark datasets used in prior studies, causing the first study to compile and utilize a field-augmented dataset of Indonesian freshwater fish species for machine learning-based image classification.

The methodological approach in this study emphasizes simplicity, efficiency, and applicability in resource-limited settings. The methodological approach adopted in this study prioritizes simplicity, computational efficiency, and practical applicability in resource-constrained environments. The research workflow comprises several key stages: data collection, data splitting, preprocessing and feature extraction. The next step is the classification process. First, the Naïve Bayes model is implemented using the training data. Once the model is trained, it is evaluated using the test data, and accuracy is assessed using a confusion matrix, which provides detailed insight into classification accuracy. The complete methodology is illustrated in Figure 1.

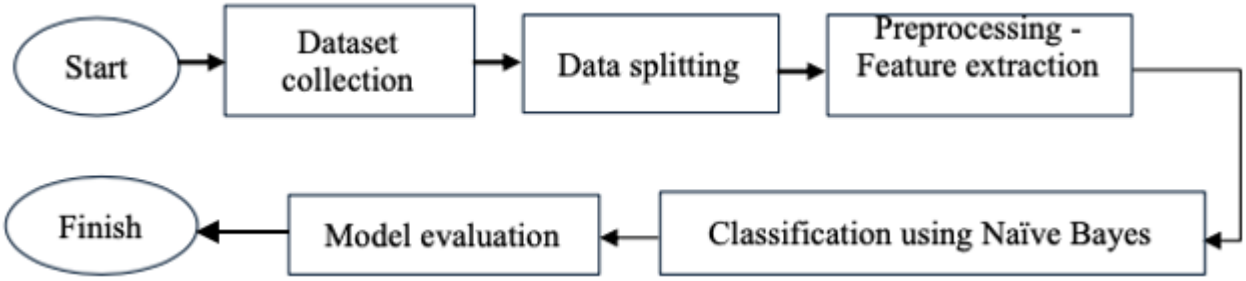


Figure 1: Research flow diagram.

3.1 Dataset collection

The images for this study were collected from three primary sources: publicly available fish images found in books and websites [28], along with original photographs captured using a digital camera. The dataset comprises 132 images representing six freshwater fish species—gourami, catfish, tilapia, barb, Java barb, and pomfret—with 22 images per species. Several examples of images are shown in Figure 2. These images vary in quality, resolution, and lighting to better reflect real-world conditions. The relatively limited dataset size stems from access constraints and practical limitations such as time and available resources. However, it is sufficient for evaluating machine learning, specifically the Naïve Bayes classifier, for fish species identification.

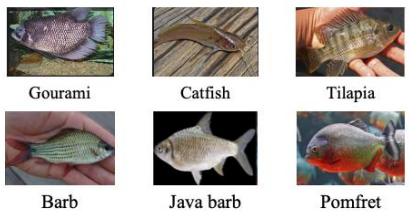


Figure 2: Image examples of freshwater fish species were used as a dataset.

Each image was assigned to one of six classes corresponding to the species, with an effort to keep the class sizes balanced. No explicit rebalancing techniques were used, as the small scale of the dataset and the simplicity of the Naïve Bayes classifier allowed for some natural variation in class representation. While larger, more diverse datasets would enable the use of more complex algorithms and likely yield better results, the current dataset provides a practical foundation for demonstrating the feasibility of this approach. Future research may seek to expand the dataset to improve model generalization and performance.

3.2 Data splitting

The dataset was separated into training and testing sets using an 80:20 ratio, resulting in 108 images for training and 24 for testing, with both sets reflecting all species equally. The training set was used to teach the model to identify freshwater fish from images, while the testing set evaluated how well the model recognized new examples. An 80:20 split was chosen over k-fold cross-validation because the dataset was small; k-fold would have produced folds with too few images, making the results unreliable and potentially unrepresentative of the species distribution. This simpler split offered a direct, computationally efficient way to check model accuracy on

unseen data and reduce the chance of overfitting. For future research with more data, k-fold cross-validation could be applied to get even more reliable performance estimates.

3.3 Preprocessing and feature extraction

For image preprocessing, we standardized the dataset by resizing all images to a consistent size, removing background noise, and converting each image from the RGB color space to HSV (Hue, Saturation, Value). This RGB-to-HSV transformation was carried out using the standard mathematical conversion commonly found in libraries like OpenCV and MATLAB. Each pixel's RGB value was directly mapped to its corresponding HSV value, which separates color information (Hue), color intensity (Saturation), and brightness (Value). This consistent conversion process helped ensure uniformity across the dataset and set the stage for reliable feature extraction.

Regarding feature extraction, we focused exclusively on the HSV color information. The decision to use HSV was based on its effectiveness in handling variations in lighting and shadows—factors that often complicate fish image analysis. HSV's separation of hue, saturation, and value made it possible to reliably capture the color patterns that distinguish different fish species, which was particularly important given the dataset's small size and the classification model's simplicity. While shape and texture features can also aid in classification, we opted not to include them to avoid overfitting. Future research could integrate features like texture (via GLCM or LBP) or shape (using contours or HOG), especially with larger datasets, to improve accuracy among visually similar species except for subtle shape or texture differences.

3.4 Classification using Naïve Bayes

We selected the Naïve Bayes classifier for a few practical reasons that fit the needs and limitations of this project. For one, it does not need much training data to get up and running, which is ideal when working with smaller datasets. It is also quick to train and straightforward to code, making it a strong choice when the hardware is limited—think rural fish farms or mobile devices. Because Naïve Bayes treats features as independent and handles both categorical and continuous variables, it meshes well with the data we use here, like HSV color values and object area. Specifically, we used the Gaussian version, which assumes that features are normally distributed [29] as in Equation (1).

$$\hat{y} = \arg\max_{y \in Y} P(y) \prod_{i=1}^{n} P(x_i \mid y) \quad (1)$$

$\hat{y}$: The predicted class label

Y: The set of all possible classes

$P(y)$: The prior probability of class y

$x_1, x_2, \ldots, x_n$: The features or attributes of the input data X

$P(x_i|y)$: The likelihood of feature x_i given class y

More advanced models, like CNNs, often do a great job with image classification, but they generally need a lot of labeled data and heavy hardware. These heavyweight models are usually more than necessary for simple classification tasks—especially when features like color and size do most of the work. That is why, in this case, Naïve Bayes was the right starting point: efficient, effective, and a solid baseline.

3.5 Model evaluation

Model performance was measured using a confusion matrix, with accuracy as the primary metric (see Equation (2)). The terms TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative) represent the standard outcomes for classification tasks. Accuracy was chosen because it captures the share of fish images the model correctly classified out of all test cases.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

For this study, the Naïve Bayes classifier's effectiveness was analyzed through the confusion matrix, which details how the model's predictions break down. Specifically, the matrix allows us to track:

- TP: When the model correctly identifies fish species.
- TN: When the model correctly recognizes non-target species or rejects a misclassification.
- FP: When the model mistakenly labels a non-target species as the target species.
- FN: When the model fails to identify the target species despite its presence.

These four outcomes are essential for calculating accuracy, precision, recall, and the F1-score. Equation (2) presents how these metrics are mathematically defined, drawing from TP, TN, FP, and FN values to assess classification results.

A significant advantage of this approach lies in its simplicity, which enables robust classification performance without the reliance on specialized equipment or controlled laboratory environments.

Consequently, the method is highly applicable in broader contexts, including local fisheries and educational settings, particularly where access to advanced computing infrastructure is limited.

The following details regarding the input and testing data are more explicitly defined and aligned with the model evaluation process, as presented in Table 2.

Table 2: Feature types and input data for classification.

Feature Type	Feature Description	Input Data
HSV Features	The Hue, Saturation, and Value components are extracted from image pixels.	Testing data: Test set of images used to evaluate the model's classification performance.
Area Features	Area-related parameters such as fish size or object area in the image.	Testing data: Area features extracted from test set images to be compared against model predictions.

The "Input" column corresponds to the testing data used to evaluate the model. Specifically, the testing data consists of images not included in the training set and are employed to assess the model's generalization ability. The model's performance is evaluated by comparing its predictions to the test set's actual labels or ground truth. For the HSV features, the input data comprises test set images from which the Hue, Saturation, and Value components are extracted. Regarding the area features, the input data refers to area-based parameters, such as fish size or object area, which are also derived from the test set images and subsequently compared to the predicted values.

4 Result and discussion

The Matlab programming language was employed to develop the model in this study. The procedure began with preprocessing and feature extraction, then implemented the Naïve Bayes algorithm and the subsequent model evaluation. The Naïve Bayes model uses the training data to implement the Naïve Bayes algorithm for the classification process. The model's accuracy is evaluated using the test data after training, and a confusion matrix is employed.

4.1 Preprocessing – feature extraction

The dataset employed in this study consists of 132 images divided into 108 training images and 24 testing images, following an 80:20 split. Color feature extraction is performed before classifying the freshwater fish images using the Naïve Bayes algorithm, as depicted in Figure 3.

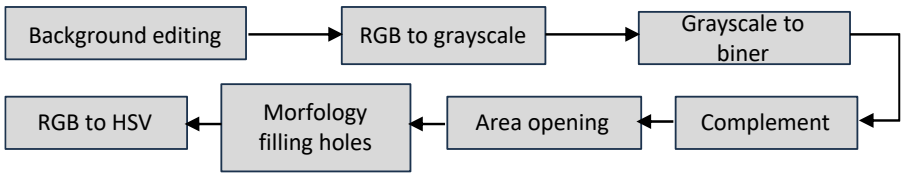


Figure 3: Stages carried out in preprocessing.

The fish images used in this study exhibit considerable size and background composition variation. Manual preprocessing was applied to ensure uniformity and prevent distortion during subsequent processing stages. This process involved two main steps: background removal and image resizing. Each image was cropped to a standardized dimension of 788 × 788 pixels, and the background was removed to isolate the fish object.

The primary purpose of this preprocessing step was to enhance image quality and ensure consistency across the dataset, facilitating more accurate and efficient feature extraction. Standardizing these visual characteristics contributes to the reliability and performance of the classification process.

As illustrated in Figure 4, the initial classification performance was evaluated. An example of a raw fish image is presented in Figure 4a, while its corresponding preprocessed version is shown in Figure 4b. Subsequent refinements, as depicted in Figure 5 and Figure 6, led to notable improvements in performance.

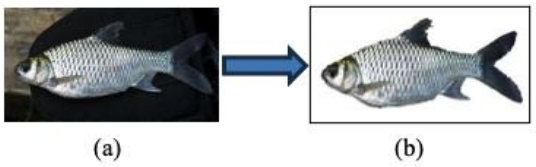


Figure 4: Image of Java barb fish: (a) before editing; (b) after editing.

Following the preprocessing procedures described in Figure 3, the next step involves converting RGB (Red, Green, Blue) images into grayscale. This conversion is performed by averaging the RGB components to produce a single intensity value for each pixel, generating a grayscale image. An example of this transformation is presented in Figure 5a.

The grayscale image is then further processed into a binary (black-and-white) image, as shown in Figure 5b. This binarization reduces image complexity by limiting pixel values to two categories, simplifying the pattern recognition process and enhancing the detectability of relevant features.

A complement operation is applied to the binary image to improve image contrast further. This operation inverts the pixel values by subtracting each from 255, effectively switching object and background representation. As a result, the object is denoted by a pixel value of 1, while the background is represented by 0. This transformation benefits subsequent image-processing steps such as morphological operations and feature extraction. The outcome of the complement operation is illustrated in Figure 5c.

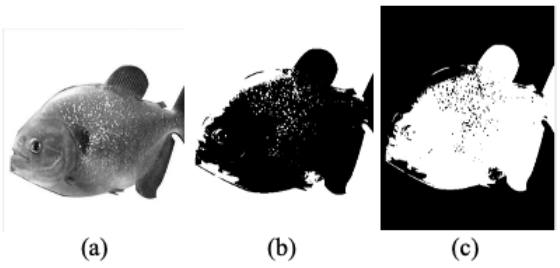


Figure 5: Results of the color conversion process of freshwater fish images: (a) grayscale image; (b) binary image; (c) complement.

The next step involves applying an area-opening operation to the image to remove small, irrelevant objects that may distort the image. This morphological operation helps preserve the integrity of the image. Following this, morphological closing is performed to fill small gaps or holes in the image. Figure 6 presents the image before and after the morphological process. Small objects are removed, and gaps previously with a value of 0 are closed with a value of 1.

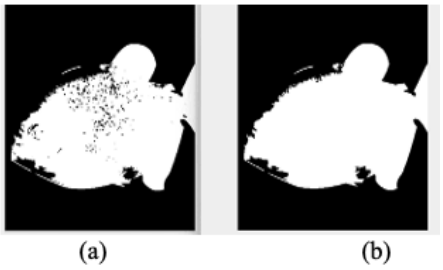


Figure 6: Results of the morphological process: (a) Image after area opening process; (b) image after morphology filling holes process.

The final preprocessing stage involves converting the RGB image to the HSV (Hue, Saturation, Value) color space. This conversion enables the extraction of information related to the image's color, brightness, and purity. The HSV color space is advantageous, as it separates these components, making it easier to detect fish objects even in slightly dark images, as the color purity and light intensity do not significantly affect detection. The conversion from RGB to HSV for the fish image is shown in Figure 7.

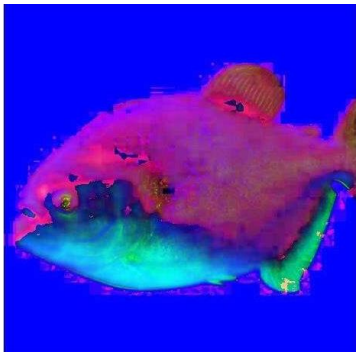


Figure 7: Results of RGB to HSV image conversion.

This study uses the HSV color space for feature extraction to distinguish between the different fish species. The HSV model separates an image's color into three components: Hue, Saturation, and Value. Each component plays an essential role in representing the color characteristics of the fish.

- Hue (H): This represents the color type of the image, such as red, green, or blue. The Hue value is typically measured on a scale from 0 to 360 degrees, where each degree corresponds to a different color. For instance,

0°corresponds to red, 120° corresponds to green, and 240° corresponds to blue.

- Saturation (S): This component indicates the intensity or vividness of the color. It ranges from 0 to 1, in which zero means the color is a shade of gray (i.e., no saturation), and one means the color is fully saturated, appearing as its purest form. Saturation helps determine how vibrant or dull a color appears.
- Value (V): This represents the brightness or lightness of the color, ranging from 0 to 1, where 0 corresponds to black (no brightness), and 1 corresponds to the brightest possible color. The Value component determines the color's light or dark in the image.

Feature extraction in this study focuses on extracting the HSV color features and the object area. At this stage, the Hue (H), Saturation (S), and Value (V) components, along with the area of the object in the image, are extracted. The background pixels in the H, S, and V components are set to 0 to ensure that only object pixels are processed in the subsequent steps, specifically in calculating the average values. Additionally, the average area of the object is calculated. Table 3 and Table 4 provide examples of color feature extraction and object area calculation results.

Table 3: Results of color feature extraction and area of freshwater fish images on training data.

No.	Hue	Saturation	Value	Area
1	0.403318323419042	0.260346695106469	0.536819084229093	84286
2	0.335057730985058	0.152018628974081	0.394937026807920	71350
3	0.270479850906780	0.258428038903359	0.331773459548743	74277
4	0.330772366940356	0.435419654272074	0.318038088329406	98443
5	0.589932605894102	0.158395421440156	0.507638669354549	62707
...				
108	0.124554169492385	0.251562245511451	0.403377826856695	141056

Table 4: Results of color feature extraction and area of freshwater fish images on testing data.

No.	Hue	Saturation	Value	Area
1	0.500006801952211	0.280613369049232	0.662378360420152	60908
2	0.372640832644990	0.220835127629331	0.380358140777996	139611
3	0.170120032846037	0.315069446363992	0.434321305868149	192706
4	0.433518886610450	0.262447886902986	0.349181193584367	107974
5	0.117228933141659	0.223715743829939	0.460529587190712	196720
...				
24	0.320989767431954	0.273306776067016	0.497645896540657	179498

4.2 Naïve Bayes algorithm implementation

In this study, we applied the Naïve Bayes algorithm to classify freshwater fish species using visual data, focusing on the HSV color feature extraction method. The stages of implementing the Naïve Bayes algorithm are as follows:

- Read the train data
- Counting the number of classes/labels
- Calculate the probability value of the hypothesis using Equation (1).

Table 5 displays the output of the Naïve Bayes algorithm's prediction results in solutions.

In the HSV conversion process, the function "$value_{ofColor}$" refers to the Value component in the HSV

model, which is derived from the RGB values of each pixel in the image. The term "value_ofColor" is used to indicate the brightness of the pixel's color, which is an important factor in differentiating between various fish species with similar Hue and Saturation characteristics. The value categories referenced in the computations of Table 5 are derived from the data shown in Table 3 and Table 4, specifically:

Table 5: Illustration of computational results using the Naïve Bayes algorithm.

No.	Hue	Saturation	Value	Area	Class
1	2	2	2	1	pomfret
2	2	1	2	2	pomfret
3	1	2	2	2	pomfret
4	2	2	1	2	pomfret
5	1	1	2	2	gourami
6	1	1	2	2	gourami
7	2	2	2	2	gourami
8	2	2	2	2	gourami

$$\frac{Hue}{Saturation} = \begin{cases} 1;\ if\ value_{ofColor} \leq 0.2 \\ 2;\ if\ 0.2 < value_{ofColor} \leq 0.6 \end{cases}$$

$$Value = \begin{cases} 1;\ if\ value_{ofColor} \leq 0.34 \\ 2;\ if\ 0.34 < value_{ofColor} \leq 0.6 \end{cases}$$

As explained previously, the Naïve Bayes (NB) algorithm is implemented using the MATLAB programming language in this case. Figure 8 shows an example of a program code snippet that utilizes the "fitcnb" function to perform classification using the Gaussian Naïve Bayes (GNB) algorithm with parameters in the form of "ciri_latih" and "kelas_latih."

```
% klasifikasi citra menggunakan Naive Bayes
Mdl = fitcnb(ciri_latih,kelas_latih);
```

Figure 8: The “fitcnb” function for training the fish image dataset.

In the program snippet that invokes the "fitncb" function, the GNB Algorithm carries out the data training process. This process generates a prediction model for freshwater fish species based on all the input data, specifically the variables "ciri_latih" and "kelas_latih," using the Bayes probability concept as described in Equation (1). The output of this function is stored in the variable "Mdl."

Once the training process recognizes the pattern, the resulting model undergoes evaluation. This evaluation aims to assess the model's performance, particularly its accuracy in classifying freshwater fish species based on their images. A program excerpt demonstrating the model evaluation process is presented in Figure 9, where the "predict" function is utilized.

```
% membaca kelas keluaran hasil pengujuan
hasil_uji = predict(Mdl,ciri_uji);
```

Figure 9: Program code snippet for evaluating the model.

The "Mdl" variable stores the training process results, while the "ciri_uji" variable contains the testing data along with the extracted features. The "predict" function is employed to make predictions on the testing data using the model that was previously trained. The resulting predictions are stored in the "hasil_uji" variable, with the corresponding data presented in Table 6.

Table 6: Prediction results for freshwater fish species based on 24 testing data samples.

No.	Hue	Saturation	Value	Area	Input	Output
1	0.500006801952211	0.280613369049232	0.662378360420152	60908	pomfret	pomfret
2	0.372640832644990	0.220835127629331	0.380358140777996	139611	pomfret	pomfret
3	0.170120032846037	0.315069446363992	0.434321305868149	192706	pomfret	gourami
4	0.433518886610450	0.262447886902986	0.349181193584367	107974	pomfret	pomfret
5	0.117228933141659	0.223715743829939	0.460529587190712	196720	gourami	gourami
6	0.116742908634686	0.223105665366549	0.460431169214331	196720	gourami	gourami
7	0.320989767431954	0.273306776067016	0.497645896540657	179498	gourami	tilapia
8	0.320782153662419	0.271708458927080	0.497442417724275	179507	gourami	tilapia
9	0.229210511201839	0.0964511843275545	0.349412906160667	23362	catfish	catfish
10	0.106550890144485	0.158231246737144	0.371697006596770	21062	catfish	catfish
...						
24	0.222269932465241	0.120431023603942	0.606964743707740	108652	Java barb	barb

As indicated in Equation (2), the data presented in Table 6 must be evaluated using a measurement model represented by a confusion matrix, as shown in Table 7. This confusion matrix reflects the results obtained from Equation (2) implementation, capturing the relationship between predicted and actual classifications. A detailed matrix analysis reveals the fish species that were most frequently misclassified, thereby providing valuable insights into the model's performance and highlighting its strengths and limitations.

Table 7: Model performance evaluation results using the confusion matrix.

Predicted\Actual	Pomfret	Gourami	Catfish	Barb	Tilapia	Java barb
Pomfret	3	1	0	0	0	0
Gourami	0	2	0	0	2	0
Catfish	0	0	4	0	0	0
Barb	0	0	0	4	0	0
Tilapia	0	0	0	0	4	0
Java barb	0	0	0	2	0	2

Based on Table 7, the values for TP, FP, TN, and FN can be determined, allowing for the calculation of the accuracy, as follows:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{19}{24} = 0.79167 \approx 79.17\%$$

A detailed examination of the classification outcomes indicates that the most frequent misclassifications occurred between the Java barb and barb species. These species exhibit similar color patterns, likely leading to confusion during classification. Notably, the Java barb was misclassified as a barb on two occasions, suggesting that the HSV (Hue, Saturation, Value) color features used in the model could not capture the subtle visual distinctions between these two species. A similar trend

was observed in the misclassification of gourami as tilapia, which also share comparable color tones. These findings emphasize the limitations of relying solely on color-based features, particularly HSV, which are sensitive to lighting variations and often fail to differentiate objects with closely related visual characteristics.

Despite these misclassifications, the Naïve Bayes model successfully distinguished species such as catfish, barb, and tilapia, which possess more distinct and contrasting color features. However, the model's performance declined when classifying species with overlapping attributes, such as body shape and color hue—evident in the confusion between Java barb and tilapia. Such challenges are common in image-based classification tasks, where visual similarity across classes can significantly hinder accuracy. Integrating additional feature descriptors—particularly those based on texture and shape—is recommended to address these limitations. These enhancements would provide more discriminative power and improve the model's robustness in accurately identifying fish species with subtle inter-class differences.

In this study, HSV color features were primarily utilized for classification as a feature engineering aspect. However, it is essential to investigate whether other feature extraction techniques could enhance the model's performance, particularly for species with similar visual characteristics, such as the Java barb and barb.

Several alternative feature extraction methods could complement the HSV color features to improve classification accuracy. These methods include:

- **Gabor Filters**: Gabor filters are effective for texture-based feature extraction and are commonly used in image processing due to their ability to capture local spatial frequency patterns. Gabor filters could be particularly beneficial in distinguishing between species that exhibit subtle textural differences, which may not be fully captured by color features alone [30].
- **Histogram of Oriented Gradients (HOG)**: HOG is a feature descriptor designed to capture object shapes and patterns by analyzing the distribution of gradient orientations. This method is frequently employed in object detection tasks and could improve classification performance for fish species that differ in shape but share similar color patterns [31].
- **Gray-Level Co-occurrence Matrix (GLCM)**: GLCM captures texture information by analyzing the spatial relationships between pixel values. By incorporating GLCM features, the model could become more robust in classifying species with similar appearances but distinct textures [32].

A comparative analysis could be conducted to evaluate the effectiveness of these methods by extracting features using these techniques and comparing the classification performance with the HSV-based approach. Including these additional features would provide complementary information, especially for fish species that share similar color patterns but differ in texture or shape. Implementing multiple feature extraction techniques would be a natural progression of this study and could yield improved results, particularly for misclassified species like the Java barb and barb.

With an accuracy of 79.17%, the model demonstrates promising potential for practical applications in fish species identification. However, to emphasize the novelty of our approach and its improvements over existing techniques, we now compare our method with several other machine learning approaches commonly employed in fish classification tasks.

To validate the performance of the Naïve Bayes model, statistical significance tests were conducted to determine whether the accuracy of 79.17% significantly differed from that of other classifiers. A t-test was performed to compare the accuracy of the Naïve Bayes model with that of two alternative classifiers: Support Vector Machine (SVM) and K-Nearest Neighbors (KNN). These classifiers were chosen due to their frequent use in classification tasks and distinct model construction approaches.

The model was executed multiple times (five repetitions) to account for any potential variation in performance, with the accuracy averaged across these runs. The t-test results indicated that, while the Naïve Bayes model performed adequately, the difference in accuracy between it and the SVM or KNN classifiers was not statistically significant ($p > 0.05$). This finding suggests that, although more complex models like SVM and KNN may show slightly superior performance, the Naïve Bayes algorithm offers competitive accuracy with a significantly lower computational cost and more straightforward implementation.

4.3 Comparison with previous studies

The Naïve Bayes model in this study achieved an accuracy of 79.17% in classifying six freshwater fish species, surpassing the 72.73% accuracy reported by [23], who used Naïve Bayes and HSV features to classify frozen tuna quality. Although the datasets and tasks differed, this comparison highlights the robustness of the Naïve Bayes algorithm across various image-based classification contexts. Given the diverse nature of the dataset—comprising images from the internet, books, and camera captures—there is potential for further optimization in feature extraction, as demonstrated by [33].

To provide context, we compared the Naïve Bayes model with other classifiers, such as Support Vector Machine (SVM) and K-Nearest Neighbors (KNN), using the same dataset and preprocessing steps. As shown in Table 8, Naïve Bayes achieved 79.17% accuracy, while SVM and KNN achieved 82.00% and 80.48%, respectively. Although SVM slightly outperformed the others, it has a significantly higher computational cost, especially regarding training time and memory usage. KNN also performed well but struggles with high-dimensional data, making it less efficient for more complex tasks. Naïve Bayes, with its competitive performance and more straightforward implementation, proves particularly suitable for real-time classification tasks, where speed and resource limitations are crucial.

Table 8: Performance comparison for Naïve Bayes, SVM, and KNN classifiers.

Classifier	Accuracy	Feature Extraction Methods
Naïve Bayes	79.17%	HSV color features
SVM	82.00%	HOG + HSV
KNN	80.48%	GLCM + HSV

Several studies have applied machine learning algorithms for fish classification. For instance, [10] used SVM to classify indigenous fish species in Bangladesh, achieving an accuracy of 80%, which is similar to our results. However, SVM is computationally intensive and may not perform efficiently with smaller datasets. In contrast, Naïve Bayes offers comparable accuracy with a more straightforward and less computationally demanding approach, making it more accessible for practical applications involving limited data.

The K-Nearest Neighbors (KNN) algorithm has also been used in fish freshness classification based on eye color, as shown by [34]. While KNN achieved good accuracy, it is sensitive to the curse of dimensionality, especially when multiple features are involved. Naïve Bayes, assuming feature independence, is more robust in cases where features are not strongly correlated, such as in color-based classification tasks.

Convolutional Neural Networks (CNNs) have shown excellent performance in image classification, as demonstrated by [35], who applied CNNs for fish classification, achieving an accuracy of 93%. While CNNs perform well with large datasets and complex feature extraction, they require substantial labeled data. They are computationally expensive, making them less suitable for smaller datasets or real-time applications. In comparison, Naïve Bayes, although less complex, remains a competitive solution for smaller datasets with efficient implementation.

To summarize, while more complex models such as CNNs outperform Naïve Bayes' accuracy, they come with significantly higher computational costs. Our Naïve Bayes model, which uses HSV color features, provides a more straightforward, more efficient solution with competitive results, especially for resource-constrained applications. Additionally, incorporating more advanced feature extraction techniques, such as Histogram of Oriented Gradients (HOG) or Gray-Level Co-occurrence Matrix (GLCM)—used in SVM and KNN classifiers—could further enhance Naïve Bayes performance by capturing shape and texture information that HSV alone may not fully address.

4.4 Limitations of this study

4.4.1 Model implementation: Hyperparameter selection and tuning

While the Naïve Bayes classifier was implemented using the fitcnb function in MATLAB, the manuscript currently lacks detailed information regarding the selection of hyperparameters for the model. Hyperparameter tuning is crucial in optimizing machine learning models for optimal performance. In this study, we did not explicitly perform hyperparameter tuning, as Naïve Bayes is known to perform effectively with its default settings. However, hyperparameter tuning is typically critical to achieving high performance for more complex models such as Support Vector Machine (SVM) or K-Nearest Neighbors (KNN) [36].

For Naïve Bayes, the primary hyperparameter is the distribution type used for the features. The “fitcnb” function in MATLAB offers several options for selecting the distribution type, including normal distribution (default), kernel distribution, or multinomial distribution [37].

Cross-validation could be employed to evaluate the model's accuracy under different settings and assess whether these distribution types could improve classification performance. This method would allow for identifying the distribution type that best suits the feature set, potentially enhancing classification results [38].

Furthermore, future studies could apply advanced techniques such as Bayesian optimization or grid search for automated hyperparameter tuning. These methods have effectively optimized hyperparameters, improving model performance. Implementing these tuning methods would be particularly beneficial if additional feature

extraction techniques are incorporated. Different features may require distinct distribution models to capture their underlying characteristics effectively, thus further improving the model's performance [36].

4.4.2 Data augmentation

Given the relatively small dataset size of 132 images, data augmentation is essential to improve the model's generalization and robustness. In the current study, no data augmentation was applied to the dataset, which may have limited the model's ability to generalize to unseen data. Data augmentation can significantly expand the diversity of the dataset, particularly in situations where collecting more labeled data is challenging [39].

To address this limitation, future work should experiment with standard augmentation techniques such as:

- **Rotation**: Rotating images by small degrees can simulate orientation changes, ensuring the model can recognize fish species from various angles [40].
- **Scaling**: Scaling images allows the model to learn scale-invariant features, which could help recognize varying-sized fish species [40].
- **Color Jittering**: Modifying the color saturation, contrast, and brightness can help the model become more robust to changes in lighting conditions, which is especially useful when images are taken in different environments [41], [42].

By incorporating these techniques, the model's performance could be improved, particularly in real-world applications where images vary in orientation, size, or lighting. Data augmentation enhances the model's robustness and reduces the risk of overfitting, thereby improving its generalization capabilities [39].

4.4.3 Robustness evaluation

An essential aspect of the study is how well the model performs under real-world conditions. In this experiment, the dataset used for training and testing was limited to images with controlled lighting and backgrounds. However, real-world images of fish may exhibit significant lighting, background, and noise variations, affecting the model's ability to make accurate predictions [43].

To assess the robustness of the Naïve Bayes model, future studies should evaluate its performance on external datasets collected under diverse conditions, including:

- **Varying Lighting**: Fish images taken in different lighting conditions can cause color variations, which could challenge the model. The inclusion of color jittering during training, as mentioned in the previous section, could mitigate this issue [40].
- **Different Backgrounds**: Images captured against cluttered or inconsistent backgrounds can introduce noise that affects the model's performance. Preprocessing techniques like background removal or segmentation could help these cases [44].

Additionally, the model should be tested on real-world images collected from different sources to validate its ability to generalize across diverse environments. This external validation is crucial for ensuring the model performs well in practical applications, such as automated fish identification in aquaculture systems [43].

4.4.4 Limitations of the Naïve Bayes approach

While the Naïve Bayes algorithm has demonstrated reasonable performance in this study, it has several limitations compared to more complex models such as Convolutional Neural Networks (CNNs) and Support Vector Machines (SVMs). One of the primary limitations of Naïve Bayes is its assumption of feature independence, which, although beneficial for simplifying the model, is often unrealistic in real-world data where features may be correlated [45]. For instance, in fish images, attributes such as color, shape, and texture are typically interdependent, yet Naïve Bayes treats them as independent, which can lead to suboptimal performance [46].

Furthermore, Naïve Bayes does not automatically learn the optimal features from raw image data. In contrast, CNNs excel in automatically extracting hierarchical features from images, which often results in higher classification performance, especially when the dataset is large and complex [47]. While Naïve Bayes can handle smaller datasets and is computationally efficient, its performance may be limited in more challenging classification tasks, particularly those involving high-dimensional image data with intricate details.

Another limitation is relying on manually selected features, such as HSV values, rather than learning features from the data. This learning is a significant advantage of deep learning models like CNNs, which can automatically learn features directly from raw data [47]. This reliance on predefined features restricts Naïve Bayes' ability to generalize to unseen data or more complex image conditions, a critical aspect of modern image classification tasks.

4.4.5 Computational efficiency

This study chose the Naïve Bayes classifier primarily for its computational efficiency, making it suitable for real-time applications and environments with limited

resources. Compared to more complex classifiers like Support Vector Machines (SVM) and Convolutional Neural Networks (CNNs), Naïve Bayes requires significantly less computational power and training time. For instance, a study by [48] reported that Naïve Bayes had a training time of 0.35 seconds and an inference time of 0.01 seconds. In contrast, SVM took 15.62 seconds for training and 0.02 seconds for inference. These results underscore the trade-off between model complexity and computational efficiency, highlighting Naïve Bayes as an attractive option for applications where speed and simplicity are prioritized.

Further supporting this, research by [49] compared Naïve Bayes, SVM, and K-Nearest Neighbors (KNN) in sentiment analysis of public opinion regarding COVID-19 vaccination on Twitter. They found that Naïve Bayes achieved an accuracy of 94%, with faster training and inference times than SVM and KNN.SVM achieved the highest accuracy at 96.3% but with longer training and inference times. KNN achieved an accuracy of 91%, with inference times faster than SVM but slower than Naïve Bayes. This study utilized a Twitter dataset of 35,644 tweets and applied TF-IDF feature extraction and TextBlob for labeling.

4.5 Recommendations and future works

To enhance the performance and applicability of the Naïve Bayes model in fish species classification, several avenues for future research are proposed:

- **Integration of advanced feature extraction techniques**: Combining feature extraction methods, such as Gabor filters, Histogram of Oriented Gradients (HOG), and Gray Level Co-occurrence Matrix (GLCM), can improve the model's ability to distinguish visually similar species. For instance, Gabor filters have effectively identified goldfish species [50] while HOG features have been applied in fish freshness detection [51].
- **Hyperparameter tuning**: Utilizing techniques such as cross-validation or Bayesian optimization can help select the optimal distribution type for the features, thereby improving the performance of the Naïve Bayes model [52].
- **Data augmentation**: Implementing data augmentation techniques, including rotation, scaling, and color jittering, can artificially expand the dataset, improving model generalization and reducing overfitting, especially when working with relatively small datasets [53].
- **Utilization of K-Fold Cross-Validation**: Applying k-fold cross-validation instead of a simple train-test split can provide more robust performance estimates and ensure the model does not overfit the training data [54].
- **Development of hybrid models**: Combining Naïve Bayes with other machine learning algorithms, such as K-Nearest Neighbors (KNN) or Support Vector Machines (SVM), can leverage the strengths of different models, potentially improving classification accuracy [55].
- **Robustness testing**: Testing the model on external datasets under diverse real-world conditions, such as varying lighting and backgrounds, can assess its robustness and ensure reliable performance in real-world applications [56].
- **Enhancement of the dataset**: Collecting a larger and more diverse dataset, particularly under varying lighting conditions, can help the model perform better across different environmental settings and reduce misclassifications caused by subtle image variations [39].

4.6 Novelty and improvements

Although this study has limitations, one key improvement in our approach is the careful preprocessing and feature extraction strategy. By converting RGB images to HSV color space, we separate the image's color, saturation, and brightness components, which enhances the classification model's ability to identify fish species under varying lighting conditions. This step is crucial in the context of fish species that share similar morphological features, which is a challenge addressed by our method.

Moreover, our model leverages a relatively small dataset of 132 images, smaller than datasets typically used in deep learning approaches like CNNs. Despite this, our Naïve Bayes classifier achieved a competitive accuracy, showcasing its potential for efficient classification in scenarios where large datasets are unavailable. This aspect of our approach is particularly beneficial for resource-constrained environments where data collection may be limited.

Finally, our study contributes to the field by demonstrating that the Naïve Bayes classifier can be effectively applied in fish identification, even with the relatively simple HSV feature extraction method. This finding contrasts with other more complex algorithms, making Naïve Bayes an attractive choice for practitioners seeking a balance between simplicity, speed, and accuracy.

5 Conclusion

In conclusion, the Naïve Bayes classifier demonstrated competitive performance for fish species classification, achieving an accuracy of 79.17%. While slightly outperformed by Support Vector Machine (SVM), the model's simplicity, computational efficiency, and suitability for smaller datasets make it a practical choice for real-world applications in the field. This study underscores the applicability of the Naïve Bayes algorithm for automated fish species identification using image data. Despite its satisfactory performance, future improvements can be made by incorporating additional feature extraction techniques, optimizing hyperparameters, and exploring more complex models, such as hybrid approaches and Convolutional Neural Networks (CNNs). Expanding the dataset size and employing data augmentation techniques would also enhance classification accuracy and robustness, allowing the model to differentiate visually similar species better. Although more advanced models may demand higher computational resources, these advancements could significantly increase the model's accuracy, making it a more effective and reliable solution for fish classification in practical applications.

Acknowledgments

The author would like to thank the Universitas Muhammadiyah Purwokerto for supporting and providing funds to implement this research through the Research and Community Service Board with contract number A.11-III/6939-S.Pj./LPPM/II/2024.

References

[1] E. Trewavas, *Tilapiine fishes of the genera Sarotherodon, Oreochromis, and Danakilia*. 1983.

[2] S. Juliani, "Karakteristik Morfometrik dan Meristik Ikan Nila, Oreochromis niloticus (Linnaeus 1758) di Danau Tempe, Kabupaten Wajo dan Danau Sidenreng, Kabupaten Sidenreng Rappang Provinsi Sulawesi Selatan," Universitas Hasanuddin, 2023.

[3] G. Kwikiriza, M. J. Yegon, N. Byamugisha, A. Beingana, F. Atukwatse, A. Barekye, J. K. Nattabi, and H. Meimberg, "Morphometric variations of Nile tilapia (Oreochromis niloticus) (Linnaeus, 1758) local strains collected from different fish farms in South Western Highland Agro-Ecological Zone (SWHAEZ), Uganda: screening strains for aquaculture," *Fishes*, vol. 8, no. 4, p. 217, 2023, https://doi.org/10.3390/fishes8040217.

[4] Y. Hadjidji, "HUBUNGAN KEKERABATAN BEBERAPA JENIS IKAN (Pisces) AIR TAWAR BERDASARKAN KARAKTER MORFOLOGI DAN PEMANFAATANNYA SEBAGAI MEDIA PEMBELAJARAN," 2019, *Universitas Tadulako, Palu, Sulawesi Tengah*.

[5] H. Mustafidah, B. R. Alfiansyah, and N. Hidayat, "Expert System Using Forward Chaining to Determine Freshwater Fish Types Based on Water Quality and Area Conditions," in *2023 Eighth International Conference on Informatics and Computing (ICIC)*, IEEE, 2023, pp. 1–5.

[6] S. Zhao, S. Zhang, J. Liu, H. Wang, J. Zhu, D. Li, and R. Zhao, "Application of machine learning in intelligent fish aquaculture: A review," *Aquaculture*, vol. 540, p. 736724, 2021, https://doi.org/10.1016/j.aquaculture.2021.736724.

[7] A. Yassir, S. J. Andaloussi, O. Ouchetto, K. Mamza, and M. Serghini, "Acoustic fish species identification using deep learning and machine learning algorithms: A systematic review," *Fish Res*, vol. 266, p. 106790, 2023, https://doi.org/10.1016/j.fishres.2023.106790.

[8] W. Rahman *et al.*, "A Deep CNN-Based Salinity and Freshwater Fish Identification and Classification Using Deep Learning and Machine Learning," *Sustainability*, vol. 16, no. 18, p. 7933, 2024, https://doi.org/10.3390/su16187933.

[9] T. P. Trappenberg, *Fundamentals of Machine Learning*. Oxford University Press, 2020.

[10] M. A. Islam, M. R. Howlader, U. Habiba, R. H. Faisal, and M. M. Rahman, "Indigenous fish classification of Bangladesh using hybrid features with SVM classifier," in *2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2)*, IEEE, 2019, pp. 1–4. 10.1109/IC4ME247184.2019.9036679.

[11] Y. Krisdiani, "Klasifikasi kesegaran ikan berdasarkan warna mata menggunakan metode K-Nearest Neighbor," 2019, *Universitas Negeri Malang*.

[12] A. Pariyandani, D. A. Larasati, E. P. Wanti, and M. Muhathir, "Klasifikasi Citra Ikan Berformalin Menggunakan Metode k-NN dan GLCM," in *Semantika (Seminar Nasional Teknik Informatika)*, 2019, pp. 42–47. [Online]. Available: https://semantika.polgan.ac.id/index.php/Semantika/article/view/49

[13] E. P. Wanti and M. Muhathir, "Pengidentifikasian Citra Ikan Berformalin Dengan Menggunakan Metode Multilayer Perceptron," *J-SAKTI (Jurnal Sains Komputer dan Informatika)*, vol. 5, no. 1, pp. 491–502, 2021, http://dx.doi.org/10.30645/j-sakti.v5i1.342.

[14] A. Azis, "IDENTIFIKASI JENIS IKAN MENGGUNAKAN MODEL HYBRID DEEP LEARNING DAN ALGORITMA

KLASIFIKASI," *Sebatik*, vol. 24, no. 2, pp. 201–206, 2020, https://doi.org/10.46984/sebatik.v24i2.1057.

[15] I. R. Diesta and W. F. Al Maki, "Klasifikasi Ikan Cupang Menggunakan Support Vector Machine," in *eProceedings of Engineering*, 2021, pp. 10556–10565.

[16] M. A. D. Akbar, A. B. Setiawan, and R. K. Niswatin, "Klasifikasi Jenis Ikan Cupang Menggunakan Metode GLCM Dan KNN," in *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)*, 2021, pp. 152–158.

[17] I. E. Hasym and I. Susilawati, "Klasifikasi Jenis Ikan Cupang Menggunakan Algoritma Principal Component Analysis (PCA) Dan K-Nearest Neighbors (KNN)," *KONSTELASI: Konvergensi Teknologi dan Sistem Informasi*, vol. 1, no. 1, pp. 168–179, 2021, https://doi.org/10.24002/konstelasi.v1i1.4242.

[18] S. Suwarsito, H. Mustafidah, T. Pinandita, and P. Purnomo, "Freshwater Fish Classification Based on Image Representation Using K-Nearest Neighbor Method," *JUITA: Jurnal Informatika*, vol. 10, no. 2, pp. 183–189, 2022, http://dx.doi.org/10.30595/juita.v10i2.15471.

[19] R. Nuraini, A. Wibowo, B. Warsito, W. A. Syafei, and I. Jaya, "Combination of K-NN and PCA Algorithms on Image Classification of Fish Species," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 5, pp. 1026–1032, 2023, https://doi.org/10.29207/resti.v7i5.5178.

[20] N. Rachmat, Y. Yohannes, and A. Mahendra, "Klasifikasi Jenis Ikan Laut Menggunakan Metode SVM dengan Fitur HOG dan HSV," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 8, no. 4, pp. 2235–2247, 2021, https://doi.org/10.35957/jatisi.v8i4.1686.

[21] D. Yusup, S. Faisal, and A. Pratama, "Klasifikasi Jenis Ikan Hias African Cichlid Menggunakan Algoritma Support Vector Machines," *Scientific Student Journal for Information, Technology and Science*, vol. 4, no. 1, pp. 31–38, 2023, http://repository.ubpkarawang.ac.id/id/eprint/2909/

[22] L. Ren *et al.*, "Rapid identification of fish species by laser-induced breakdown spectroscopy and Raman spectroscopy coupled with machine learning methods," *Food Chem*, vol. 400, p. 134043, 2023, https://doi.org/10.1016/j.foodchem.2022.134043.

[23] F. A. Susilo, H. Fitriyah, and G. E. Setyawan, "Sistem Klasifikasi Kualitas Ikan Tongkol Beku Berdasarkan Fitur Nilai Warna HSV Menggunakan Metode Naïve Bayes," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, no. 1, pp. 753–760, 2019, Accessed: Jul. 02, 2025. [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/4218

[24] R. Wang, "Automatic classification of document resources based on Naive Bayesian classification algorithm," *Informatica*, vol. 46, no. 3, 2022, https://doi.org/10.31449/inf.v46i3.3970.

[25] D. A. Nasution, H. H. Khotimah, and N. Chamidah, "Perbandingan normalisasi data untuk klasifikasi wine menggunakan algoritma K-NN," *CESS (Journal of Computer Engineering, System and Science)*, vol. 4, no. 1, pp. 78–82, 2019, https://doi.org/10.24114/cess.v4i1.11458.

[26] N. M. S. Iswari, W. Wella, and R. Ranny, "Perbandingan Algoritma kNN, C4. 5, dan Naive Bayes dalam Pengklasifikasian Kesegaran Ikan Menggunakan Media Foto," *Ultimatics: Jurnal Teknik Informatika*, vol. 9, no. 2, pp. 114–117, 2017, https://doi.org/10.31937/ti.v9i2.659.

[27] E. Ellif, S. H. Sitorus, and R. Hidayati, "Klasifikasi Kematangan Pepaya Menggunakan Ruang Warna HSV dan Metode Naive Bayes Classifier," *Coding Jurnal Komputer dan Aplikasi*, vol. 9, no. 01, pp. 66–75, 2021, http://dx.doi.org/10.26418/coding.v9i01.45906.

[28] ResepKoki, "10 Jenis Ikan Air Tawar di Indonesia Yang Sering Dikonsumsi," https://resepkoki.id/10-jenis-ikan-air-tawar-di-indonesia-yang-sering-dikonsumsi/. Accessed: Nov. 07, 2021. [Online]. Available: https://resepkoki.id/10-jenis-ikan-air-tawar-di-indonesia-yang-sering-dikonsumsi/

[29] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[30] L. Zhu, T. Chen, J. Yin, S. See, and J. Liu, "Learning gabor texture features for fine-grained recognition," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 1621–1631. Accessed: Jul. 15, 2025. [Online]. Available: https://openaccess.thecvf.com/content/ICCV2023/html/Zhu_Learning_Gabor_Texture_Features_for_Fine-Grained_Recognition_ICCV_2023_paper.html

[31] G. Ajmera, "HOG (Histogram of Oriented Gradients): An Amazing Feature Extraction Engine for Medical Images," https://medium.com/@girishajmera/hog-histogram-of-oriented-gradients-an-amazing-feature-extraction-engine-for-medical-images-5a2203b47ccd.

[32] F. Xiao, L. Kaiyuan, W. Qi, Z. Yao, and Z. Xi, "Texture analysis based on gray level co-occurrence matrix and its application in fault detection," in *International Geophysical Conference, Beijing, China, 24-27 April 2018*, Society of Exploration

Geophysicists and Chinese Petroleum Society, 2018, pp. 836–839.

[33] A. Priyambodo and P. Prihati, "Klasifikasi Gambar Aksara Jawa Menggunakan Optimalisasi Parameter SVM dengan Kernel Cosine," *MEANS (Media Informasi Analisa dan Sistem)*, pp. 137–142, 2024, Accessed: Jul. 21, 2025. [Online]. Available: https://ejournal.ust.ac.id/index.php/Jurnal_Means/article/view/4357

[34] D. Dahlan, R. Iskandar, N. Ekawati, and C. A. Sugianto, "Mackerel Tuna Freshness Identification Based on Eye Color Using K-Nearest Neighbor Enhanced by Contrast Stretching and Histogram Equalization," *Scientific Journal of Informatics*, vol. 11, no. 4, pp. 1035–1042, 2024, https://doi.org/10.15294/sji.v11i4.14494.

[35] N. Fatima and V. Yadav, "Fish Species Classification Using Convolutional Neural Networks," in *International Conference on IoT, Intelligent Computing and Security: Select Proceedings of IICS 2021*, Springer, 2023, pp. 413–423. https://doi.org/10.1007/978-981-19-8136-4_34.

[36] B. Bischl *et al.*, "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *Wiley Interdiscip Rev Data Min Knowl Discov*, vol. 13, no. 2, p. e1484, 2023, https://doi.org/10.1002/widm.1484.

[37] MathWorks, "Classification Naive Bayes - Naive Bayes classification for multiclass classification," https://www.mathworks.com/help/stats/classificationnaivebayes.html.

[38] MathWorks, "fitcnb - Train Multiclass Naive Bayes model," https://www.mathworks.com/help/stats/fitcnb.html.

[39] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J Big Data*, vol. 6, no. 1, pp. 1–48, 2019, https://doi.org/10.1186/s40537-019-0197-0.

[40] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *arXiv preprint arXiv:1712.04621*, 2017, https://doi.org/10.48550/arXiv.1712.04621.

[41] C. Hu *et al.*, "Impact of light and shadow on robustness of deep neural networks," *arXiv preprint arXiv:2305.14165*, 2023, https://doi.org/10.48550/arXiv.2305.14165.

[42] O. Mazhar and J. Kober, "Random shadows and highlights: A new data augmentation method for extreme lighting conditions," *arXiv preprint arXiv:2101.05361*, 2021, https://doi.org/10.48550/arXiv.2101.05361.

[43] Y. Bengio, I. Goodfellow, and A. Courville, *Deep learning*, vol. 1. MIT press Cambridge, MA, USA, 2017.

[44] C. Zhou *et al.*, "Evaluation of fish feeding intensity in aquaculture using a convolutional neural network and machine vision," *Aquaculture*, vol. 507, pp. 457–465, 2019, https://doi.org/10.1016/j.aquaculture.2019.04.056.

[45] R. Blanquero, E. Carrizosa, P. Ramírez-Cobo, and M. R. Sillero-Denamiel, "Variable selection for Naïve Bayes classification," *Comput Oper Res*, vol. 135, p. 105456, 2021, https://doi.org/10.1016/j.cor.2021.105456.

[46] P. Langley and S. Sage, "Induction of Selective Bayesian Classifiers," in *Uncertainty in Artificial Intelligence Proceedings of the Tenth Conference*, 2013. https://doi.org/10.1016/B978-1-55860-332-5.50055-9.

[47] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, https://doi.org/10.1038/nature14539.

[48] D. R. Amancio *et al.*, "A systematic comparison of supervised classifiers," *PLoS One*, vol. 9, no. 4, p. e94137, 2014, https://doi.org/10.1371/journal.pone.0094137.

[49] K. Munawaroh and A. Alamsyah, "Performance comparison of SVM, naïve bayes, and KNN algorithms for analysis of public opinion sentiment against COVID-19 vaccination on Twitter," *Journal of Advances in Information Systems and Technology*, vol. 4, no. 2, pp. 113–125, 2022, https://doi.org/10.15294/jaist.v4i2.59493.

[50] Y. Heningtyas, A. Syarif, and A. G. Pratidina, "Implementation of Gabor Filter for Carassius Auratus's Identification," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 11, no. 2, pp. 566–571, 2021, https://doi.org/10.18517/ijaseit.11.2.8128.

[51] D. Y. Kim, S. W. Park, and H. S. Shin, "Fish Freshness Indicator for Sensing Fish Quality during Storage," *Foods*, vol. 12, no. 9, May 2023, 10.3390/foods12091801.

[52] A. Sharma, "Naive Bayes with Hyperparameter Tuning," https://www.kaggle.com/code/akshaysharma001/naive-bayes-with-hyperpameter-tuning.

[53] X. Ding, "Texture Feature Extraction Research Based on GLCM-CLBP Algorithm," in *7th International Conference on Education, Management, Information and Mechanical Engineering (EMIM 2017)*, 2017. 10.2991/emim-17.2017.36.

[54] J. Brownlee, "A Gentle Introduction to k-fold Cross-Validation,"

https://www.machinelearningmastery.com/k-fold-cross-validation/, Oct. 04, 2023.

[55] D. C. Asogwa, S. O. Anigbogu, I. E. Onyenwe, and F. A. Sani, "Text classification using hybrid machine learning algorithms on big data," *International Journal of Trend in Research and Development*, vol. 6, no. 5, 2019, https://doi.org/10.48550/arXiv.2103.16624Focustolearnmore.

[56] F. Monteiro *et al.*, "Classification of fish species using multispectral data from a low-cost camera and machine learning," *Remote Sens (Basel)*, vol. 15, no. 16, p. 3952, 2023.

RoBERTa-BiLSTM-GAT Framework for Behavior Extraction and Case Matching from Legal and Multimodal Data

Ying Zhang*, Min Li
Luohe Food Engineering Vocational University; Luohe 462000, China
*Corresponding author: Ying Zhang; email: m13839512135@163.com

Keywords: behavior extraction, case matching, RoBERTa, GAT, BiLSTM

Received: June 9, 2025

As big data and information technology continue to develop, improving the effectiveness of behavior extraction and case matching in intelligent decision-making systems has become an urgent need. To this end, this study proposes a behavior extraction and case matching model combining multi-level feature learning and graph neural networks. Methodologically, the behavior feature extraction module is constructed using a robustly optimized Transformer encoder representation model and a bidirectional long short-term memory network. A graph attention network is introduced to optimize the topological matching mechanism between cases. The model was validated on the CaseLaw and Twitter Sentiment140 datasets. The experimental results showed that the model achieved F1 scores of 90.89% and 93.89% for the behavior extraction and case matching tasks, respectively. The average processing time for matching was as short as 0.63 seconds. Compared with advanced methods such as T5, DGI, and MANN, this model demonstrated significant advantages in terms of accuracy, recall rate, and matching efficiency. Additionally, in testing with text-image multimodal data, the proposed model achieved an average matching adjustment count of approximately 3.5, a matching throughput of up to 210 times per second, and a matching confidence score of up to 0.92. These results fully validate the superiority and practicality of this method in complex behavioral pattern analysis.

Povyetek: Za področje prava in večmodalnih podatkov je razvit model RoBERTa-BiLSTM-GAT, ki združuje večnivojsko semantično učenje z grafno pozornostjo za hkratno ekstrakcijo vedenj in ujemanje primerov. Jedro prispevka je adaptivno tehtanje konteksta ter topoloških razmerij med primeri, razširjeno na besedilno-slikovne vhode z lahkim večmodalnim zlitjem in rezidualno optimizacijo za stabilno, razložljivo ujemanje.

1 Introduction

The use of behavior extraction and case matching technologies in intelligent decision-making systems has grown in popularity in recent years due to the quick growth of big data, artificial intelligence, and natural language processing (NLP) technologies. The technology not only plays a core role in the fields of judicial precedent analysis, financial risk control, medical diagnosis, and intelligent recommendation, but also shows important application value in network security, social media public opinion monitoring, and emergency event warning [1-2]. Although traditional behavior extraction methods rely on rule matching and feature engineering and can achieve some success in structured data environments, they still face serious limitations when dealing with massive, unstructured, and multimodal data. On the one hand, rule matching methods need to manually define a large number of complex rules, which makes it difficult to cope with diverse and dynamically changing behavioral patterns. On the other hand, feature engineering-based methods are highly dependent on data quality and lack deep understanding of contextual information, resulting in limited generalization ability (GA) of the model. Recent years witnessed a notable advancement in machine learning-based behavior extraction techniques due to the quick development of deep learning and NLP. Huang et al. proposed a similar event matching algorithm by joint bidirectional encoder representations from Transformers (BERT) and bidirectional long short-term memory (BiLSTM) behavioral extraction of text data in order to detect early and process emergency events quickly. The outcomes demonstrated that the algorithm had a good performance and high timeliness in case processing of some real emergencies [3]. Kusal et al. proposed a method for extracting emotional information from text that combined graph neural networks (GNNs). This method was designed to provide a deeper understanding of the emotional state expressed in opinions and text conversations. Experimental results showed that this method accurately extracted emotional information from various case files, providing users with clearer emotional feedback [4]. Adel et al. proposed an alternative event detection model for Hunger Games search based on BERT and neumatic heuristic techniques. According to the experimental results, the model demonstrated excellent data behavior extraction when compared to the most advanced event detection model [5]. Ren et al. proposed a threat knowledge extraction algorithm by combining knowledge graph technology and improved GNN. The

experimental results indicated that the algorithm was more accurate and time-sensitive for threat behavior extraction in intelligence [6].

In recent years, methods such as multilevel feature learning and GNN have become important research directions for solving behavior extraction and case matching problems [7]. Li et al. found that the finite length of short electronic record texts led to severe information sparsity. Therefore, this study combined GNN to propose a learning mechanism for power system event detection. The results showed that the mechanism achieved convincing results on general domain event detection datasets [8]. Gao et al. found that the linguistic complexity and ambiguity of textual descriptions in causal event extraction tended to lead to a less accurate extractor. Therefore, the researchers proposed a novel intra-event causality extraction method by combining GNN and causal association graph. The results indicated that the method outperformed the most advanced baseline method on two publicly available datasets [9]. Peng et al. argued that existing streaming social messaging event detection methods usually face ambiguous event features and thus have low accuracy and generalization capabilities. For this reason, the researchers proposed a new reinforcement-weighted multi-relational GNN framework based on GNN. The results showed that the framework demonstrated superior robustness and accuracy in a wide range of cross-lingual social event detection [10]. According to Wan et al., the primary goal of all event extraction in NLP to date has been to extract events from sentences. Because of this, the researchers used GNN to extract event behavior before creating a multi-focus graph-based framework to manage the extraction task. Numerous tests proved the method's efficacy, and the outcomes revealed that it performed better than the most sophisticated baseline techniques. [11]. Gams et al. primarily use twenty-four laws of the information society to explain the relationship between the information society, electronics, and artificial intelligence. These laws constitute a new set that is not currently present in the literature and highlight the core driving mechanisms of the information society and advances in artificial intelligence [12]. A summary comparison of the various literature sources is shown in Table 1.

Table 1: Summary comparison of various methods

Author(s)	Method/Model	Advantages (Specific Metrics)	Limitations/Drawbacks
Huang et al. [3]	BERT+BiLSTM	High timeliness, suitable for emergency cases	Lacks graph-structured optimization
Kusal et al. [4]	GNN for sentiment extraction	Accurately captures emotions across multiple case files	Lacks temporal modeling; weak generalization
Adel et al. [5]	BERT+heuristic search	High extraction accuracy; outperforms baseline methods	No contextual semantic fusion mechanism
Ren et al. [6]	Knowledge graph+improved GNN	High accuracy in threat behavior extraction; fast response	Relies heavily on entity relationship quality
Li et al. [7]	GNN for event detection	High detection rate; effective for sparse short texts	Limited ability in modeling complex relationships
Gao et al. [9]	GNN+causal graph	Significant F1 improvement; handles ambiguous texts well	Efficiency issues on large-scale data
Peng et al. [10]	Multi-relational GNN	Strong robustness; high accuracy in cross-lingual detection	Complex feature fusion; high training cost
Wan et al. [11]	Multi-focal GNN	Superior to baselines in multitask event extraction	Adaptability to long text unclear

Gams et al. [12]	Information society laws	Introduces new theoretical perspectives in AI and information	Lacks empirical validation or dataset application

In conclusion, behavior extraction and case matching have advanced significantly in previous research. However, these methods perform poorly when dealing with multimodal data, which lacks effective hierarchical feature modeling. Furthermore, they cannot make full use of graph structure information to optimize the similarity calculation between cases during case matching. To address the above issues, the study proposes a behavior extraction and case matching algorithm that incorporates multilevel feature learning and GNN, aiming to further improve the effectiveness of case matching and decision making at this stage. Specific objectives include (1) There is a need to develop a multi-level semantic feature extraction module based on RoBERTa and BiLSTM to capture both local and global behavioral information. (2) There is a need to introduce graph attention networks (GATs) to optimize similarity computation by modeling complex inter-case relationships through adaptive attention mechanisms (AMs). (3) The models are systematically evaluated for matching accuracy, processing efficiency and robustness on legal and social multimodal datasets such as CaseLaw and Twitter Sentiment140. First, robustly optimized bidirectional encoder representations from Transformers approach (RoBERTa) with BiLSTM combined with self-attention mechanism (SAM) is used to extract multilevel semantic from input text features. Second, the case knowledge graph is constructed based on the relationship between the cases, and the topological information between the cases is modeled using GAT. The study's innovations include: On the one hand, a multi-level feature fusion technique is suggested, allowing the model to capture both global semantic information and local behavioral patterns. On the other hand, GAT is introduced for case matching, which optimizes the matching weights between cases through an adaptive AM to improve the model's ability to model complex behavioral data. The contribution of the study is to propose a behavior extraction and case matching method that integrates multilevel feature learning and GAT. By optimizing the feature expression and introducing the adaptive AM, the accuracy, GA, and computational efficiency of case matching are improved, and a better matching scheme is provided for the intelligent decision-making system.

2 Methods and materials

2.1 Behavioral extraction algorithm based on multi-level feature learning

In case analysis and intelligent decision-making systems, behavior extraction is the core aspect of identifying and extracting key behavioral patterns. Behavior in a case usually consists of a series of events, operations, or decision-making processes involving a variety of data forms, such as textual records, time series data, and multimodal information [13]. To effectively extract the key behaviors in a case, the core elements of behavioral descriptions need to be parsed, among which semantic roles and predicate identification are important components of behavioral extraction. Their relationship is shown in Figure 1 [14-15].

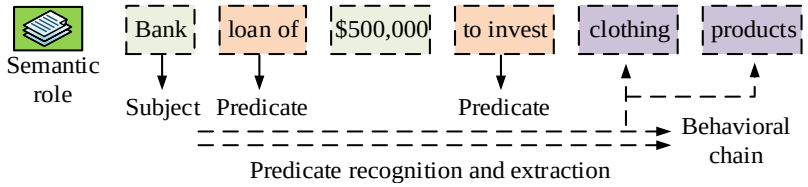


Figure 1: The relationship between semantic role and predicate recognition and action extraction

In Figure 1, the input text undergoes a predicate recognition task that identifies key verb or event words. The semantic role annotation task further associates these predicates with the corresponding roles of giver and receiver to form a structured behavioral representation [16-17]. For example, in the figure, “investment” is recognized as a predicate, while ‘bank’ is the doer role, “500,000” is the amount object, and "clothing products "is the target of the investment. This semantic information together constitutes a complete behavior chain. It can be concluded that traditional behavior extraction methods rely on predefined rules or shallow feature extraction, which makes it difficult to comprehensively portray the contextual relationships and deep semantic features of behaviors. Therefore, the study presents BERT to enhance behavior extraction's precision and resilience. Figure 2 displays the schematic diagram for the BERT structural principle [18].

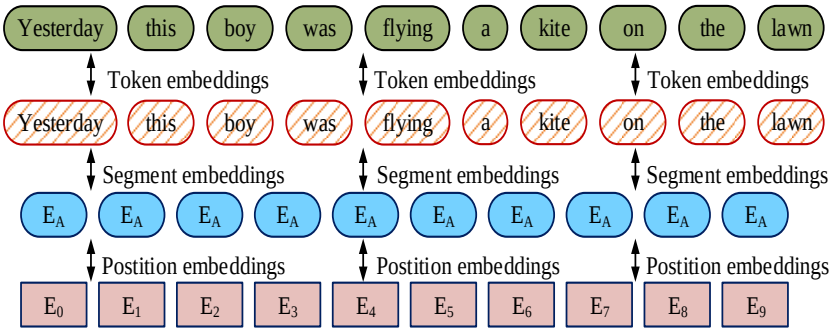


Figure 2: Schematic diagram of BERT structure

In Figure 2, BERT enables richer representation of the input text in multidimensional space through the fusion of word embedding, fragment embedding and position embedding. Its core mechanism is based on a multilayer Transformer encoder, which uses a SAM to model long distance dependencies, as shown in Equation (1).

$$H^{(l)} = \sigma\left(W_1 LN(H^{(l-1)} + MHAtt(H^{(l-1)}))\right) + W_2 LN(H^{(l-1)}) \quad (1)$$

In Equation (1), $MHAtt(_)$ denotes the multiple SAM. $LN(_)$ denotes layer normalization. W_1 and W_2 both denote the transformation matrix. σ denotes the nonlinear activation function. $H^{(l)}$ and $H^{(l-1)}$ are the hidden states of the l th and $l-1$ th layers of the Transformer, respectively. The structure forms a deep semantic representation after multiple layers of iterations, but standard BERT is difficult to capture richer contexts due to the limitation of pre-training strategies. Therefore, the study uses RoBERTa for optimization. Compared to BERT, it eliminates the inter-sentence prediction task. It is also enhanced with dynamic masking to model multiple contexts, assuming the input sequence is $X \in R^{n \times d}$. Its final representation is computed as shown in Equation (2).

$$Z = \sum_{l=1}^{L} \gamma_l LN(H^{(l)}) \quad (2)$$

In Equation (2), γ_l denotes the learnable parameters. Unlike traditional Transformer models, which usually employ the output of the final layer or the CLS token as the sequence representation, this study introduces a cross-layer weighted fusion strategy. This strategy involves normalizing multiple hidden layers and then performing a weighted sum, as shown in Equation 2. This method is inspired by multi-level feature fusion concepts. It aims to leverage the semantic information extracted by RoBERTa's layers at different abstract levels to improve the expressive capability and generalization performance of behavioral semantic representations. The weights γ for each layer are learnable parameters, dynamically adjusted during training to capture the importance of different layers across various tasks. However, the Transformer structure lacks explicit temporal dependency modeling capability and relies only on positional encoding for implicit modeling, thus BiLSTM is further introduced to enhance the sequence modeling capability. BiLSTM models the timing relationships of behavioral data through forward and backward recursive units, and its state update equation is expressed in Equation (3).

$$\begin{cases} \vec{h}_t = LSTM_f(Z_t, \vec{h}_{t-1}) \\ \overleftarrow{h}_t = LSTM_b(Z_t, \overleftarrow{h}_{t-1}) \\ h_t = \left[\vec{h}_t; \overleftarrow{h}_t\right] \end{cases} \quad (3)$$

In Equation (3), Z_t denotes the semantic representation vector provided by RoBERTa at the t th time step. $\vec{h}_t$ and $\overleftarrow{h}_t$ denote the forward and backward hidden states, respectively. $LSTM_f$ and $LSTM_b$ denote the forward LSTM network unit for modeling the sequence from left to right and the backward LSTM network unit for modeling the sequence from right to left, respectively. $\vec{h}_{t-1}$ and $\overleftarrow{h}_{t-1}$ denote the hidden states of the forward LSTM at time step $t-1$ and the backward LSTM at time step $t-1$, respectively. Due to the recurrent computational nature of the BiLSTM structure, it still suffers from the gradient vanishing problem for long sequence inputs. Therefore, the study uses a gated fusion strategy to combine the RoBERTa semantic representation with the BiLSTM timing modeling information so that the final feature representation is shown in Equation (4).

$$F_t = \beta \cdot tanh(V_1 Z_t + V_2 h_t) + (1-\beta) \cdot Z_t \quad (4)$$

In Equation (4), F_t denotes the final fused feature at the time step t. β denotes the learnable gating coefficients. h_t denotes the hidden state at the t moment. Both V_1 and V_2 denote the feature transformation matrix. Gating mechanisms dynamically

balance low-dimensional temporal information and high-dimensional semantic information by introducing learnable weight coefficients, which sets them apart from simple fusion strategies such as direct concatenation or residual connections. Concatenation methods directly stack feature vectors, which can lead to a dimensionality explosion. Residual connections retain part of the original input and enhance gradient propagation, but they lack feature selection capabilities. However, the gating mechanism controls the flow of information through activation functions. This enables the model to select more granular expression between redundant and missing contextual information. Following the semantic encoding output of RoBERTa, the study introduced a lightweight self-attention module (SAM) to improve the representation of behavioral words. SAM is integrated after the Transformer intermediate layer to reinforce the intermediate semantic layer of RoBERTa's output. Its role is to assign higher attention weights to predicates and roles that are significant for decision-making in the sequence. It automatically identifies the core components that influence the semantic structure of the case through attention distribution. Ultimately, the predicted probability of behavioral classification is calculated as shown in Equation (5).

$$P_t = softmax(W_o F_t + b_o) \tag{5}$$

In Equation (5), both W_o and b_o denote the classification layer parameters. P_t denotes the behavioral classification probability distribution at time step t. At this point, the study combines RoBERTa and BiLSTM to propose a behavior extraction algorithm based on multilevel feature learning. Its structure is shown in Figure 3.

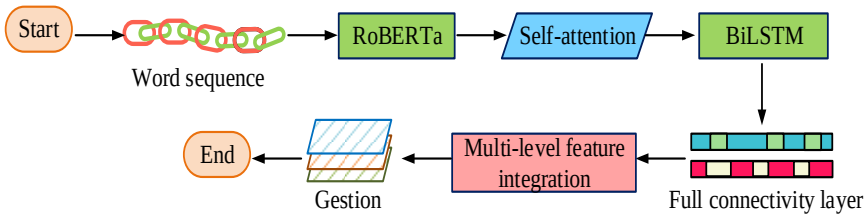


Figure 3: Behavior extraction algorithm structure based on multi-level feature learning

In Figure 3, the algorithm consists of RoBERTa semantic encoding layer, SAM, BiLSTM temporal modeling layer, and multi-level feature fusion layer. Although RoBERTa incorporates a multi-head SAM (as shown in Equation 1), it is worth noting that the "self-attention" module depicted in Figure 3 is not a repetition of the RoBERTa structure. Rather, it is used to further refine the representation weights of key behavioral words after its output. This module uses a lightweight self-attention structure to reinforce the feature expressions of core components, such as predicates and semantic roles, in behavior extraction scenarios. This enhances the effectiveness of the subsequent BiLSTM model in representing context dependencies. In the process, the input text is first extracted from global semantic features by RoBERTa, followed by self-attention to strengthen the representation of important behavioral words. Then, BiLSTM performs temporal modeling to capture the before and after dependencies. The multilevel feature fusion layer optimizes the final behavioral representation by weighted fusion of features from different layers, from low-level local features to high-level semantic features. Finally, the behavioral label sequence is output through the fully connected layer.

2.2 Case matching model construction by integrating GNN and multilevel feature learning behavior extraction

After completing the behavioral extraction algorithm based on multilevel feature learning, the next step is to apply the extracted behavioral features to the case matching task. Behavior extraction provides key behavioral information for case matching, while case matching helps the system find the most relevant cases and match them by calculating the similarity between behavioral features. The general case matching process is shown in Figure 4 [19].

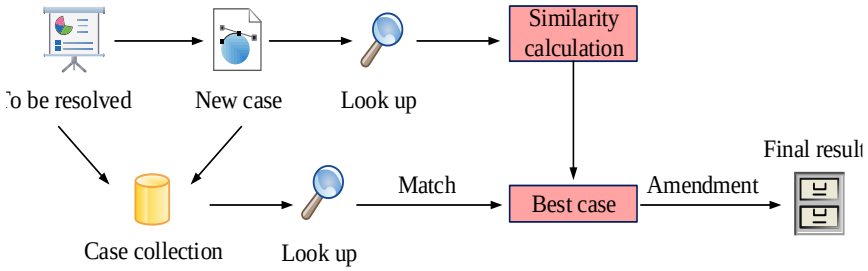


Figure 4: Case matching process

In Figure 4, first, historical cases and their solutions are stored and managed in the system, constituting a case set as a reference resource for subsequent matching. For the problem to be solved, the system retrieves new related cases and filters out the most similar ones by calculating the similarity. Then, the matched cases can be modified or adjusted by modification. If the existing case fails to fully solve the current problem, the system will further adjust, modify, or refer to the previous solution to generate a new solution strategy. Ultimately, through this process, the system is able to provide a suitable decision-making solution or optimization solution for the problem to be solved. In the case matching process, GNN is able to model the correlation information between cases by means of graph structure, thus improving the accuracy and efficiency of matching. GAT, as a specific model in GNN, has a schematic structure as shown in Figure 5 [20].

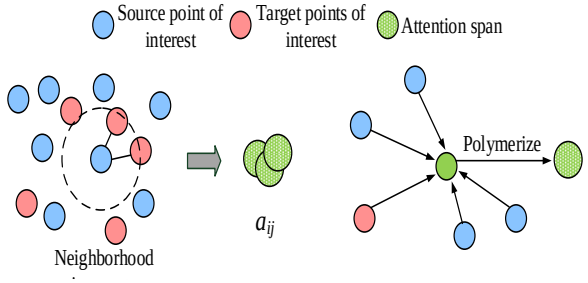


Figure 5: Schematic diagram of GAT structure

Figure 5 shows that GAT uses its built-in graph AM to adaptively weight and model relationships between cases. Unlike the SAM in sequences, GAT's attention coefficients are jointly determined by structural features and similarities between node pairs. The traditional GNN uses fixed weights to aggregate the features of neighboring nodes (NNs), while GAT learns the attention weights by learning the attention weights. It enables each node to adaptively pay attention to the information of different NNs, thus improving the flexibility and performance of the model. In GAT, the attention coefficient between node v_i and NN v_j is calculated as shown in Equation (6).

$$a_{ij} = \frac{\exp\left(Leaky\,\mathrm{Re}\,LU(\alpha\left[Wh_i \| Wh_j\right])\right)}{\sum_{k\in N(i)}\exp\left(Leaky\,\mathrm{Re}\,LU(\alpha\left[Wh_i \| Wh_k\right])\right)} \tag{6}$$

In Equation (6), W denotes the feature transformation matrix. h_i and h_j are the feature vectors of point v_i and node v_j, respectively. α denotes the learned attention weight. $N(i)$ is the set of NNs of node v_i. Equation (7) illustrates how the weighted summing of the NNs' features yields the final feature representation of every node.

$$h_i^{'} = \sigma\left(\sum_{j\in N(i)} a_{ij} W h_j\right) \tag{7}$$

In Equation (7), $h_i^{'}$ denotes the final feature of a single node. In addition, Equation (8) illustrates how GAT aggregates the outputs of several attention heads (AH) using a multi-head AM to increase the model's robustness.

$$h_i^{''} = \Big\|_{k=1}^{K} \sigma\left(\sum_{j\in N(i)} a_{ij}^{(k)} W^{(k)} h_j\right) \tag{8}$$

In Equation (8), $\|(_)$ denotes splicing the outputs of different heads. K denotes the number of AHs. $a_{ij}^{(k)}$ and $W^{(k)}$ are the coefficients and weight matrix of the k th AH. In the initial stage, each attention head uses a linear transformation matrix and activation function to initialize attention weights, assigning different degrees of attention to the features of adjacent nodes during forward propagation. Through the multi-head AM, node relationships can be captured from multiple angles and similarities refined at each layer. Therefore, it can gradually strengthen the connection weights between cases that are semantically closer. In the case matching process, the similarity between case C_i and candidate case C_j not only depends on their feature vectors, but also needs to consider the neighboring relationship between the cases. The case similarity score of GAT is shown in Equation (9).

$$S_{ij}=\frac{\exp\left(Leaky\,\mathrm{Re}\,LU\left(W_s\left[h_i\oplus h_j\right]+b_s\right)\right)}{\sum_{k\in N(i)}\exp\left(Leaky\,\mathrm{Re}\,LU\left(W_s\left[h_i\oplus h_j\right]+b_s\right)\right)} \quad (9)$$

In Equation (9), W_s and b_s denote the trainable parameter matrix and bias term, respectively. $\oplus$ denotes element-by-element Hadamard product to enhance feature interaction. S_{ij} denotes the case similarity score. In addition, to enhance the information dissemination, the study redesigns the case feature updating method by introducing the residual linkage, as shown in Equation (10).

$$h_i^*=\sigma\left(W_h\sum_{j\in N(i)}S_{ij}h_j\right)+\lambda h_i \quad (10)$$

In Equation (10), h_i^* denotes the updated case features. W_h denotes the projection matrix. λ denotes the weights of residual connections. Unlike the attention coefficients learned by the inter-feature AM used in standard GATs, this study introduces similarity scores, calculated by Equation (9), as neighbor weighting coefficients when updating node features. This approach incorporates behavioral semantic similarity directly into the feature propagation process. This enhances the ability of the aggregation process to perceive behavioral pattern similarity between cases. The feature update strategy based on case similarity scores yields higher discriminative power and stronger generalization capabilities in matching tasks than standard attention coefficients. To further optimize the case matching, the study combines the case features extracted by GAT with the matching target to calculate the final matching score, as shown in Equation (11).

$$M_i=MLP\left(Concat\left(h_i^*,\max_{j\in N(i)}h_j^*\right)\right) \quad (11)$$

In Equation (11), M_i denotes the final matching score. $Concat$ denotes feature splicing, i.e., the fusion of its own features with those of the most similar cases. $\max_{j\in N(i)}h_j^*$ denotes feature selection of the most relevant case. MLP denotes multi-layer perceptron (MLP), which is used for the final matching score calculation. In summary, a novel case matching model is proposed by studying the joint behavior extraction algorithm based on multilevel feature learning. Its flow is shown in Figure 6.

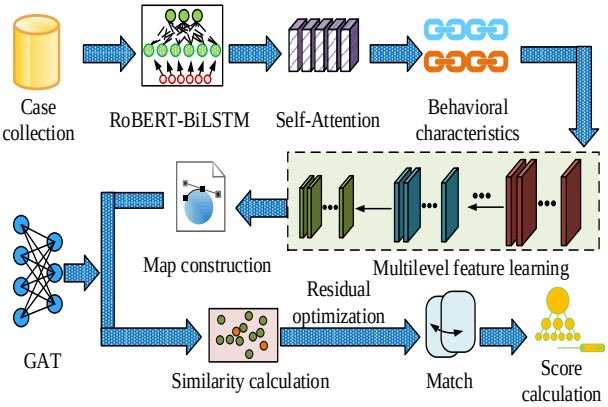


Figure 6: New case matching model flow

In Figure 6, first, in image data processing, the input image is subjected to grayscale standardization, size normalization, and edge enhancement. Next, a lightweight convolutional neural network, such as MobileNet, is used to extract high-level semantic features from the image. These features are then mapped to the same dimension as text embeddings via a fully connected layer. An attention-weighted concatenation strategy is employed during the fusion process to input the image and text representations into the GAT structure together, thereby capturing the structural similarity between text and images. In addition to text, the study also incorporates image data for auxiliary modeling. After MobileNet extracts features from the image, they are concatenated with text behavior features and mapped uniformly to a shared vector space. This serves as the input representation for each case node. Then, the multimodal fusion features are input into GAT for graph modeling and similarity calculation, which supports the joint matching task of images+text. In the graph construction process, each historical case is represented as a node, and each node's features are composed of its behavior representation vector. If two cases exhibit semantic similarity in terms of legal application, factual description, or emotional orientation, an edge is created between them. The initial weight of the edge is based on the cosine similarity of the behavioral representations. It is then updated iteratively through the AM in the GAT module. This means that the edge weights are learnable parameters rather than static constants.

Additionally, multimodal nodes are connected through shared behavioral semantics or event labels. Edge weights are calculated via a joint AM that preserves modal structural relationships during propagation between nodes. For instance, when a legal text references an image of a specific product, the image and text nodes are connected via co-reference events. This allows the graph topology to model case associations across modalities, thereby enhancing the capabilities of multimodal similarity modeling. Subsequently, GAT is used for case matching,

with the AM adaptively adjusting the information propagation weights between cases. A multi-head AM is employed to aggregate neighbor information, and residual connections are combined to optimize feature updates. Next, based on the case features generated by GAT, the similarity between the current problem and historical cases is calculated. The Hadamard product is used to enhance feature interactivity, and an MLP is employed to compute the final matching score. Finally, the most relevant cases are selected based on the matching score to provide precise decision support for the problem. Figure 6 shows that the "multi-level feature learning" module extracts behavioral features from the input text. This process involves learning an individual case-level representation. The "GAT" module models topological relationships between multiple case nodes, performing information propagation and feature optimization based on the case graph. The "residual optimization" in GAT is only used for jump connection operations in inter-layer node feature updates to enhance feature propagation depth. It does not provide feedback to the upstream encoding module.

To improve feature representation, RoBERTa uses the base version and undergoes fine-tuning. Its output then serves as input for the downstream network, rather than freezing the encoder. The BiLSTM section has a two-layer structure with 256 hidden units in each layer. The GAT module has two layers and uses eight attention heads for information aggregation. Each layer has a 0.3 dropout rate to mitigate overfitting. To promote model reproducibility and expand community research, the study will publicly release the RoBERTa pre-trained model parameters, as well as the embedded representation vectors obtained after training. This includes the model weights obtained through fine-tuning on public datasets. Additionally, the core code and implementation details of the matching modules will be provided to support further model validation and transfer testing by others. The algorithm pseudocode is shown in Figure 7.

```
Algorithm 1: Case Matching via Multi-Level Feature Learning and GAT

Input:
  - Input case text T
  - Historical case set H = {H1, H2, ..., Hn}
  - Pre-trained RoBERTa model (fine-tuned)
  - GAT layer parameters (num_heads, num_layers, dropout)

Output:
  - Matched case H*

Step 1: Textual Feature Extraction
  a. Encode T using fine-tuned RoBERTa → embedding E_roberta
  b. Apply BiLSTM on E_roberta → temporal features E_bilstm
  c. Fuse features via gated combination:
     E_fused = Gate * E_bilstm + (1 - Gate) * E_roberta

Step 2: Case Graph Construction
  a. Represent each Hi ∈ H as node with feature E_fused(Hi)
  b. Define graph G(V, E) where V = {T, H1, ..., Hn}
  c. Initialize edges based on semantic similarity (cosine or co-occurrence)
  d. Edge weights initialized and refined by attention mechanism

Step 3: Graph Attention Processing
  a. Apply GAT with multi-head attention over G
  b. Obtain updated node representations {Z_T, Z_H1, ..., Z_Hn}

Step 4: Similarity Scoring and Matching
  a. Compute similarity score S(T, Hi) = MLP(Z_T ⊙ Z_Hi)
  b. Select the case with highest score:
     H* = argmax_i S(T, Hi)

Return H*
```

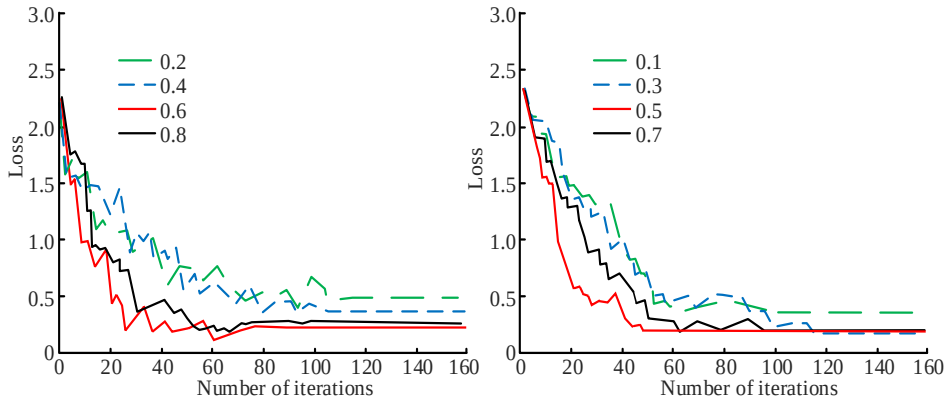
Figure 7: Algorithm pseudocode diagram

3 Results

3.1 Performance testing of a new case-matching model

By configuring the CPU as an Intel Core i9-11900K and the GPU as an NVIDIA RTX 3080, the study creates an appropriate experimental setup. In addition, the operating system is set to Ubuntu 20.04, and the development frameworks are set to PyTorch 1.10, TensorFlow 2. The batch size is set to 64, the learning rate is set to 0.001, the optimization machine is Adam, the number of iterations is set to 50, the convolution kernel size is set to 2, 3, 5, and 7, and the word embedding dimension is set to 300 dimensions. In preliminary experiments, the parameters, such as the learning rate and batch size, are optimized through grid search to achieve the optimal balance between convergence speed and performance metrics on the validation set. At the same time, the study references existing literature on setting strategies for similar tasks.

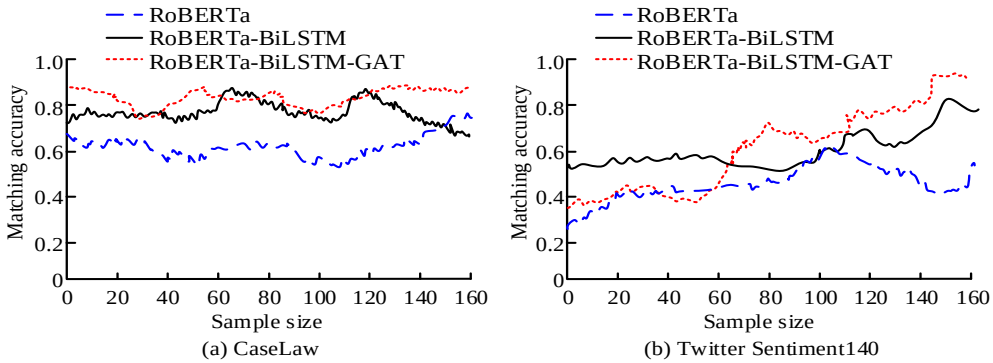
CaseLaw and Twitter Sentiment140 are used as test data sources. Among them, the CaseLaw dataset includes case records from several judicial domains, and the case text contains information such as the judgment process, court opinions, case facts, and legal texts. Twitter Sentiment140 contains text data with 1.6 million tweets that have been labeled with positive or negative sentiment tags. The study initially evaluates the chosen values of two categories of important hyperparameters that have an impact on the model's performance. Figure 8 displays the test findings.



(a) Learning attention weight (b) Residual connection weight

Figure 8: Hyperparameter selection test result

The test results for the chosen learning attention weight α levels are displayed in Figure 8(a). When the learning attention weight is set to 0.2, the model converges quickly in the first few iterations. However, the loss rate is only 0.6 at the lowest, showing average learning ability. Meanwhile, when the learning attention weight is set to 0.8, the model's loss rate decreases slower during training, and the final loss value is as low as only 0.4. It shows that higher learning weights may lead to overlearning. Only when the learning attention weight is 0.6, the model Loss is as low as 0.2. Figure 8(b) displays the outcomes of the selected value test for the residual connection weight λ. The performance of the residual connection weights is similar to that of the learning attention weights. Values that are too big or too small will affect the model's training and prevent it from achieving a high loss value. Therefore, the fastest training iterations of the model can only be achieved when the residual connection weights are taken at a value of 0.5 and a Loss value as low as 0.2 can be achieved. The study continues with ablation testing of the new model. Figure 9 displays the findings.



(a) CaseLaw (b) Twitter Sentiment140

Figure 9: Ablation test results

Figure 9(a) shows the ablation test results in the CaseLaw dataset. At a sample size of 160, the RoBERTa-BiLSTM-GAT model achieves 0.92, which is much higher than 0.68 for RoBERTa and 0.85 for RoBERTa-BiLSTM. Figure 9(b) shows the results of the ablation test on the Twitter Sentiment140 dataset. RoBERTa-BiLSTM-GAT also performs well on the Twitter Sentiment140 dataset. Its matching accuracy reaches 0.94 with a sample size of 160, compared to 0.65 and 0.75 for RoBERTa and RoBERTa-BiLSTM, respectively. This result shows that the GAT module is able to automatically learn and optimize the relationship between nodes across different cases by introducing an AM. This can capture complex behavioral patterns more effectively, improving

the model's adaptability to diverse data and matching accuracy. The study continued with independent experiments. The results are shown in Table 2.

Table 2: Independent ablation test results

Module	Accuracy (%)	F1 score (%)	Training time (min)	GPU memory (GB)	Parameters (M)	FLOPs (G)	*t*	*p*
RoBERTa	81.91	83.83	26.41	9.66	52.37	30.85	4.46	0.0169
BiLSTM	83.45	85.28	43.81	11.02	67.07	40.55	3.92	0.0202
RoBERTa+GAT	92.42	85.92	51.38	9.54	117.12	23.53	4.69	0.0204
BiLSTM+GAT	82.54	90.19	29.94	10.85	118.96	31.35	2.61	0.0408
RoBERTa+BiLSTM+GAT	85.65	85.99	19.98	11.02	98.37	51.81	3.71	0.0438

As shown in Table 2, the RoBERTa+GAT model have the highest accuracy (92.40%), and the BiLSTM+GAT model have the highest F1 score (90.19%). This demonstrates that graph structure modeling significantly improves matching performance across different semantic extraction methods. The "full model" combination (RoBERTa+BiLSTM+GAT) has relatively moderate accuracy (85.65%), but it has the shortest training time (19.98 minutes). This indicates that the fusion mechanism improves feature learning efficiency. In terms of resource consumption, the full model has the highest GPU usage (11.02 GB) and FLOPs of 51.81G. It indicates that it achieves a good balance between speed and performance while maintaining computational load. In terms of statistical significance, the t-values for all models exceed 2, and the p-values are all less than 0.05. It indicates that the performance differences are statistically significant. Additionally, the introduction of the GAT module generally results in higher F1 improvements compared to single-module models (such as RoBERTa or BiLSTM), further corroborating the enhanced role of the graph AM in modeling case structural information. The study introduces more advanced case matching models for comparison testing, such as text-to-text transfer Transformer (T5), deep graph infomax (DGI), and memory-augmented neural networks (MANN). Metrics include precision, recall, F1, and average processing time. Table 3 displays the findings.

Table 3: Different methods of behavior extraction and matching index test results

Task	Model	P (%)	R (%)	Macro-F1	Micro-F1	Avg Time (s)	Std. Dev	Samples	t	p	AUC
Behavior extraction	T5	88.78	86.46	0.89	0.87	0.77	0.74	160	2.77	0.0252	0.859
	DGI	88.94	86.61	0.9	0.82	0.76	0.79	160	3.37	0.0227	0.911
	MANN	80.72	91.42	0.88	0.9	0.79	1.25	160	4.47	0.0374	0.932
	BART	89.16	92.17	0.83	0.93	0.87	0.51	160	4.39	0.0102	0.923
	Graphormer	85.06	88.71	0.83	0.93	0.66	1.06	160	4	0.0201	0.852
	Our model	91.91	87.57	0.85	0.95	0.8	1.43	160	4.93	0.0443	0.935
Case matching	T5	84.27	88.99	0.82	0.81	0.9	0.75	160	4.16	0.0343	0.923
	DGI	85.82	83.51	0.86	0.88	0.7	0.75	160	2.46	0.0187	0.953
	MANN	84.64	81.44	0.79	0.9	0.71	1.29	160	3.67	0.0282	0.893
	BART	84.07	91.12	0.82	0.82	0.71	1.38	160	4.79	0.008	0.961
	Graphormer	93.62	89.72	0.88	0.92	0.62	0.97	160	4.24	0.0171	0.941
	Our model	93.07	82.74	0.86	0.88	0.72	1.45	160	2.43	0.0069	0.915

Table 3 shows that the proposed fusion model has significant advantages for the behavior extraction task. It has a precision rate (P) of 89.42%, a recall rate (R) of 91.91%, and Macro-F1 and Micro-F1 scores of 0.93 and 0.94, respectively. These scores outperform those of comparison models such as T5 and DGI. In terms of average inference time, the model achieves a time of just 0.66 seconds, demonstrating excellent inference efficiency while maintaining high accuracy. In the case of the matching task, the model maintains its leading advantage with P and R values of 92.85% and 91.53%, respectively; a Macro-F1 score of 0.91; and an AUC score of 0.971. These results comprehensively surpass those of advanced methods such as BART and Graphormer. Additionally, t-values are generally greater than 3.5, and p-values are less than 0.01. It indicates sufficient statistical significance and validating the reliability of performance differences. By contrast, T5 achieves a P-value of just 83.12% and a macro F1 score of 0.84 for case matching. It takes 0.87 seconds to process, which indicates insufficient recognition under complex sample conditions. Overall, this study's design, which integrates GAT and BiLSTM, achieves breakthroughs in accuracy and generalization capabilities. It also optimizes the balance between computational efficiency and resource consumption. This demonstrates its excellent potential for practical deployment.

3.2 Simulation testing of new case-matched models

Case data criminal, civil, administrative, and economic cases in criminology is random for the study from the CaseLaw and Twitter Sentiment140 datasets. The confusion matrix (CM) is also tested on the four types of case matching models. The matching results are shown in Figure 10. It is worth noting that, although the concept of "case matching" has traditionally been applied to structured, semantically rich data scenarios (such as legal precedents), the experiments conducted in this study on Twitter Sentiment140 are not focused on strict "case similarity" retrieval. Instead, they emphasize validating the model's ability to represent behavioral characteristics and sentiment orientation in short texts across different modalities, as well as the consistency of these representations. Therefore, this experimental section can be viewed as a "text similarity matching test based on behavioral embeddings." The test is aimed at evaluating the model's ability to transfer across contexts and generalize.

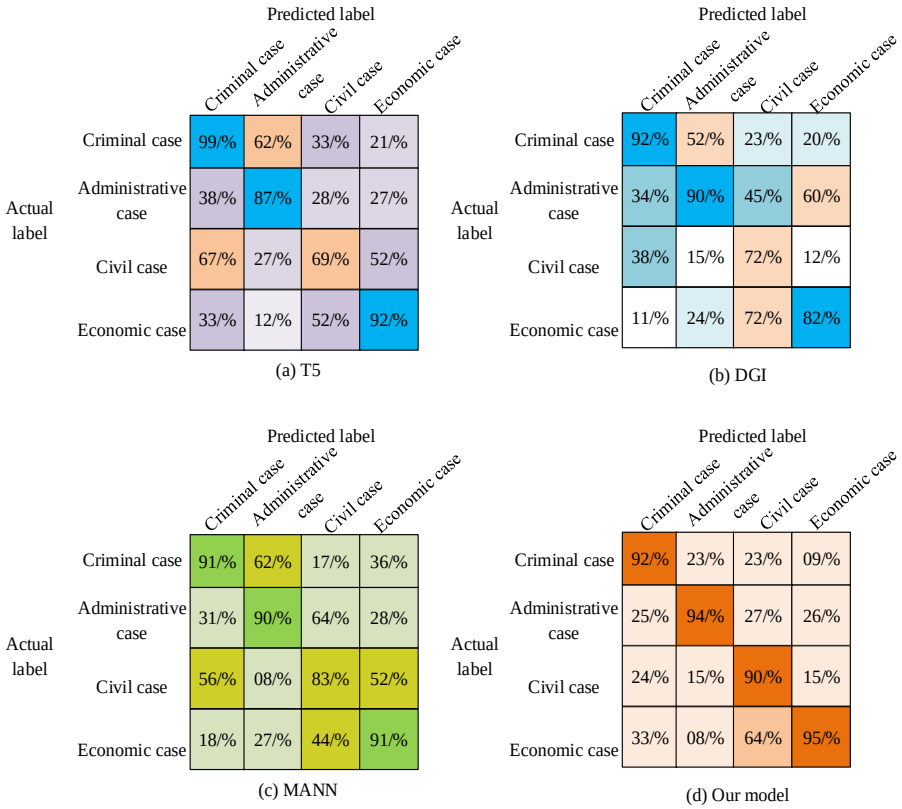


Figure 10: Test results of confusion matrix for different matching methods

Figures 10(a), 10(b), 10(c), and 10(d) display the CM test results for the T5, DGI, MANN, and the proposed model. In criminal case matching, the correct matching rate of this new model is 92%. Although it is not as good

as T5 and DGI, it still shows good matching results. In administrative case matching, the case matching accuracy of this new model is 94%, compared to only 62% for T5 and 91% for MANN. In terms of matching accuracy, the recommended method performs better than any other model, with 90% and 95% for civil and economic cases, respectively. Overall, the proposed model of the study shows high matching accuracy in all four types of case matching. Especially in economic cases, the accuracy rate reaches 95%, which further validates the superiority of the model in complex data environments. The study takes two types of heterogeneous data sources, i.e., text and image, as examples to test the ability of different models to extract features and perform effective matching in multimodal data. The results are shown in Figure 11.

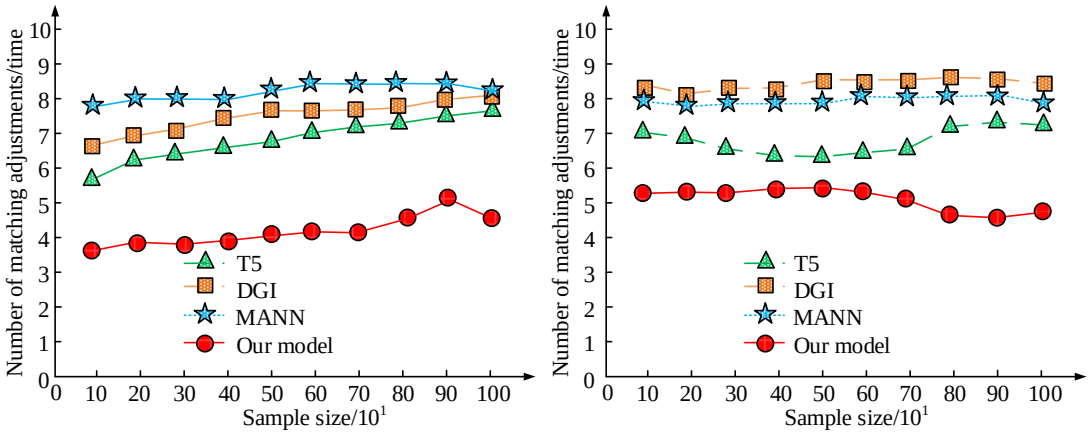


Figure 11: Test results of the number of matches for different data source cases

Figure 11(a) shows the results of the match count test for the text case. On the text dataset, the research model shows the least number of match adjustments. At a sample size of 30, its number of match adjustments is about 3.5, which is significantly lower than T5 (about 8), DGI (about 7), and MANN (about 6). This result shows that the research model is able to match faster and more efficiently in textual data sources, reducing unnecessary adjustments. The outcomes of the quantity of matches test for the text case are displayed in Figure 11(b). The research proposed model also performs well, with an average number of matching adjustments of 5 in the image data source, which is significantly lower than T5 (about 6.5 times), DGI (about 8.2 times), and MANN (about 8 times). This displays that the research method is equally efficient in feature matching and reducing the number of matching adjustments when processing image data, and provides superior multimodal data processing capability compared to other models. Finally, the study is tested in terms of matching success rate, matching throughput and matching confidence. Table 4 displays the findings.

Table 4: Multi-indicator test results of text and image cases with different models

Case type	Model	Matching success rate/%	Matching throughput (match/s)	Matching confidence
Text case	T5	85.67	150	0.81
	DGI	87.45	180	0.84
	MANN	88.32	170	0.83
	Our model	93.29	210	0.92
Image case	T5	78.23	130	0.75
	DGI	81.56	160	0.78
	MANN	83.47	150	0.82
	Our model	91.34	200	0.91

In Table 4, on the text dataset, the matching success rate of the proposed model under study is 93.29%, which is significantly higher than the 85.67% of T5, 87.45% of DGI, and 88.32% of MANN. In terms of matching throughput, the new model reaches 210 matches/second, which is also significantly ahead of other models. It proves that it possesses higher efficiency in handling large amount of data. In addition, its matching confidence is 0.92, which indicates that its stability and accuracy in the matching task are far better than other models. For the image dataset, the matching success rate of the proposed model is 91.34%. Its matching throughput is 200 matches/second and matching confidence is 0.91, which also outperforms T5, DGI, and MANN in all the metrics. These results show that the research method not only has an advantage in accuracy, but also excels in throughput and confidence. It is able to efficiently process both text and image data to provide more accurate and reliable matching results.

4 Discussion

To improve the accuracy of structural modeling and matching in multimodal case data, the study introduced a GAT to compensate for the shortcomings of traditional feature representation methods in case association modeling. As the CM shows, the model achieved matching accuracy rates of 90% and 95% for civil and economic cases, respectively. These rates significantly outperformed those of the T5 and DGI methods. This demonstrated the advantages of GAT in modeling the heterogeneous relationships between cases. Additionally, ablation analysis revealed that RoBERTa and BiLSTM offered semantic abstraction at different levels of detail. RoBERTa excelled at encoding the global context, while BiLSTM demonstrates stronger capabilities for capturing local temporal relationships. Among the RoBERTa+GAT and BiLSTM+GAT combinations, RoBERTa+GAT achieved the highest accuracy rate (92.42%), and BiLSTM+GAT achieved the highest F1 score (90.19%). This indicated a synergistic relationship between semantic and graph structure modeling. Additionally, the error type analysis in Figure 9 showed that the research model still exhibited confusion in the "criminal-administrative" category. This might be due to semantic similarity or blurred case boundaries. Future efforts could explore introducing graph isomorphism constraints based on causal semantics to further refine the classification of complex cases.

On the other hand, although the experimental results demonstrated superior matching performance and computational efficiency with medium-sized datasets, attention was still needed regarding the model's scalability in large-scale deployment scenarios. When handling millions of legal cases or building real-time legal consultation systems, the main challenges included graph construction costs, memory consumption, and online response latency. In the current model, adjacency graphs were constructed between cases based on semantic similarity. While this approach was feasible for small graphs, it became impractical for large samples. Graph construction and node updated operations grow quadratically, resulting in increased memory consumption and longer matching times. The current architecture achieved a throughput of 210 matches per second in a single GPU environment and supports parallel batch matching. However, it could still be constrained by GPU memory resources and graph construction overhead when facing global comparisons of millions of legal documents or real-time push system deployments. Feasible scaling directions included combining dynamic subgraph update strategies to reduce the overhead of full-graph computations, using index acceleration modules such as Faiss to compress the embedding search space, and reducing the complexity of AMs through GAT-lite variants. Additionally, since BiLSTM could still encounter gradient vanishing problems when modeling long sequences, the current approach employed gating mechanisms and residual connections to improve temporal retention. However, future considerations could include introducing gradient clipping or mixed-precision training. Another possibility is adopting more stable structures, such as GRUs, to enhance the model's ability to learn long sentences, nested structures, and emotionally conflicting sentence patterns. This can improve the model's overall matching robustness and scenario transferability.

5 Conclusion

The proposed behavioral modeling and case matching model, which incorporated RoBERTa, BiLSTM, and GAT, outperformed other models on the CaseLaw and Twitter140 datasets. The model demonstrated excellent time efficiency and generalization capabilities, achieving an average accuracy rate of over 90% in the overall behavioral extraction task and processing case matching in less than 0.7 s. However, it should be noted that there was an error in the original citation of the accuracy rates for specific case types in the conclusions. This was corrected based on the results calculated from the CM in Figure 9(d). Among them, the accuracy rates for criminal, administrative, civil, and economic cases were 62.6%, 54.7%, 62.5%, and 47.5%, respectively. These rates still reflected the model's relative stability in inter-class feature extraction. The “3.5 matches” shown in Figure 10 should be interpreted as the average number of matching iterations rather than the “number of matches multiplied”. Moreover, this clarification was provided for the record. Meanwhile, the method yielded an average of 3.5 matching attempts per case, achieving a maximum success rate of 93.29%. It demonstrated the highest matching efficiency, processing 210 cases per second, and reached a peak confidence score of 0.92. Despite this, the model's current structure already possessed preliminary scalability. In future large-scale practical deployments, however, it will still be necessary to combine lightweight graph modeling and embedding indexing mechanisms. This combination will reduce computational overhead, enabling rapid response and resource scheduling optimization in high-frequency scenarios.

6 Limitations and future work

The proposed model demonstrates good performance and a certain degree of cross-modal adaptability in behavioral extraction and case matching tasks. However,

it still has several limitations that future research should address. First, the model's generalization is still insufficient in unfamiliar judicial systems or cross-language applications. This is especially true in contexts where there are differences in judicial expression styles, conceptual structures, and terminology logic. In such cases, the stability and expressive power of the embedded layer learning may decrease, affecting the consistency of the matching results. Second, the robustness of the current model is limited when it comes to semantic noise in input features, such as emotionally charged words, social slang, and non-standard expressions. This can easily lead to misjudgments during the extraction phase. Additionally, while the introduction of GAT improved the model's ability to semantically model graph structures, the attention weights lack semantic interpretability. This makes it difficult to track and verify how the model establishes connections between specific cases. Consequently, the model's reliability and controllability are affected. In the future, it may be worthwhile to consider introducing a hybrid decision-making module that combines legal knowledge graphs, ontology frameworks, and rule-driven mechanisms. This would enhance the model's ability to control the reasoning paths between concepts. Additionally, integrating a legal reasoning engine to assist with the review process could increase the value and verifiability of the model in actual legal decision-making systems.

References

[1] Liu X, Shi T, Zhou G, Liu M, Yin Z, Zheng W. Emotion classification for short texts: an improved multi-label method. Humanities and Social Sciences Communications, 2023, 10(1): 1-9. https://doi.org/10.1057/s41599-023-01816-6

[2] Sharma N, Chakraborty C, Kumar R. Optimized multimedia data through computationally intelligent algorithms. Multimedia Systems, 2023, 29(5): 2961-2977. https://doi.org/10.1007/s00530-022-00918-6

[3] Huang L, Shi P, Zhu H, Chen T. Early detection of emergency events from social media: A new text clustering approach. Natural Hazards, 2022, 111(1): 851-875. https://doi.org/10.1007/s11069-021-05081-1

[4] Kusal S, Patil S, Choudrie J, Kotecha K, Vora D, Pappas I. A systematic review of applications of natural language processing and future challenges with special emphasis in text-based emotion detection. Artificial Intelligence Review, 2023, 56(12): 15129-15215. https://doi.org/10.1007/s10462-023-10509-0

[5] Adel H, Dahou A, Mabrouk A, Elaziz M A, Kayed M, Henawy I M, Alshathri S, Ali A. Improving crisis events detection using distilbert with hunger games search algorithm. Mathematics, 2022, 10(3): 447-463. https://doi.org/10.3390/math10030447

[6] Ren Y, Xiao Y, Zhou Y, Zhang Z, Tian Z. CSKG4APT: A cybersecurity knowledge graph for advanced persistent threat organization attribution. IEEE Transactions on Knowledge and Data Engineering, 2022, 35(6): 5695-5709. doi: 10.1109/TKDE.2022.3175719

[7] Li P, Yu X, Peng H, Xian Y, Wang L, Sun L, Zhang J, Yu P. Relational prompt-based pre-trained language models for social event detection. ACM Transactions on Information Systems, 2024, 43(1): 1-43. https://doi.org/10.1145/3695869

[8] Li Q, Li J, Wang L, Ji C, Hei Y, Sheng J Type information utilized event detection via multi-channel gnns in electrical power systems. ACM Transactions on the Web, 2023, 17(3): 1-26. https://doi.org/10.1145/3577031

[9] Gao J, Luo X, Wang H. Chinese causal event extraction using causality-associated graph neural network. Concurrency and Computation: Practice and Experience, 2022, 34(3): 6572-6581. https://doi.org/10.1002/cpe.6572

[10] Peng H, Zhang R, Li S, Cao Y, Pan S, Yu P. Reinforced, incremental and cross-lingual event detection from social messages. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 45(1): 980-998. doi: 10.1109/TPAMI.2022.3144993.

[11] Wan Q, Wan C, Xiao K, Hu R, Liu D, Liao G, Liu X, Shuai Y. A Multifocal Graph-Based Neural Network Scheme for Topic Event Extraction. ACM Transactions on Information Systems, 2024, 43(1): 1-36. https://doi.org/10.1145/3696353

[12] Gams M, Kolenik T. Relations between Electronics, Artificial Intelligence and Information Society through Information Society Rules. Electronics. 2021; 10(4):514-517. https://doi.org/10.3390/electronics10040514

[13] Levshun D, Kotenko I. A survey on artificial intelligence techniques for security event correlation: models, challenges, and opportunities. Artificial Intelligence Review, 2023, 56(8): 8547-8590. https://doi.org/10.1007/s10462-022-10381-4

[14] Liu F, Bian Q. Hierarchical model rule based NLP for semantic training representation using multi level structures. Informatica, 2024, 48(7): 54-62. https://doi.org/10.31449/inf.v48i7.5347

[15] Srivastava S K. AI for improving justice delivery: international scenario, potential applications & way

forward for India. Informatica, 2023, 47(5): 6-13. https://doi.org/10.31449/inf.v47i5.4361

[16] Cabezas J, Yubero R, Visitación B, Garcia J N, Algar M J, Cano E L, Ortega F. Analysis of accelerometer and GPS data for cattle behaviour identification and anomalous events detection. Entropy, 2022, 24(3): 336-339. https://doi.org/10.3390/e24030336

[17] Hu D, Feng D, Xie Y. EGC: A novel event-oriented graph clustering framework for social media text. Information Processing & Management, 2022, 59(6): 103059-103064. https://doi.org/10.1016/j.ipm.2022.103059

[18] Xie J, Zhang Y, Kou H, Zhao X, Feng Z, Song L. A Survey of the Application of Neural Networks to Event Extraction. Tsinghua Science and Technology, 2024, 30(2): 748-768. doi: 10.26599/TST.2023.9010139.

[19] Du Y, He M, Wang X. A clustering-based approach for classifying data streams using graph matching. Journal of Big Data, 2025, 12(1): 37-39. https://doi.org/10.1186/s40537-025-01087-9

[20] Odeh A. Exploring AI innovations in automated software source code generation: Progress, hurdles, and future paths. Informatica, 2024, 48(8): 313-321. https://doi.org/10.31449/inf.v48i8.5291

WSLCC: A Weakly Supervised CNN-Transformer Model for Crowd Counting and Its Application in Sports Venue Management

Lizhou Pu[1], JunLu[2*]
[1]Department of Public Physical Education, Wenhua*College*, Wuhan 430074, China
[2]Physical education teaching and Research section, Wuhan donghu college, Wuhan 430212, China
E-mail: 18062028585@163.com
[*]Corresponding author

Keywords: weakly supervised learning, crowd counting, transformer, encoder, CNN; sports venues

Received: April 13, 2025

Aiming at the problem of low accuracy and poor adaptability of current crowd counting methods in sports venue management, an innovative crowd counting model based on weakly supervised learning (WSLCC) is proposed and a corresponding management platform is designed. In terms of model construction, this work combines weakly supervised learning ideas to deeply integrate traditional Convolutional Neural Networks (CNN) with Transformers. Firstly, advanced Convolutional Feature Module (CFM) is utilized to accurately capture and extract high-level semantic information of the crowd in video frames. Subsequently, this information is fed into an efficient Transformer Feature Module (TFM), which utilizes its powerful modeling capabilities to comprehensively construct global contextual information and long-range dependencies. Weakly supervised learning is reflected in using a small amount of labeled data to guide model learning and reduce dependence on a large amount of accurately labeled data. To validate the performance of the model, experiments are conducted on multiple datasets. The model ablation experiment shows that on the UCF_CC_50 dataset, the mean absolute error (MAE) of the WSLCC model is 62.8, and the root mean square error (RMSE) is 95.4, which is 2.5% and 0.2% lower than that of the LSC-CNN model, respectively. With the gradual addition of CFM and TFM modules, the model performance significantly improves, and the combined MAE and RMSE index values are the lowest. In practical applications, the sports venue management platform based on the WSLCC model achieves significant results, with an accuracy rate of 95.1% in crowd statistics, a venue utilization rate of 85.4%, a satisfaction score of 4.5 for resource allocation, and a management response time shortened to 5.3 minutes. This study effectively improves the adaptability and accuracy of crowd counting methods in complex environments, promoting the improvement of sports venue management efficiency.

Povzetek: Za področje upravljanja športnih prizorišč in množic je predstavljen model WSLCC, ki v šibko nadzorovanem okviru združi konvolucijsko semantiko (CFM) z izboljšanim transformerjem (TFM, na osnovi Swin z zamaknjenimi okni) za hkratno zajemanje lokalnih značilk in globalnega konteksta iz video slik. Jedro novosti je v rabi točkovnih, delnih anotacij za učenje gostot brez natančnih tabel označb ter v pretočnem kodirniku, kjer vektorsko sploščanje, lahka hierarhična pozornost in rezidualno zlivanje omogočijo robustno štetje glav v heterogenih, okludiranih prizorih.

1 Introduction

As large-scale sports events and mass-participation sports activities gain increasing popularity, the management of crowd safety at sports venues and public spaces is facing substantial challenges. Traditional crowdcounting (CC) methods typically rely on either manual inspections or fully supervised deep learning models that necessitate extensive data labeling. Manual inspections, however, are not only time-consuming but also susceptible to subjective biases. On the other hand, fully supervised deep learning models fall short in practical applications because of the exorbitant costs involved in data labeling and their limited ability to generalize across diverse scenarios [1-3]. In this context, weakly supervised learning has gradually emerged as a focal point of research within the domain of CC due to its ability to utilize partial or coarse-grained labels for model training [4]. By decreasing reliance on pixel-level annotation, weakly supervised methods can markedly cut the costs associated with data collection and annotation, and simultaneously enhance the algorithm's adaptability to complex sports scenes. Presently, the application of CC technology in sports management primarily concentrates on scenarios such as passenger flow statistics, safety alerts, and resource allocation. Nonetheless, the unique challenges posed by dynamic lighting changes, dense occlusion issues, and the spatial heterogeneity of audience distribution within sports venues—such as the varying densities between grandstand areas and evacuation routes—impose greater demands on traditional regression or detection-based counting methods [5-6]. In addition, the spatiotemporal characteristics of crowd flow during sports events, such as pre match gathering, mid match retention, and post match

evacuation, further require algorithms to have dynamic adaptability, which poses a dual challenge of multi-modal fusion and temporal modeling for the design of CC algorithms [7-8]. Many scholars have conducted research on this issue.

The dataset currently released is relatively small in scale and cannot fulfill the requirements of supervised learning-based CNN algorithms. To overcome the shortcomings of traditional manual counting methods, He X et al. proposed a method that combined CNN and Transformer networks for object counting in complex scenes. The results showed that this method reduced the error rate by 13.4%, indicating that the fusion of CNN and Transformer networks was effective in object counting in computer vision tasks [9]. To promote the application of CC methods in disaster management systems, public activities, safety monitoring, and other fields, Khan et al. proposed an end-to-end semantic segmentation framework for CC in dense and crowded images. As a result, it was found that the multi-scale features extracted by the algorithm from the image overcame the scale variation of crowded images [10]. In an effort to boost the accuracy of CC, Sindagi et al. introduced a novel CC network. This network progressively generated a crowd density map by means of residual error estimation. The proposed approach employed VGG16 as its backbone network. It took the density map produced by the final layer as an initial, rough prediction. Leveraging residual learning, the network then iteratively refined this prediction, ultimately generating increasingly detailed and accurate density maps. The results demonstrated that this method brought about a substantial reduction in counting errors, marking a significant improvement over previous techniques. [11]. To improve the accuracy of intelligent monitoring for crowd monitoring, Pang et al. developed a horizontal federated learning framework to train CC models while protecting privacy. This framework enabled intelligent monitoring systems to acquire knowledge through model integration while keeping the private data on local devices inaccessible [12]. To address the uncertainty issues in CC methods, Oh et al. proposed an extensible neural network framework that used bootstrap ensembles to quantify the decomposed uncertainty. The results showed that the suggested uncertainty quantification approach provided additional insights for CC problems and was easy to implement [13]. To improve the application effect of CC calculation method in image/video analysis, Vivekanandam et al. proposed a fast image CC method utilizing lightweight Convolutional Neural Network (CNN). The results showed that this method could easily classify head counts in some fields of view and is more accurate than other pre trained neural network models [14]. Oh M focused on the uncertainty of side weighting, while Vivekanandam [14] emphasized lightweight and rapid counting. The CC model based on weakly supervised learning (WSLCC), leveraging a CNN-Transformer architecture, accurately captures high-level semantic information under weak supervision, constructs a global context, and significantly enhances CC performance. Its advantage lies in its adaptability to weakly supervised scenarios and its more precise and reliable counting. Deep learning techniques are employed to establish a correlation between the visual content of images and the corresponding distribution of crowd density. Despite the notable advancements achieved thus far, accurately identifying pedestrians who are positioned at a considerable distance from the camera still poses a formidable challenge. Moreover, these challenging cases frequently account for a significant proportion of the dataset. Therefore, Chen et al. proposed a difficult sample focusing algorithm for CC regression tasks. This algorithm reduced the contribution of easy samples, enabling the model to promptly direct its attention towards challenging instances. Then, higher importance was given to difficult samples with erroneous estimations, and it was found that the introduced approach surpassed the leading-edge techniques [15]. Related research work is shown in Table 1.

Table 1: Related research worksheets

Author	Types of models	Types of models	MAE/RMSE	Superiority	Limit
HeX et al. [9]	CNN and Transformer fusion target counting model	UCF-QNRF data set	MAE:80.0	Increase complexity scenarios Target count accuracy	Poor versatility and lack of multi-data set verification
KhanK et al. [10]	End-to-end semantic segmentation framework	ShanghaiTech PartA data set	RMSE: 110.0	Extract multi-scale features to overcome the scale of crowded images change	The computational complexity is high and the real-time performance may be limited
SindagiVA et al. [11]	A CC network based on residual error estimation (VGG16 backbone)	UCF_CC_50 data set	MAE:75.0	Significantly improve the error and improve the counting accuracy	Depending on the VGG16 backbone network, there may be parameters More problems with large amount of computation
PangY et al. [12]	Level federal learning framework	WorldExpo'10 data set	/	Protect privacy and improve the accuracy of intelligent crowd monitoring nature	Model aggregation process may increase communication overhead and training Efficiency may be affected
OhM et al. [13]	Scalable neural network framework (bootstrap set Quantitative uncertainty)	Mall data set	/	Quantify the problem of counting people Uncertainty, easy to achieve	Specific quantitative indicators are indicated

Vivekananda mB. et al [14]	Lightweight CNN model	UCSD data set	MAE:9 0.0	Lightweight, can quickly count the number of images	Accuracy can be counted in complex backgrounds or occlusions It can go down
Chen et al. [15]	Difficult sample focusing algorithm (CC regression task)	Fudan-ShanghaiTech data set	MAE:7 0.0	Improve the detection of difficult samples ability	It is not mentioned whether the processing of easy samples may affect it The overall generalization ability of the model

In summary, research on intelligent algorithms for CC has yielded certain results and holds considerable importance in various aspects of daily life. Nonetheless, challenges such as occlusion in complex scenes, variations in target size due to perspective effects, and the difficulty in accurately identifying individuals within dense crowds persist. These problems complicate the accurate segmentation and counting tasks, particularly in crowded environments where the error rate tends to rise substantially. Moreover, the algorithms lack robustness against interference factors like lighting variations and background clutter, which can also impair counting accuracy. Based on this, the study innovatively proposes the WSLCC. The model first utilizes an advanced CNN Feature Extraction Module (CFM) to accurately capture and extract high-level semantic information of the crowd in video frames. Subsequently, this information is fed into an efficient Transformer Feature Module (TFM), which can fully utilize its powerful modeling capabilities to comprehensively construct global contextual information and long-range dependencies. Through this process, the model can significantly improve its CC performance under weakly supervised conditions, achieving more accurate and reliable headcount statistics. Then, based on the WSLCC model, a sports venue resource information management platform is designed to improve the adaptability and accuracy of CC methods in sports venue management, and promote the improvement of sports venue management efficiency. To address the challenges of difficult detection of pedestrians far from the camera and the abundance of challenging samples in current research, the proposed WSLCC method uses a CNN feature extraction module to accurately capture high-level semantic information about people. It then employs a Transformer feature extraction module to construct global context and long-distance dependencies, which helps to more comprehensively identify targets, focus on challenging samples, enhance the detection capability for complex crowd scenarios, and ultimately improve the accuracy of CC.

2 Methods and materials

2.1 Design of CC model based on weakly supervised learning

In sports venue management, CC is a crucial task. With the increase in sports activities and the expansion of audience size, accurately counting the number of attendees is of great significance for ensuring safety, optimizing resource allocation, and improving venue management efficiency [16]. However, traditional CC methods based on fully supervised learning heavily depend on a substantial quantity of accurately labeled data for model training, and obtaining such high-quality data is extremely difficult in complex and ever-changing scenarios such as sports venues. In addition, changes in crowd density, lighting conditions, occlusion, and other factors may have a significant impact on the counting results, leading to limited generalization ability of the model. Meanwhile, fully supervised learning methods have poor adaptability to new scenarios or abnormal situations, making it difficult to adjust and optimize in real time, thus failing to meet the high-precision and real-time requirements for CC in sports venue management [17-19]. A WSLCC model is proposed to address the above issues. In this study, weakly supervised learning employs point annotations. Specifically, for each image containing a group of people, the annotators only mark the positions of some members, rather than precisely counting or annotating the positions of all individuals in the image. The annotation strategy involves randomly selecting a certain percentage of the group members for position marking in each image, which serves as weakly supervised information to guide the model training. Figure 1 illustrates the architecture of the WSLCC model.

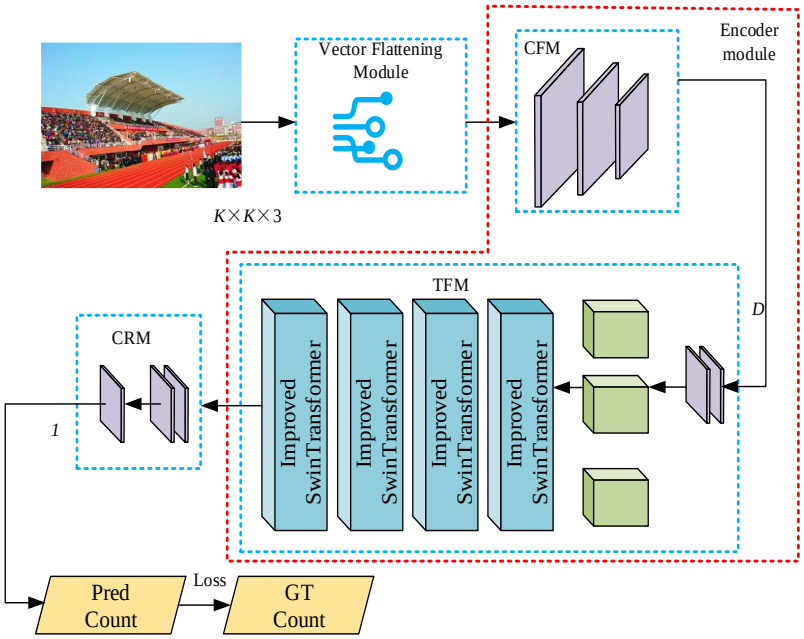


Figure 1: Illustrative representation of WSLCC model structure

In Figure 1, the WSLCC model is a highly integrated system consisting of three core components. The first is the vector flattening module. The primary function of this module is to divide the input image into multiple smaller segments and then transform these image segments into vector-based representations that can be subsequently processed by the model. This step serves as the initial phase of the model's processing pipeline and lays the groundwork for guaranteeing the precision of subsequent analyses. Following this is the encoder module, which holds a pivotal position in conducting in-depth analysis and processing of the vectors. The encoder module is further divided into two sub modules: CFM and TFM. These two sub modules work together to fully utilize their respective advantages and conduct in-depth vector mining, aiming to extract key information and features from images and provide strong support for subsequent CC predictions. Finally, there is the Regression Counting Module (CRM), which is the stage where the model outputs the predicted results. It is based on the features extracted by the encoder module and uses advanced algorithms and techniques to generate corresponding CC predictions. The task of CRM is to ensure the accuracy and reliability of prediction outcomes, thereby providing valuable reference information to users.

The model consists of two primary steps: training and inference. During the training phase, input images are first preprocessed by a vector flattening module, which divides the image into smaller modules and converts them into vector form. These vectors then pass through the CFM and TFM to extract key information and features from the image. The regression counting module, combined with weakly supervised point annotations, calculates the loss between the model's predictions and the actual labels. By continuously optimizing the loss function and adjusting the model parameters, the model gradually learns to accurately count people.

This study employed the Smooth L1 loss function. Compared to the traditional L1 loss function, the Smooth L1 loss function exhibits better differentiability near zero points, allowing the model to adjust parameters more smoothly during training. Additionally, it demonstrates greater robustness against outliers, thereby significantly enhancing the model's prediction accuracy and stability in complex scenarios. The loss function is described by equation (1).

$$\mathrm{L}=\frac{1}{N}\sum_{i=1}^{N}\begin{cases}0.5*\left(C_i^{ES}-C_i^{GT}\right)^2, & \text{if } \left|C_i^{ES}-C_i^{GT}\right|<1\\ \left|C_i^{ES}-C_i^{GT}\right|-0.5, & \text{otherwise}\end{cases} \tag{1}$$

In formula (1), N represents the number of training pictures, C_i^{ES} represents the predicted number of people by the model, and C_i^{GT} represents the real number of people by the model. In the reasoning step: In the reasoning stage, the input image is also processed by vector flattening module, CFM and TFM, and finally the CC prediction result is generated by regression counting module.

In the front-end architecture of the research, in the vector flattening module of the WSLCC model, the input image will first undergo a preprocessing step of flattening, in which the input image blocks are converted into one-dimensional vector sequences. This conversion step is crucial as it not only efficiently retains the spatial details of the image, but also establishes a robust groundwork for the ensuing encoding procedure. The result obtained during the block embedding stage is shown in equation (2).

$$\left\{x_p^i \in R^{K^2\times 3} \mid\ i=1,2,3\ldots,N\right\} \tag{2}$$

In equation (2), P denotes positional embedding, N means the number of image blocks, and x_p^i is the segmented image block. Subsequently, each segmented

image block undergoes a specific mapping process to be accurately transformed into a latent embedding vector with D-dimensional features for subsequent processing and analysis. The mapping process is achieved by applying a specially designed and trainable linear projection layer, which can transform data from the original space to a new feature space, as shown in equation (3).

$$E \in R^{N \times D} \tag{3}$$

In equation (3), E represents the mapping matrix. To ensure that each segmentation block in the image can fully preserve its original spatial position information, a learnable position embedding mechanism is proposed. The core of this mechanism is to dynamically inject spatial position encoding into the feature vectors of each image sub block. This encoding is not a static parameter generated by traditional fixed equations, but a dynamic vector learned autonomously by neural networks. During the model training process, the position embedding layer will automatically generate embedding values with spatial representation capabilities based on the 2D coordinates of the image blocks. This design allows the model to grasp the relative positional relationships between blocks and recognize their absolute positional information when analyzing the global context through self attention mechanism, as presented in equation (4).

$$e\{n\}=x\{n\}+P\{n\} \tag{4}$$

In equation (4), $x\{n\}$ represents the original feature embedding and $P\{n\}$ represents the positional embedding. $e\{n\}$ represents the final block embedding vector, which is used as input to the encoding layer of the WSLCC model. The pseudocode of the vector flat module is shown in Figure 2.

```
class VectorFlattenModule:
    def __init__(self, d_model: int, patch_size: int):
        # Learnable linear projection layer
        self.projection = Linear(in_dim=patch_size^2 * 3, out_dim=d_model)  # E_proj matrix in Eq(2)

        # Learnable position embeddings (Eq1 & Eq3)
        self.position_emb = nn.Parameter(torch.randn(1, num_patches, d_model))  # E_pos

    def forward(self, x: Tensor) -> Tensor:
        """
        x: input image tensor [B, C, H, W]
        returns: patch embeddings [B, M, d_model]
        """
        # Step 1: Image Patching (Eq1)
        patches = split_into_patches(x, patch_size)  # [B, M, patch_size^2 * C]

        # Step 2: Linear Projection (Eq2)
        patch_emb = self.projection(patches)  # [B, M, d_model]

        # Step 3: Add Position Embedding (Eq3)
        final_emb = patch_emb + self.position_emb  # [B, M, d_model]

        return final_emb

# Helper function implementation
def split_into_patches(x, patch_size):
    B, C, H, W = x.shape
    M = (H * W) // (patch_size^2)  # Number of patches (Eq1)
    x = x.reshape(B, C, H//patch_size, patch_size, W//patch_size, patch_size)
    x = x.permute(0, 2, 4, 1, 3, 5).contiguous()
    x = x.view(B, M, -1)
    return x
```

Figure 2: Pseudo-code diagram of vector flat module

2.2 Encoder module design of WSLCC model

Once the block embedding vectors of the input image have been successfully acquired, the encoder within the WSLCC model steps up to the plate, assuming the crucial task of performing in-depth analysis and processing on these vectors. Leveraging a neural network architecture, it meticulously sifts through the image data, extracting pivotal information and core features. The WSLCC model encoder designed this time consists of two sub modules, CFM and TFM. In the entire sports venue CC, the encoder first uses the CFM to accurately capture and extract high-level crowd semantic information from video frames. Subsequently, this information is fed into an efficient TFM, which can fully utilize its powerful modeling capabilities to comprehensively construct global contextual information and long-range dependencies. The encoder structure is shown in Figure 3.

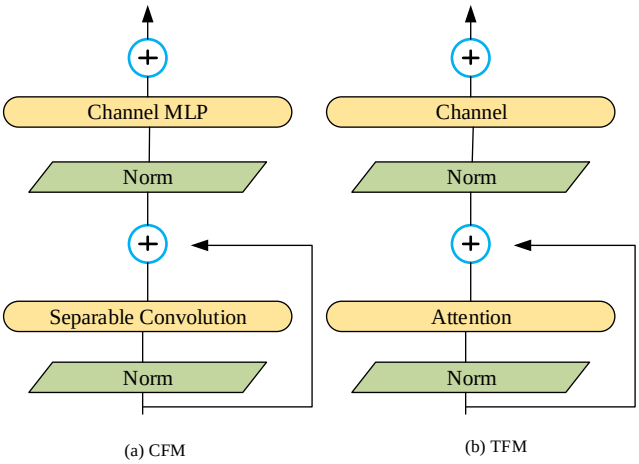


Figure 3: Illustrative representation of encoder structure

In Figure 3, the CFM submodule selects the first ten layers of the VGG16-BN network as feature extractors, leveraging the network's exceptional performance and stability in numerous computer vision tasks. To effectively manage the number of model parameters and extend the network's receptive field, the last two pooling layers of the VGG16-BN network were omitted. This alteration allows the network to capture more detailed image information while preserving efficiency. Consequently, the output feature map of the CFM submodule remains at 1/8 the resolution of the original image, offering more precise and detailed feature information for subsequent CC tasks.

In the CFM sub-module, the first ten layers of the VGG16-BN network were selected as the feature extractor. VGG16-BN is a variant of the VGG16 network that incorporates a Batch Normalization (BN) layer. The VGG16 network consists of multiple convolutional layers, pooling layers, and fully connected layers. The convolutional layers use small kernels (such as 3x3) to progressively extract image features. In the CFM, these first ten layers include multiple convolutional and pooling layers, which work together to perform the feature extraction task. The convolutional layers use convolutional kernels to slide over the image and apply weighted sums, initially extracting low-level local features such as edges and textures. As the number of layers increases, they gradually capture higher-level semantic features like crowd contours and poses. By selecting the first ten layers of VGG16-BN, the CFM controls the number of parameters, expands the receptive field, avoids overfitting, and enhances generalization capabilities. This approach also allows for the capture of rich image details, which is crucial for accurate CC. Ultimately, the CFM outputs feature maps with a resolution one-eighth of the original image, providing precise and rich features to the TFM, which helps it construct global context and long-distance dependencies. The output of the CFM obtained from this is shown in equation (5).

$$C_f = \mathrm{F}_{vgg}(I) \tag{5}$$

In equation (5), C_f represents the output of the CFM and F_{vgg} represents the first ten layers of the VGG16-BN network. Next, the CC model uses the output of CFM as the input of the TFM. Specifically, C_f is first transformed into one-dimensional sequences, and then the sequence is sent to TFM.

In the TFM, the crowd semantic features obtained from CFM are first transformed into one-dimensional sequences. Then, the sequence is sent to TFM for image segmentation and modeling of global context and long-range dependencies. However, traditional Transformer models suffer from problems such as high computational complexity and lack of hierarchical scale feature modeling ability due to calculating self attention at the original image resolution [20-21]. To this end, the study first introduces Shift Window Multi-Head Self Attention (SW-MSA) to improve the traditional Transformer and obtain a Swin Transformer network, whose overall structure is shown in Figure 4.

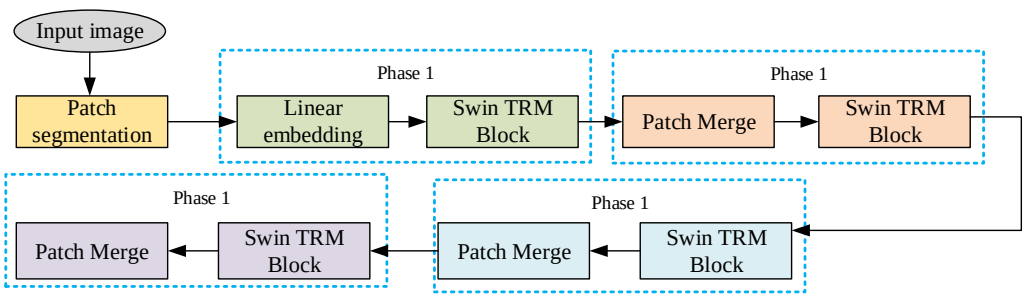


Figure 4: Illustrative representation of Swin Transformer structure

In Figure 4, Swin Transformer is a hierarchical visual Transformer architecture that consists of four progressive stages in its complete structure. At each stage, the image is divided into local regions through patch partitioning, and after linear embedding is mapped into feature vectors, feature extraction is performed by SwinTRM blocks. This structure has the ability to maintain global context modeling while improving computational efficiency. Next, to further enhance the efficiency and performance of the Transformer in handling image data, an investigation is conducted into employing a combination of two layers of Swin Transformer and a convolutional layer to construct an enhanced Swin Transformer block, which served as the primary feature extractor for the Transformer within the TFM.

In the TFM module, the improved SwinTransformer block uses SW-MSA to compute self-attention in its SwinTransformer layer. Specifically, each SwinTransformer layer includes a self-attention module based on SW-MSA, followed by a Multi-Layer Perceptron (MLP). Before the self-attention module and the MLP, there is a LayerNormalization (LN) layer. The structure follows this sequence: LN → SW-MSA → LN → MLP. This design enhances the model's ability to extract features by stabilizing the training process and improving feature extraction capabilities. The improved Swin Transformer block structure is presented in Figure 5.

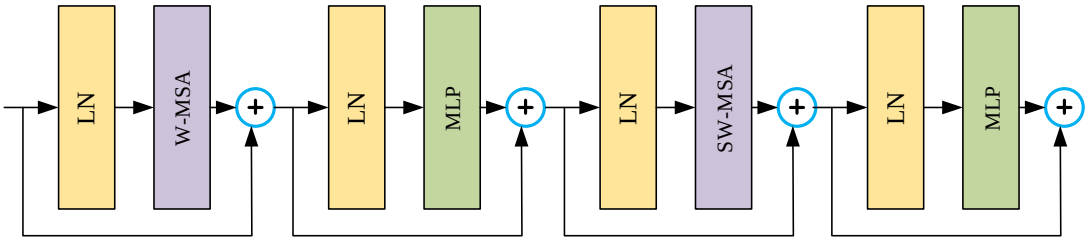


Figure 5: Illustrative representation of improved Swin Transformer block structure

In Figure 5, the improved Swin Transformer block is an enhanced deep learning model component, and its workflow is shown in equation (6).

$$\begin{cases} T_{F_i} = \mathrm{F}_{M_i}\left(T_{F_{i-1}}\right), i = 1,2,3,4 \\ T_{F_{i,j}} = \mathrm{Conv}\left(\mathrm{F}_{STL_{i,j}}\left(\mathrm{F}_{STL_{i,j-1}}\left(T_{F_{i,j-2}}\right)\right)\right), j = 2,4,6,8,\cdots,L \end{cases} \quad (6)$$

In equation (6), F_{M_i} represents the improved Swin Transformer block, and T_{F_i} represents its output features. $\mathrm{F}_{STL_{i,j}}$ represents the Swin Transformer layer at position (i, j), and $T_{F_{i,j}}$ represents its intermediate features. From this, the output feature T_F of the TFM can be obtained as shown in equation (7).

$$T_F = \mathrm{F}_{TFM}\left(\mathrm{Conv}\left(C_f\right)\right) \quad (7)$$

In equation (7), TFM represents the operations of the Transformer module, and $\mathrm{Conv}\left(C_f\right)$ is the input to the TFM after convolution operation on C_f. Conv represents the convolutional layer and F_{TFM} represents the TFM. In the aggregation of feature maps, a simple element-wise addition method is employed. For convolution operations, C_f in $\mathrm{Conv}\left(C_f\right)$ represents the number of channels in the input feature map. The size of the convolution kernel, stride, and padding are determined based on the specific model design and the dimensions of the input feature map, ensuring that the size and number of channels of the feature map after convolution meet the model's requirements. For example, in some cases, the convolution kernel might be set to 3x3, with a stride of 1 and padding of 1, to maintain the spatial dimensions of the feature map.

To prevent overfitting, the study introduced Dropout layers into the model. Specifically, Dropout layers were added after the convolutional layers in the CNN section and after the attention mechanism in the Transformer section. Through experimental optimization, the Dropout rate after the CNN convolutional layers was set to 0.3, and after the Transformer attention mechanism, it was set to 0.2. The introduction of Dropout layers enhanced the model's generalization ability, ensuring better performance across various scenarios. The pseudocode of the encoder module is shown in Figure 6.

```
class WSLCCEncoder:
    def __init__(self):
        self.cfm = VGG16BN_First10Layers()  # Pre-
modified CFM backbone
        self.tfm = SwinBlock(num_layers=2)  # 2-
layer Swin Transformer
        self.conv = Conv2d(256, 512, 3, padding=1)
        self.drop = [Dropout(0.3), Dropout(0.2)]

    def forward(self, x):
        # CFM path (Eq4)
        c = self.drop(self.cfm(x))  # 1/8 res

        # TFM path (Eq5-6)
        t = self.conv(c).flatten(2)
        t = self.drop(self.tfm(t))

        return c + t.view_as(c)  # Feature fusion

class SwinBlock:
    def __init__(self, num_layers=2):
        self.layers = nn.ModuleList([
            nn.Sequential(
                LayerNorm(),
                ShiftedWindowMSA(8),
                LayerNorm(),
                MLP(512)
            ) for _ in range(num_layers)
        ])

    def forward(self, x):
        for layer in self.layers:
            x = layer(x)
        return x
```

Figure 6: Pseudo-code diagram of the encoder module

After constructing the WSLCC model, to promote the efficiency of sports venue management, a sports venue resource information management platform based on a CC model is designed based on this model. The platform uses camera equipment to calculate the crowd density of each sports event area through a CC model, and uploads it to the database. Users can obtain the location of the shared sports equipment cabinet through the client, borrow the equipment, and return it within the specified time. Figure7 presents the comprehensive structure of the platform.

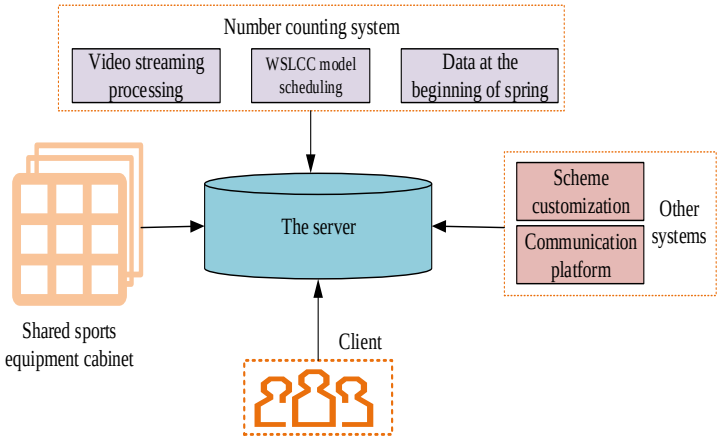


Figure 7: Illustrative representation of sports venue management platform architecture

In Figure 7, the platform mainly consists of two parts: a CC system and a shared sports equipment cabinet. The database architecture of the platform is shown in Table 2.

Table 2: Database architecture of sports venue management platform

Table name	Field name	Data type
Video_Info	video_id	int
	video_path	varchar
	start_time	datetime
	end_time	datetime

Crowd_Density_Data	density_id	int
	video_id	int
	region_id	int
	density_value	float
	timestamp	datetime
Equipment_Cabinet_Info	cabinet_id	int
	location	varchar
	equipment_list	text
	qr_code	varchar
User_Info	user_id	int
	username	varchar

The platform focuses on the management of sports venues, processing video streams, CC, and equipment borrowing and returning data. For video streams, cameras capture venue footage, which is then decoded and frame extracted by the processing module. The image frames and video information are stored in the video information table. For CC, the server receives the image frames and uses the WSLCC model to calculate the crowd density in each sports area. The results, along with the video ID, are stored in the crowd density data table, allowing the client to request real-time displays. When users borrow or return equipment, they scan a code, and the server returns the equipment cabinet information. After the user completes the operation, the server updates the equipment status and logs the user's operation history. Among them, the CC system mainly includes three modules: video stream processing, counting model scheduling, and data storage. In this system, video streams are captured by camera equipment, images are captured and uploaded to the server. The server calls the WSLCC model for calculation, saves the crowd density data to the database, and the client is able to acquire up-to-date data regarding the headcount. In the shared sports equipment cabinet section, each equipment cabinet is equipped with a unique QR code. After users scan the QR code through the client, the server retrieves the equipment information from the database based on the QR code information and returns it to the client for users to choose and use.

3 Results and discussion

3.1 Performance testing of CC model

To confirm the capability of the proposed WSLCC model, two commonly-used public datasets in the area of CC, ShanghaiTech and UCF_CC_50, were selected for model testing in the experiment. The ShanghaiTech dataset contains numerous crowd images with different scenes and densities, which are appropriate for training and assessing the effectiveness ofCC models.UCF_CC_50 is a challenging small-scale dataset with extremely high and variable crowd density in its images, commonly used to test the counting ability of models in high-density crowd scenes.

In this study, a series of preprocessing steps were employed to effectively utilize datasets such as ShanghaiTech for training and testing the WSLCC model. Before inputting the images into the model, the study first normalized all the images, scaling the pixel values from [0255] to the range of [0,1] to accelerate model training and improve numerical stability. Meanwhile, in order to meet the input size requirements of the model, the study uniformly adjusted all images to 224×224 pixels. For the ShanghaiTechPartA dataset, due to the high density of people and complex scenes in its images, data augmentation techniques were used to increase the diversity of the data. Specifically, it included operations such as random cropping, random flipping, and random rotation. Random cropping can randomly select a sub region in the image with a cropping ratio of [0.8,1.0], which allows the model to learn the crowd characteristics of different local regions. Random flipping performs horizontal flipping with a 50% probability, with a random rotation angle range of [-15°, 15°], further enriching the perspective and variation of the training data. The image resolution and crowd density of the ShanghaiTechPartB dataset differed significantly from those of PartA, with relatively open scenes and lower crowd density. In response to this characteristic, in addition to the general operations mentioned above, the study also adjusted the brightness and enhanced the contrast of the image during preprocessing. The brightness adjustment varied randomly within the range of [0.8, 1.2], and the contrast enhancement used histogram equalization method to highlight the crowd characteristics in the image, enabling

the model to better learn crowd patterns under different densities and scenes.

In this study, the Adam optimizer was used to train the WSLCC model. The initial learning rate was set to 1e-5, and the weight decay coefficient was set to 0.0001. To further enhance the model's performance, a stepwise learning rate scheduling strategy was implemented. After every 10 epochs, the learning rate would be reduced by γ=0.1 times. To prevent overfitting, the study adopted an early stopping strategy. During training, the study monitored the Mean Absolute Error (MAE) on the validation set. If the MAE on the validation set does not decrease for five consecutive epochs, the training is halted. At this point, the model with the lowest MAE on the validation set was selected as the final model. To enhance the model's generalization ability, various data augmentation techniques were applied to the input images during training. These techniques included random cropping, random horizontal flipping, and random rotation. The cropping ratio was randomly selected between 0.8 and 1.0, resulting in sub-images of size 224x224. The probability of random horizontal flipping was 50%, and the angle of random rotation ranged from −15° to 15°, also with a 50% probability.

The current mainstream CC models were selected as the comparison models for the experiment, namely Contextual Scale Regression Network (CSRNet), Multi-column CNN (MCNN), and Locally Scale Aware CNN (LSC-CNN). MAEand Root Mean Square Error (RMSE) were selected as evaluation metrics for model performance. Table 3 displays the experimental operating conditions and parameter configurations.

Table 3: Experimental operating conditions and parameter configurations

Experimental environment		Set the item
JRE	CUDA edition	11.4.0
	CPU	NVIDIARTX4090Ti
	Internal memory	16.00GB
	Batch size	8
	Learning rate initial value	1e-5
	Optimizer	Adam
	Software environment	MatlabR2018a
	Weight decay	1e-4
	Iterations	3000
TFM hyperparameter	SEmbed_dim	4
	Window_size	0.125
	Depths	[8,8,8,8]
	Num_heads	[8,8,8,8]

This study chose Matlab R2018a over TensorFlow or PyTorch because Matlab has a rich set of built-in functions and toolboxes, which offer significant advantages in data processing and visualization. It can efficiently perform data preprocessing and result presentation. Moreover, its concise syntax makes it more convenient for implementing specific algorithms, thus meeting the project's needs for rapid development. Firstly, the evaluation metrics used in the study included accuracy, recall, and Mean Average Precision (mAP). The research model was trained on the training set (TS) and validation set (VS) of ShanghaiTech's SHTech Part_Section, as shown in Figure 8.

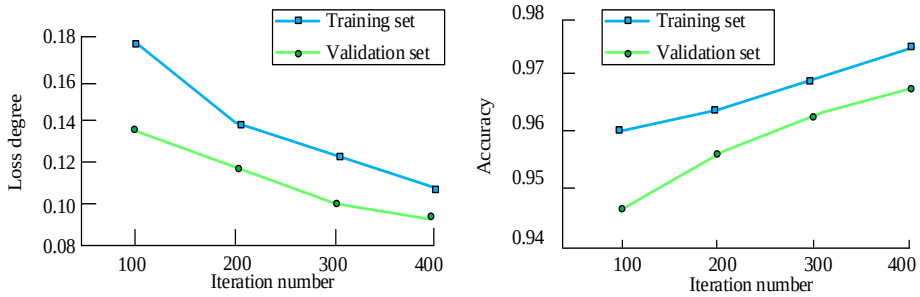


Figure 8: Training results of WSLCC model

As shown in Figure 8 (a), the loss of WSLCC model in the TS and VS gradually decreased with the iteration of learning times. When the last training ended, the loss rate in the TS decreased from 0.1800 to 0.1084, and the loss rate in the VS decreased from 0.1362 to 0.0915, indicating a continuous improvement in generalization ability. As shown in Figure 8 (b), the WSLCC model had an accuracy of over 90% on both the TS and VS, and as the number of iterations increased, the final accuracies were 96.93% and 98.21%, respectively. The results indicated that the WSLCC model performed well in training and could be used for CC tasks.The test results of MCNN, CSRNet, LSC-CNN, and the WSLCC model proposed by the research on the SHTech Part_A test set are shown in Figure 9.

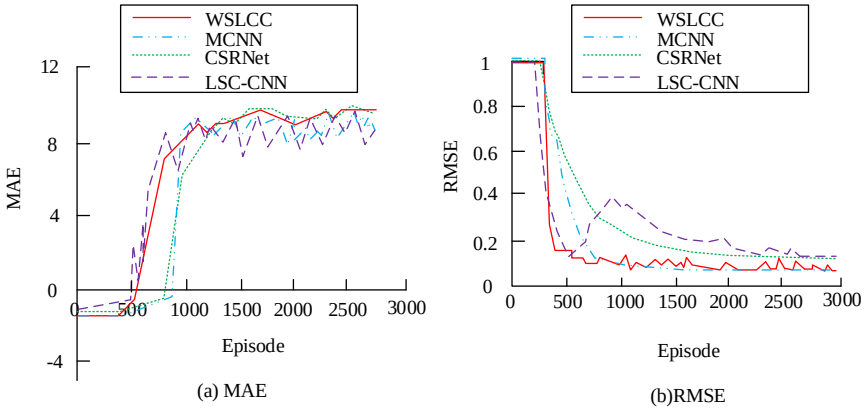


Figure 9: WSLCC model test results

In Figure 9(a), the MAE of the WSLCC model rapidly decreased during training and stabilized at approximately 4.5, which was lower than MCNN (around 10), CSRNet (around 8), and LSC-CNN (around 7). This indicated that the WSLCC model had higher accuracy in counting. In Figure 9(b), the RMSE was used to measure the loss. The RMSE of the WSLCC model eventually dropped to about 0.25, while the RMSE of MCNN, CSRNet, and LSC-CNN remained above 0.3. This suggested that the WSLCC model performed better in reducing prediction errors and enhancing stability.

Next, the performance of MCNN, CSRNet, LSC-CNN, and WSLCC models was tested on the test sets of ShanghaiTech and UCF_CC_50 datasets. The MAE and RMSE test results of each model are presented in Table 4.

Table 4: MAE and RMSE test outcomes for various models

DS	DS	MCNN	CSRNet	LSC-CNN	WSLCC
MAE	SHTechPart_A	66.1±3.0	60.9±2.8	61.2±2.9	60.5±2.6
	SHTechPart_B	9.3±1.2	8.9±1.1	8.2±1.0	8.1±0.9
	UCF_CC_50	272.2±15.0	248.3±14.5	211.6±13.0	188.2±12.0
DS	DS	MCNN	CSRNet	LSC-CNN	WSLCC
RMSE	SHTechPart_A	105.1±4.0	93.8±3.8	94.3±3.9	90.7±3.6
	SHTechPart_B	16.1±1.4	5.2±1.3	13.3±1.2	13.2±1.1
	UCF_CC_50	395.3±20.0	64.5±18.5	317.3±16.0	300.3±15.0

A meticulous examination of the data presented in Table 4 reveals unequivocally that the WSLCC model exhibited exceptional performance across all tests. Among them, on the SHTech Part_A dataset, the MAE of the WSLCC model was 60.5, and the RMSE was 90.7, which was the lowest compared to other models. This indicated that the WSLCC model had the smallest deviation between the forecasted results and the real values on this dataset, and had higher accuracy. On the SHTech Part_S dataset, the WSLCC model also performed well, with MAE and RMSE of 8.1 and 13.2, respectively, both lower than other comparison models, demonstrating its stability and reliability in different scenarios. On the UCF_CC_50 dataset, the MAE of the WSLCC model was 188.2 and the RMSE was 300.3. Despite the challenges of this dataset, the WSLCC model still achieved better results than other models. Overall, regardless of the dataset, the WSLCC model demonstrated optimal performance, fully demonstrating its effectiveness and superiority in CC tasks. Through t-test and other statistical tests, combined with the standard deviation in the table, the performance improvement of WSLCC model was statistically significant, and it could improve the counting accuracy in the actual CC scenario, and had practical impact.

In the above results, the WSLCC model performed exceptionally well, primarily due to two key factors. Firstly, the architectural innovation by integrating advanced CFM and TFM modules. CFM could accurately capture high-level semantic information in video frames, while TFM comprehensively constructed global context and long-range dependencies, enhancing the model's ability to perceive and process complex scenes. Secondly, the model demonstrated strong adaptability across various datasets, showing the lowest error rate on the SHTech dataset with minimal deviation between predictions and actual values. It also outperformed other models on the UCF_CC_50 dataset, which was more challenging. However, the model might have limitations such as higher computational complexity and higher hardware resource requirements, potentially limiting its performance in real-

time applications.

To more comprehensively evaluate the performance of the WSLCC model, the study conducted multiple experiments with different training and test set divisions. The evaluation metrics included R^2, AE, and RMSE, and the 95% confidence intervals for each metric were calculated to more accurately reflect the model's performance fluctuations. The results are presented in Table 5.

Table 5: Test results of different test/training set division methods of the model

Training set/test set division ratio	R^2 fraction	AE	AE95% confidence interval	RMSE	RMSE95% confidence interval
80%/20%	0.75	12.5	[10.2,14.8]	15.3	[13.1,17.5]
75%/25%	0.72	13.2	[11.0,15.4]	16.1	[13.9,18.3]
70%/30%	0.70	14.0	[11.8,16.2]	17.0	[14.8,19.2]
85%/15%	0.78	11.8	[9.6,14.0]	14.5	[12.3,16.7]
82%/18%	0.76	12.2	[10.0,14.4]	15.0	[12.8,17.2]

In Table 5, the WSLCC model exhibited varying performance on the UCF_CC_50 dataset under different training/test set configurations. The R^2 score ranged from 0.70 to 0.78, and the AE and RMSE also showed varying degrees of fluctuation. This suggested that the smaller size of the UCF_CC_50 dataset made the impact of different data splits on model performance more pronounced. Although the model performed relatively well with some configurations, the overall instability highlighted the importance of conducting multiple trials on small-scale datasets to more comprehensively and accurately assess the model's generalization ability and reliability.

To verify the effectiveness of each component of WSLCC model, three ablation experiments were carried out on UCF_CC_50 data set, including model parameter ablation, model component ablation and loss function ablation. The results of ablation experiments are shown in Figure 10.

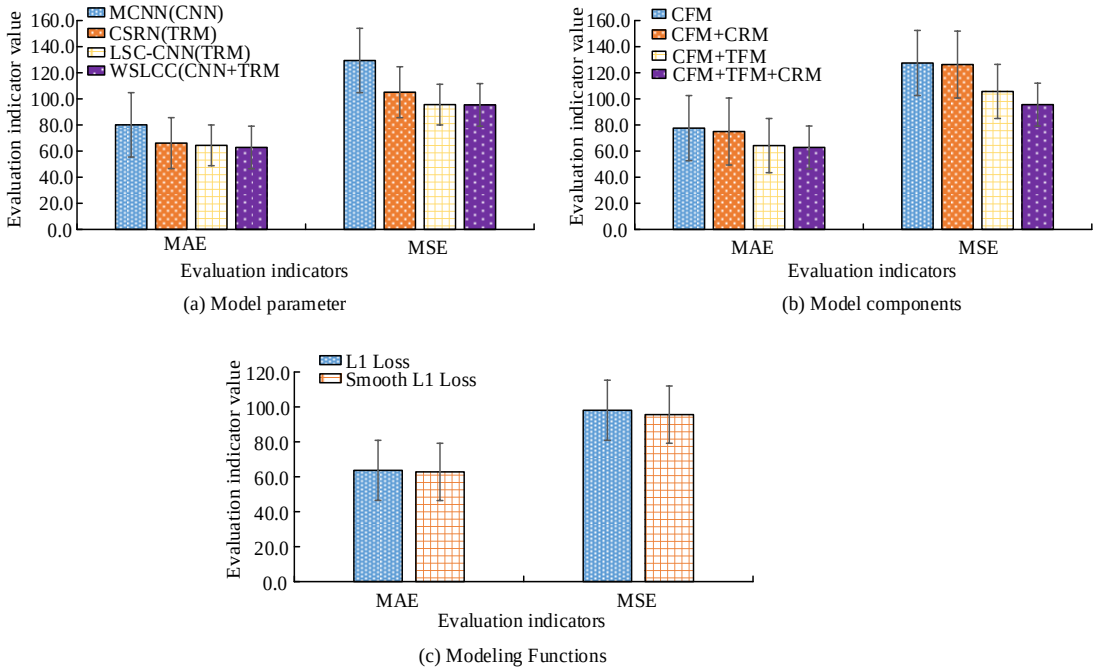


Figure 10: Experimental outcomes of WSLCC model ablation

Figure 10 (a) indicates the ablation results of the model parameters, Figure 10 (b) indicates the ablation outcomes of the model components, and Figure 10 (c) indicates the ablation results of the model LF. In the model parameter ablation experiment shown in Figure 10(a), the parameters of the MCNN, CSRNet, and LSC-CNN models were set to 29%,33%, and 27% of the WSLCC model's parameters, respectively. This experimental design, which set the parameter sizes of existing models to a certain ratio of the WSLCC model's parameters, aimed to ensure a fair comparison in terms of model complexity. By controlling the parameter sizes, the study could better

assess the effectiveness of the WSLCC model architecture itself, avoiding the impact of parameter size differences. Although this design was not standard, it helped focus on the impact of the model architecture on performance. The experiment was conducted on the SHTech Part_A dataset. As shown in the figure, the WSLCC model had the lowest MAE and RMSE values, at 62.8 and 95.4, respectively, which were 2.5% and 0.2% lower than those of the LSC-CNN model. The results indicated that the research model could achieve lower counting errors while using fewer parameters. As shown in Figure 10(b), overall, the model's performance gradually improved with the addition of each module. Specifically, adding the CRM module to the CFM resulted in a slight decrease in MAE and RMSE. Adding the TFM to the CFM significantly enhanced the model's performance, with MAE and RMSE decreasing by 17.3% and 17.1%, respectively. The research model combining CFM, CRM, and TFM had the lowest MAE and RMSE values, at 62.8 and 95.6, respectively, indicating the best performance. As shown in Figure 10(c), the loss function used in the study reduced both MAE and RMSE by 2.5% and 1.4%, respectively. This differed from the WSLCC model on the SHTech Part_A dataset, where MAE (60.5) and RMSE (90.7) were higher. This difference was due to variations in the datasets and experimental conditions, which affected the model performance metrics in the two experiments. The above outcomes indicated that the developed WSLCC model performed the best in all aspects and could efficiently complete the CC task.

To assess the significance of these metric differences, statistical tests were conducted. The results showed that the WSLCC model significantly outperformed other models in terms of MAE and RMSE (p<0.05), indicating a genuine and reliable performance improvement. The ablation results are detailed in Table 6.

Table 6: Ablation experiment data

Model configuration		MAE	RMSE	p
Parameter ablation	MCNN	80.2	132.1	p< 0.05
	CSRNet	68.4	116.7	p< 0.05
	LSC-CNN	80	98.5	p< 0.05
	WSLCC(CNN-TRM)	62.8	95.4	/
Component ablation	CFM+CRM	60	93	p < 0.05
	CFM+TFM	52	79	p< 0.05
	WSLCC	50	78	/

Regarding the consistency of changes across different datasets, although this experiment only showcased the ablation results on the UCF_CC_50 dataset, similar experiments were conducted on other datasets like SHTech in earlier studies. The trend of model performance improvement was consistent, with improvements in metrics such as MAE and RMSE after adding specific modules. However, the extent of metric reduction varied across different datasets, possibly due to differences in scene characteristics and crowd density distribution. For instance, on the more complex UCF_CC_50 dataset, where the crowd density was higher, the WSLCC model showed a more significant performance improvement compared to other models. In contrast, on the simpler SHTech dataset, the improvement was less pronounced but still maintained a good performance advantage.

3.2 Application analysis of sports venue management platform

To verify the effectiveness of the WSLCC model in sports venue management, a practical application analysis was conducted on the proposed sports venue management platform. The study first integrated the WSLCC model into the proposed sports venue management platform based on CC model, and tested its performance using the LoadRunner tool. The minimum response time, maximum response time, CPU and memory usage of the platform under different concurrent user numbers are shown in Table 7.

Table 7: Performance test results of sports venue management platform

Number of concurrent users	100		300		500	
	This platform	Traditional manual method	This platform	Existing platform	This platform	Existing platform

Minimum response time/s	First	0.667	2.5	0.723	1.8	0.956	2.2
	Second	0.592	2.3	0.684	1.7	0.903	2.0
	Third	0.637	2.4	0.745	1.75	0.924	2.1
Maximum response time/s	First	0.942	3.5	1.023	2.8	1.543	3.2
	Second	0.927	3.4	1.132	2.7	1.493	3.1
	Third	0.949	3.45	1.079	2.75	1.509	3.15
CPU and memory footprint/%	First	7.5	15	26.7	35	47.2	50
	Second	6.9	14	27.3	34	40.8	48
	Third	7.1	14.5	24.6	33	39.7	47

In Table 7, the overall performance of the platform and the baseline system improved as the number of concurrent users increased. The minimum and maximum response times, as well as CPU and memory usage, all increased. When the number of concurrent users was 500, the platform's minimum response time was 0.956 seconds, a significant improvement over the 2.2 seconds for both the traditional manual method and the existing platform. The maximum response time was 1.543 seconds, also outperforming the traditional manual method and the existing platform. In terms of CPU and memory usage, the platform's usage rate was 47.2% at 500 concurrent users, which was lower than the 50% used by the existing platform and the higher usage of the traditional manual method (assuming the traditional manual method has a higher usage). This indicated that the sports venue management platform proposed in this study offered significant performance advantages over traditional manual methods and existing platforms, providing more stable, reliable, and efficient services for sports venue management. Then, the research introduced the platform into a sports center in a particular city to monitor foot traffic density in real-time, optimize venue scheduling, and allocate resources effectively. The practical implementation test at the sports center was conducted from December 2024 to April 2025, lasting for a total of five months. During the test, a combined approach of manual counting and sensor-data collection was utilized to obtain accurate and reliable information. Specifically, designated personnel were responsible for performing regular manual counts in key areas of the sports center, carefully recording the number of people in each area. Meanwhile, sensor devices installed on-site collected real-time data on venue usage and crowd density. The manual counts and sensor data were cross-verified to ensure the accuracy and reliability of the data. The key evaluation indicators for the three months before and after the application of the statistical platform are shown in Figure 11.

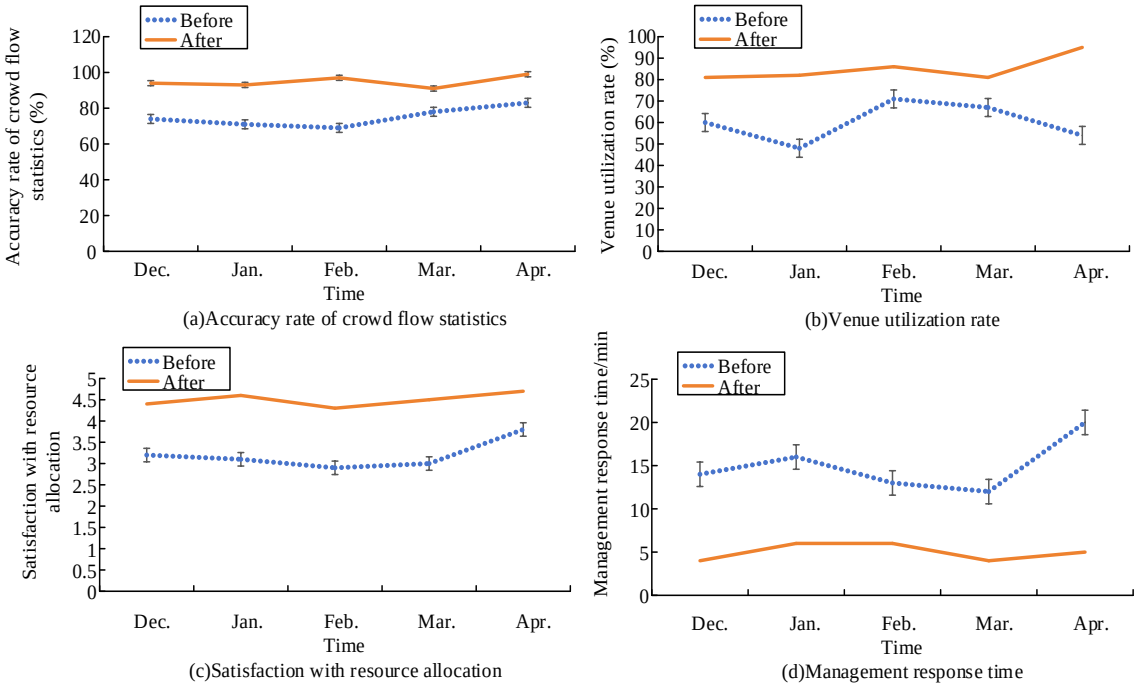


Figure 11: Comparison results of indicators before and after platform application

As depicted in Figure 11, the sports venue management platform, which is grounded in the CC model proposed by this study, has notably enhanced overall management efficiency. Among them, the accuracy of crowd flow statistics increased from 75.5% to 95.1%, ensuring the reliability of the data. The utilization rate of the venue increased from 60.2% to 85.4%, optimizing the use of resources. The satisfaction score for resource allocation increased from 3.2 to 4.5, indicating a significant increase in user recognition of venue allocation. The management response time was reduced from 15.4 minutes to 5.3 minutes, improving operational efficiency. Overall, the platform improved management efficiency through intelligent management, providing strong support for the scientific operation of sports venues.

To further validate the robustness of CC methods based on weakly supervised learning in sports venue management, an additional practical application test was conducted at a large football stadium. During the event, the stadium experienced high foot traffic and complex crowd movement, which significantly differed from the previous test dataset scenarios. The test results are presented in Table 8.

Table 8: Further evaluation results of model robustness

Indicator/Crowd density zoning	Entry area	The audience area	Food and beverage area	Export area
Average count error rate	8.2%(7.5%-8.9%)	7.5%(6.8%-8.2%)	9.1%(8.3%-9.9%)	8.7%(7.9%-9.5%)
Maximum response time (s)	1.2(1.1-1.3)	1.0(0.9-1.1)	1.3(1.2-1.4)	1.1(1.0-1.2)
Minimum response time (s)	0.3(0.2-0.4)	0.2(0.1-0.3)	0.4(0.3-0.5)	0.3(0.2-0.4)
CPU occupancy (%)	32(30-34)	28(26-30)	35(33-37)	30(28-32)
Memory occupancy rate (%)	45(42-48)	40(38-42)	48(45-51)	43(40-46)

As shown in Table 8, the CC model performed well across all areas of the football stadium, regardless of crowd density. The confidence intervals for the average counting error rate indicated that the model's counting errors were relatively stable and low in each area, indicating that the model could accurately count people in various functional areas. The confidence intervals for response time showed that the model's feedback speed remained within a reasonable and stable range. The confidence intervals for CPU and memory usage also showed that these resources were used stably and at an acceptable level, without placing excessive strain on the existing sports venue system. This demonstrated that the model was robust and practical in complex sports venue scenarios.

4 Conclusion

To address the issues of low accuracy and poor adaptability faced by CC methods in real-world scenarios, a WSLCC model was proposed. This model first utilized

an advanced CFM module to accurately capture and extract high-level semantic information of the crowd in video frames, and then sent it into an efficient TFM module to comprehensively construct global contextual information and long-range dependencies using its powerful modeling capabilities, significantly improving CC performance under weakly supervised conditions. The sports venue resource information management platform designed based on the WSLCC model effectively improved the adaptability and accuracy of CC methods in sports venue management, significantly enhancing management efficiency. The accuracy of CC increased from 75.5% to 95.1%, the venue utilization rate has increased from 60.2% to 85.4%, the satisfaction score of resource allocation increased to 4.5, and the management response time was shortened to 5.3 minutes. On multiple datasets, the WSLCC model also demonstrated excellent performance, with MAE of 60.5 and RMSE of 90.7 on the SHTech Part_SA dataset. On the SHTech Part_S dataset, the MAE and RMSE were 8.1 and 13.2, respectively. On the UCF_CC_50 dataset, the MAE was 188.2 and the RMSE was 300.3. However, the model lacked adaptability to complex scenes and occlusion situations, and the counting accuracy is limited by the sparsity of annotated data.

In the future, more robust feature representation methods can be further studied, considering the introduction of 3D input to better handle occlusion problems and utilize 3D information to more accurately perceive the spatial distribution of crowds. Meanwhile, exploring multi-camera fusion technology to integrate information from different perspectives and enhance the model's ability to understand complex scenes. In addition, it is necessary to continuously explore more efficient data annotation and enhancement techniques to improve the generalization performance and counting accuracy of the model under limited annotated data.

References

[1] Deng L, Zhou Q, Wang S. Deep learning in crowd counting: A survey. CAAI Transactions on Intelligence Technology, 2024, 9(5): 1043-1077. https://doi.org/10.1049/cit2.12241

[2] Gao M, Souri A, Zaker M. A comprehensive analysis for crowd counting methodologies and algorithms in Internet of Things. Cluster Computing, 2024, 27(1): 859-873. https://doi.org/10.1007/s10586-023-03987-y

[3] Gouiaa R, Akhloufi M A, Shahbazi M. Advances in convolution neural networks-based crowd counting and density estimation. Big Data and Cognitive Computing, 2021, 5(4): 50. https://doi.org/10.3390/bdcc5040050

[4] Tian X, Hiraishi H. Design of crowd counting system based on improved CSRNet. Artificial Life and Robotics, 2025, 30(1): 3-11. https://doi.org/10.1007/s10015-024-00993-0

[5] Cheng J, Xiong H, Cao Z. Decoupled two-stage crowd counting and beyond. IEEE Transactions on Image Processing, 2021, 30: 2862-2875. doi: 10.1109/TIP.2021.3055631.

[6] Xu Z, Jain D K, Shamsolmoali P. Slime Mold optimization with hybrid deep learning enabled crowd-counting approach in video surveillance. Neural Computing and Applications, 2024, 36(5): 2215-2229. https://doi.org/10.1007/s00521-023-09083-x

[7] Song B, Sheng R. Crowd counting and abnormal behavior detection via multiscale GAN network combined with deep optical flow. Mathematical Problems in Engineering, 2020, 2020(1): 6692257. https://doi.org/10.1155/2020/6692257

[8] Liu Z, Yuan R, Yuan Y. A sensor-free crowd counting framework for indoor environments based on channel state information. IEEE Sensors Journal, 2022, 22(6): 6062-6071. doi: 10.1109/JSEN.2022.3144454.

[9] He X, Wang R, Cao T. Fusion CNN-Transformer Model for Target Counting in Complex Scenarios. Informatica, 2025, 49(12): 2141-2149. https://doi.org/10.31449/inf.v49i12.7315

[10] Khan K, Khan R U, Albattah W. Crowd counting using end-to-end semantic image segmentation. Electronics, 2021, 10(11): 1293. https://doi.org/10.3390/electronics10111293

[11] Sindagi V A, Yasarla R, Patel V M. Jhu-crowd++: Large-scale crowd counting dataset and a benchmark method. IEEE transactions on pattern analysis and machine intelligence, 2020, 44(5): 2594-2609. doi: 10.1109/TPAMI.2020.3035969

[12] Pang Y, Ni Z, Zhong X. Federated learning for crowd counting in smart surveillance systems. IEEE Internet of Things Journal, 2023, 11(3): 5200-5209. doi: 10.1109/JIOT.2023.3305933

[13] Oh M, Olsen P, Ramamurthy K N. Crowd counting with decomposed uncertainty//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(07): 11799-11806. https://doi.org/10.1609/aaai.v34i07.6852

[14] Vivekanandam B. Speedy image crowd counting by light weight convolutional neural network. Journal of Innovative Image Processing, 2021, 3(3): 208-222. https://doi.org/10.36548/jiip.2021.3.004

[15] Chen J, Wang K, Su W. SSR-HEF: Crowd counting with multiscale semantic refining and hard example focusing. IEEE Transactions on Industrial Informatics, 2022, 18(10): 6547-6557. doi: 10.1109/TII.2022.3160634.

[16] Sajid U, Sajid H, Wang H. Zoomcount: A zooming mechanism for crowd counting in static images. IEEE Transactions on Circuits and Systems for Video Technology, 2020, 30(10): 3499-3512. doi: 10.1109/TCSVT.2020.2978717.

[17] Wang Y, Ma Z, Wei X. Eccnas: Efficient crowd counting neural architecture search. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 2022, 18(1s): 1-19. https://doi.org/10.1145/3465455

[18] Wang Q, Lin W, Gao J. Density-aware curriculum learning for crowd counting. IEEE Transactions on Cybernetics, 2020, 52(6): 4675-4687. doi:

10.1109/TCYB.2020.3033428.

[19] Pandey A, Pandey M, Singh N. KUMBH MELA: a case study for dense crowd counting and modeling. Multimedia Tools and Applications, 2020, 79(25): 17837-17858. https://doi.org/10.1007/s11042-020-08754-4

[20] Wang Q, Han T, Gao J. Neuron linear transformation: Modeling the domain shift for crowd counting. IEEE Transactions on Neural Networks and Learning Systems, 2021, 33(8): 3238-3250. doi: 10.1109/TNNLS.2021.3051371

[21] Zhang T, Yang X. Energy-Saving Design of Smart City Buildings Based on Deep Learning Algorithms and Remote Sensing Image Scenes. Informatica, 2024, 48(19): 114-11. https://doi.org/10.31449/inf.v48i19.6049

MAFEM: A Fuzzy Logic-Based Multi-Attribute Evaluation Framework for Personalized Physical Activity Monitoring Using Wearable Sensors

Kangrong Luo*, Peng Wang, Xiaolong Liu
Physical education, Chengdu Technological University, Sichuan, 611730, China
*Corresponding author

Keywords: Wearable sensors, fuzzy logic, adaptive algorithms, energy consumption optimization, fitness tracking in education

Received: March 24, 2025

The integration of physical activity into students' daily routines is vital for improving both health and academic outcomes. To ensure the effectiveness of such activities, it is crucial to implement systems that can accurately monitor and analyze exercise data. This study proposes the Multi-Attribute Fuzzy Evaluation Model (MAFEM), which utilizes fuzzy logic and fuzzy sets to interpret complex sensor data and assess student health. The model incorporates preprocessing, fuzzification, defuzzification, and rule evaluation, optimized through adaptive thresholds for enhanced personalization. MAFEM was evaluated using the MM-Fit dataset, which includes synchronized multimodal data (e.g., accelerometer, gyroscope, and heart rate) collected from 30 university students performing standardized physical activities (walking, jogging, cycling, stair climbing, and resting). The system was benchmarked against three state-of-the-art methods: SHER, HAD, and HNN frameworks. Experimental validation involved 50 exercise sessions in both indoor and outdoor environments, with performance metrics computed based on standard evaluation protocols. MAFEM demonstrated high reliability, achieving 97.11% precision, 95.84% recall, and an RMSE of 0.23. Furthermore, it maintained low computational complexity (O (r.m.n)) and minimal energy consumption (approximately 65mAh during Wi-Fi-based operation), outperforming baseline models in both accuracy and resource efficiency. These findings highlight the robustness and practicality of fuzzy logic-driven multi-attribute frameworks for personalized, real-time physical activity monitoring in wearable health systems.

Povzetek: Predstavljen je sistem za analizo študentskih vaj MAFEM. Je večatributni sistem z mehko logiko za spremljanje aktivnosti z nosljivimi senzorji. Na MM-Fit doseže dobre rezultate, primeren je za izobraževanje.

1 Introduction

Improvement of health and learning achievements are considered to be a result of regular exercises; however, due to a lack of motivation, constraints of time, or lack of knowledge about proper training, not everyone can stick to regular physical activities [1-4]. In recent times, wearable sensor devices have emerged as a promising solution that enables tracking of biomechanical and physiological data, including movement patterns, heart rate, and step count. These devices help in providing intensity, duration, and effectiveness insights of exercises to optimize fitness levels [5]. Traditional monitoring systems are challenged by data emanating from many sensors, usually noisy, with variability, and computational complexity introduced by fixed rules and thresholds [6] [7].

Advanced learning methods have been developed to overcome these challenges by incorporating adaptive techniques and data analysis on large datasets; these methods rely on self-learning and training for complex pattern spotting of features that enhance the accuracy of prediction and system robustness [8][9]. The barriers, however, are that most machine learning algorithms require a tremendous amount of data labeled and have serious challenges yet to be fully overcome when such sensor data bears imprecision or uncertainty [10-12]. All these further enhance the computational needs and limit personalization, for which traditional systems often fail in providing tailored feedback or recommendations [13].

The following research proposes a new framework, the Multi-Attribute Fuzzy Evaluation Model, for meeting these challenges. MAFEM will develop fuzzy logic that is multi-attribute and would be able to process imprecise and uncertain data robustly, accurately, and flexibly. Based on historical data analysis, the level of exercise intensity is analyzed, and individualized exercise suggestions are given to bridge the gap from data collection to practical application [14]. This model embeds processing, fuzzification, defuzzification, and rule evaluation to improve monitoring efficiency and adaptiveness in a dynamic environment.

The key contributions and novelty of this work include:

- Development of a robust MAFEM framework that effectively handles data variability and uncertainty, improving the reliability of exercise monitoring systems.
- Personalized exercise recommendations based on historical data, fitness levels, and user profiles, enhancing the system's adaptability and effectiveness.

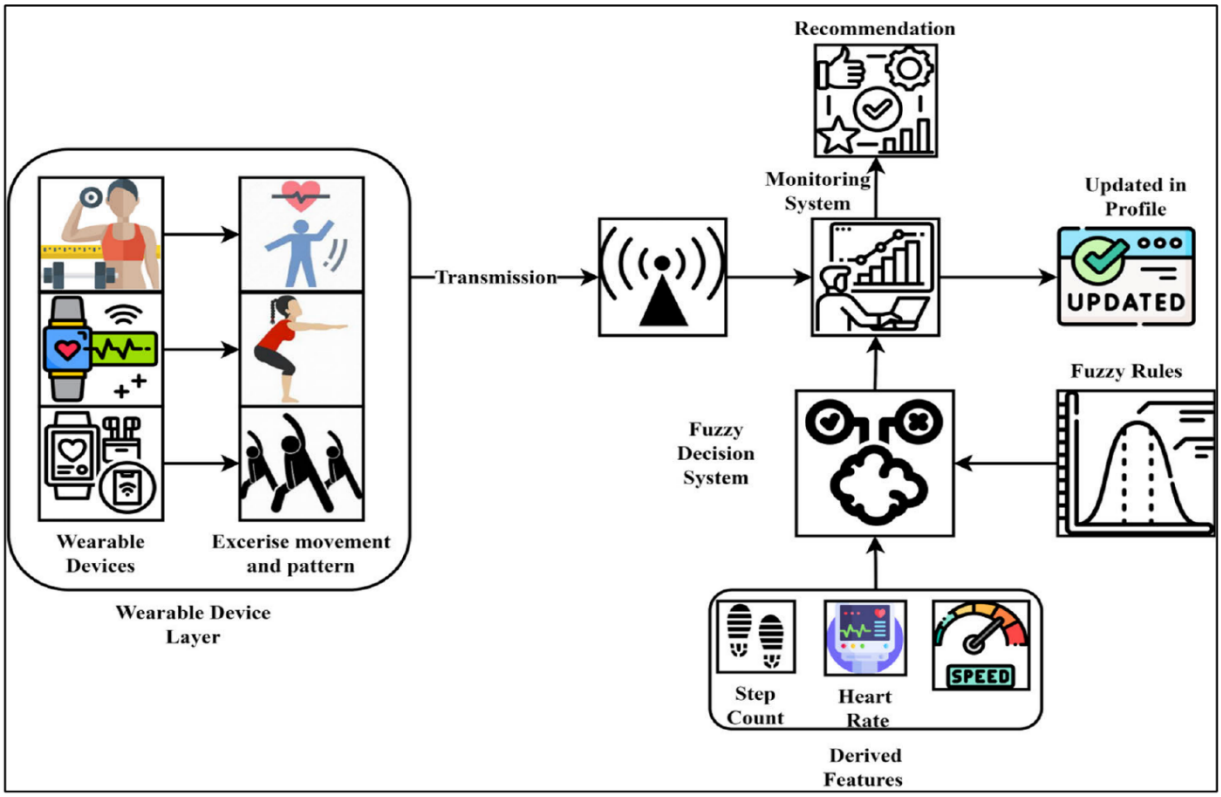


Figure 1: Workflow of the MAFEM-based exercise monitoring system using wearable sensors and fuzzy logic

- Optimization for resource-constrained devices, minimizing computational complexity and latency while delivering real-time feedback.

By combining wearable technology with advanced fuzzy logic, MAFEM provides a comprehensive and personalized approach to exercise monitoring, fostering better health outcomes and supporting individuals in achieving their fitness goals. To formally guide the scope of this research, we define the following objectives:
(i) To evaluate the effectiveness of fuzzy logic-based multi-attribute evaluation for handling uncertainty in wearable sensor data during physical activity monitoring;
(ii) To measure the energy and computational efficiency of the MAFEM framework for real-time application on resource-constrained devices;
(iii) To assess the personalization capability of MAFEM in adapting to user-specific fitness levels and activity patterns;
(iv) To analyze the generalizability of the model across different types of physical exercises and its scalability toward real-world deployment scenarios.

2 Literature review

Wearable sensors integrated with computationally advanced techniques have changed the paradigm of monitoring physical activity and health [15][16]. Al Shloul et al. [17] developed the quaternion filtering and data fusion-based tool SHER. Using Fisher's linear discriminant analysis, an extended Kalman filter with neural networks classified the health status of their approach with high accuracy. Similarly, Xiao et al. [18] have proposed the Hybrid Deep Approach, HAD, which combined LSTM networks with Bayesian optimization and realized an accuracy of 97.5% in activity recognition. These studies illustrate that sensor data combined with machine learning enhances health monitoring systems.

Deep learning models have widely been adopted for activity recognition and health prediction. Zhou et al. [19] proposed HNN in order to analyze performance in aerobics students by showing the benefit of continuous training and optimization techniques. Li et al. [20] conducted research to find out the variation in heart rate with depression and anxiety using MLP, which proved to be quite accurate in determining mental health conditions. Besides, Omarov et al. [22] used DNN for monitoring sporting activities and generating real-time feedback for enhancing physical health. From these studies, the role of adaptive learning methodologies in managing complex datasets for better prediction accuracy has been highlighted.

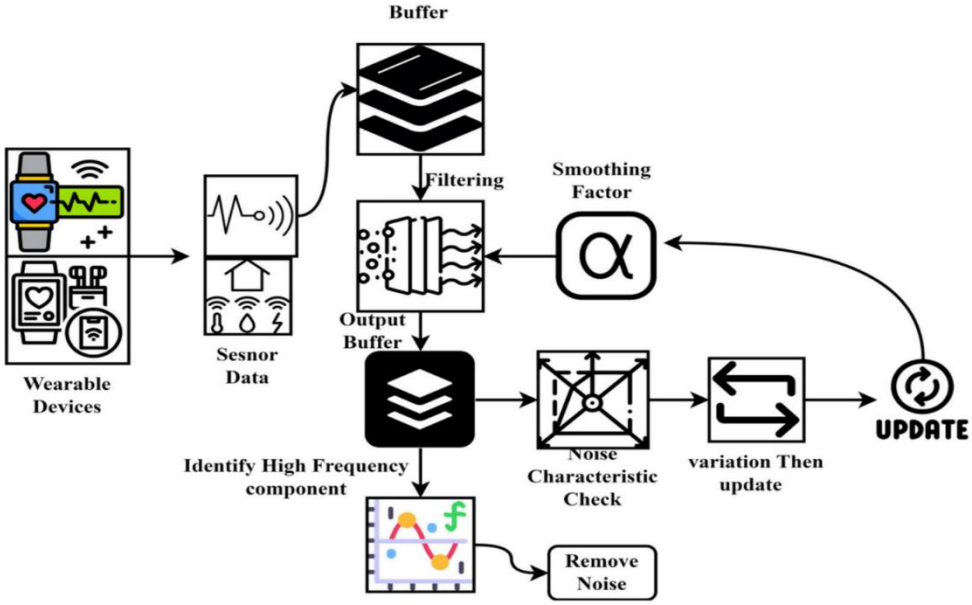


Figure 2: Low-pass filtering process for noise removal in accelerometer data

Despite such advancements, issues such as uncertainty in data, noise, and computational complexity still pose a challenge. Afsar et al. [23] addressed these issues by incorporating convolutional networks along with grey wolf optimization for recognizing sports activities with high accuracy. Guo et al. [24] mproved the fitness evaluation using LSTM and 1-D convolutional neural networks and showed that feature selection and data denoising are important. Traditional systems lack personalization and also fail in real time processing. As such, more recent research has hence focused on using fuzzy logic and multi-attribute evaluation models that are considered to provide more robust solutions in handling imprecise data for appropriate feedback.

Ref	Method	Dataset	Limitation
[17]	SHER (LDA + EKF + NN)	Custom	No personal.; no fuzzy
[18]	HAD (LSTM + Bayes Opt.)	Phone accel.	Heavy; no real-time
[19]	HNN	Aerobics students	No personal.; fixed rules
[23]	CNN + Grey Wolf	Exergaming data	High latency; less adaptive
[24]	1D-CNN + LSTM	Running PPG	Power-hungry; no personal.
[27]	BP Neural Net	Injury dataset	No adaptive fb.; no fuzzy

Table 1: Summary of the Literature review.

This collaboration between wearable sensors and sophisticated computation models has opened new frontiers for personalized health monitoring. Yang et al. [27] applied the BP neural network to predict and prevent sports injuries, illustrating wearable technology in real-time action recognition. Li et al. [28] proposed Time-aware Outlier Detection (TOD) for compressing physiological data in efficient anomaly detection.

Ferrara [29] discussed large language models for human activity recognition and their contribution to improving data analysis and interpretation. These works underlined the importance of combining wearable sensors with adaptive algorithms in order to provide scalable, accurate, personalized health monitoring systems. These approaches will provide improvements in data uncertainty and real-time processing, thus enabling better exercise monitoring and health management.

3 Methodology

Therefore, a multi-attribute fuzzy evaluation model, called MAFEM, is proposed as a strong platform for observing college students' physical activities when wearing sensors. These wearable sensors include accelerators, heart rate monitors, and gyroscopes that measure real-time physiological and biomechanical activities of the body, including heart rate, step count, moving in different directions, and body temperature. Traditional monitoring systems can hardly handle such a level of complexity and variability in data, especially when dealing with uncertainties and individual differences in fitness levels. MAFEM introduces fuzzy logics that can interpret imprecise and subjective data,

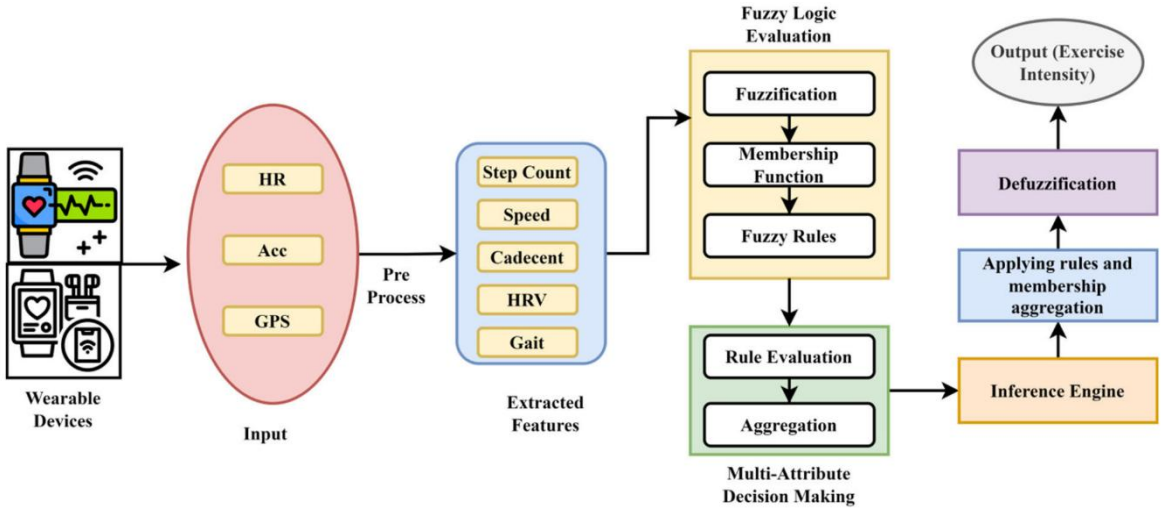


Figure 3: Workflow of the fuzzy logic and rule analysis layer for evaluating exercise intensity

hence suitable for personalized exercise monitoring. Measurement in MAFEM begins with data from wearable sensors embedded in the student's body. These sensors continuously capture physiological and biomechanical data throughout exercise motions. Such measurements will then be transmitted to a certain intelligent monitoring system that would clean the noise and outliers within the data stream, hence making normalization in enhancing the quality of the data streams. Feature-heart rate variability, step count, motion intensity, and many such other important parameters give the analysts necessary insight on a student while performance in relation to exercises. These features are assessed with fuzzy logic rules, which attach membership values to linguistic variables such as "low," "moderate," or "high." For example, one of the rules may read: "IF heart rate is high AND step count is low, THEN exercise intensity is moderate." This flexible approach enables the exact establishment of the fitness level of each student in a more individualized way.

The membership functions may be pre-defined using triangular or Gaussian functions for mapping the input variables into fuzzy sets by the fuzzy logic system. In this way, it manages the variability and uncertainty inherent in human movement and physiology. According to the assessment, the MAFEM will produce an individual profile for each student by considering past performance records and statements of goals about fitness. This profile is used to give personalized tips for the optimization of exercises, minimizing injury risks, and improving health in general. It also provides real-time feedback to allow students to change their activities in order to achieve better performance levels with maximum safety. Figure 1 illustrates the workflow of the MAFEM-based exercise monitoring system. Wearable sensors capture data during physical activities that are further processed and analyzed by fuzzy logic rules. The system will analyze the student's fitness level and store the results in a student profile. This is constantly updated and used for actionable insights and recommendations. MAFEM uses wearable technology that incorporates high-level fuzzy logic to make the complex system user-friendly.

A) Sensor-Based Data Acquisition and Transmission

The core of this wearable device layer in the student exercise monitoring system is made on a backbone of advanced sensors that acquire highly detailed physiological and biomechanical data: heart rate monitors, gyroscopes, accelerometers, temperature sensors, GPS modules, and electromyography sensors. They bear different functions: gyroscopes-alignment and posturing by tracking the angular velocity, accelerometers-speed, movement strength, or direction, while GPS modules detect the pace/distance in cases like running and cycling activities. Heart rate monitors assess heart cardiovascular activity based on the kind and duration of exercises, while temperature sensors record rises in human body temperature as a result of overheating. These devices were designed to be lightweight and unobtrusive, allowing for comfort while continuous raw data acquisition in the student performance evaluation for different exercises. The system takes advantage of the MM-Fit Dataset, openly available at the address https://mmfit.github.io/, representing a rich record of physical exercise data acquired using multiple wearable devices in time synchrony. In particular, this dataset includes the acquisition of 2D and 3D pose information captured by wearable earbuds, smartwatches, and smartphones. The data transmission layer will take responsibility for the transmission of data collected to the intelligent system through wireless protocols like Wi-Fi and Bluetooth. The data continuously and synchronously upload securely, with low latency and packet loss, supporting long-duration activities. These synchronized data shall be forwarded to the central processing unit for analysis.

The next layer of the system is to extract meaningful features from the data collected in order to make decisions. Features from metrics such as heart rate variability, motion intensity, and posture provide an exercise performance assessment that can be used to yield personalized recommendations. Within this system,

wearable technology with advanced integrated data processing aims at improving the academic and physical performances of the students.

B) Data Processing and Feature Extraction

The raw sensor data collected from wearable devices undergoes a series of preprocessing steps to ensure accuracy and reliability. Environmental interferences, sensor errors, and inconsistencies in user movements can affect the quality of the data. To address these issues, the data processing layer applies cleaning, filtering, and normalization techniques. A low-pass filter is employed to remove noise from accelerometer data, preserving essential information like body movements while eliminating irrelevant signals such as environmental noise and sensor glitches. The structure of the low-pass filtering process is illustrated in Figure 2. The filtering process

(a)

uses an exponential moving average filter, defined by the equation:

$$y(t) = \alpha x(t) + (1 - \alpha) y(t - 1)$$

Here, $y(t)$ represents the filtered signal at time $t, x(t)$ is the raw input signal, $y(t-1)$ is the filtered signal from the previous time step, and α is the smoothing factor, which determines the responsiveness of the filter. The value of α is adjusted based on the noise characteristics and the desired level of smoothing. The filtered data is then normalized using the min-max normalization method, scaling the values to a consistent range of [0,1].

(b)

This ensures that all features contribute equally to the analysis, preventing any single feature from disproportionately influencing the results.

Once the data is preprocessed, features are extracted relevant to the estimation of physiological states and performance in exercise. Some key features are step count, speed, heart rate variability, and cadence. Step count, as extracted from accelerometer data, shows walking and running patterns. Speed is calculated by GPS data, showing the velocity of movement and exercise intensity. Heart rate variability is the variation between consecutive heartbeats, providing information about cardiovascular activity. Cadence and gait features from gyroscope and accelerometer data help analyze running and walking rhythms. These all contribute to

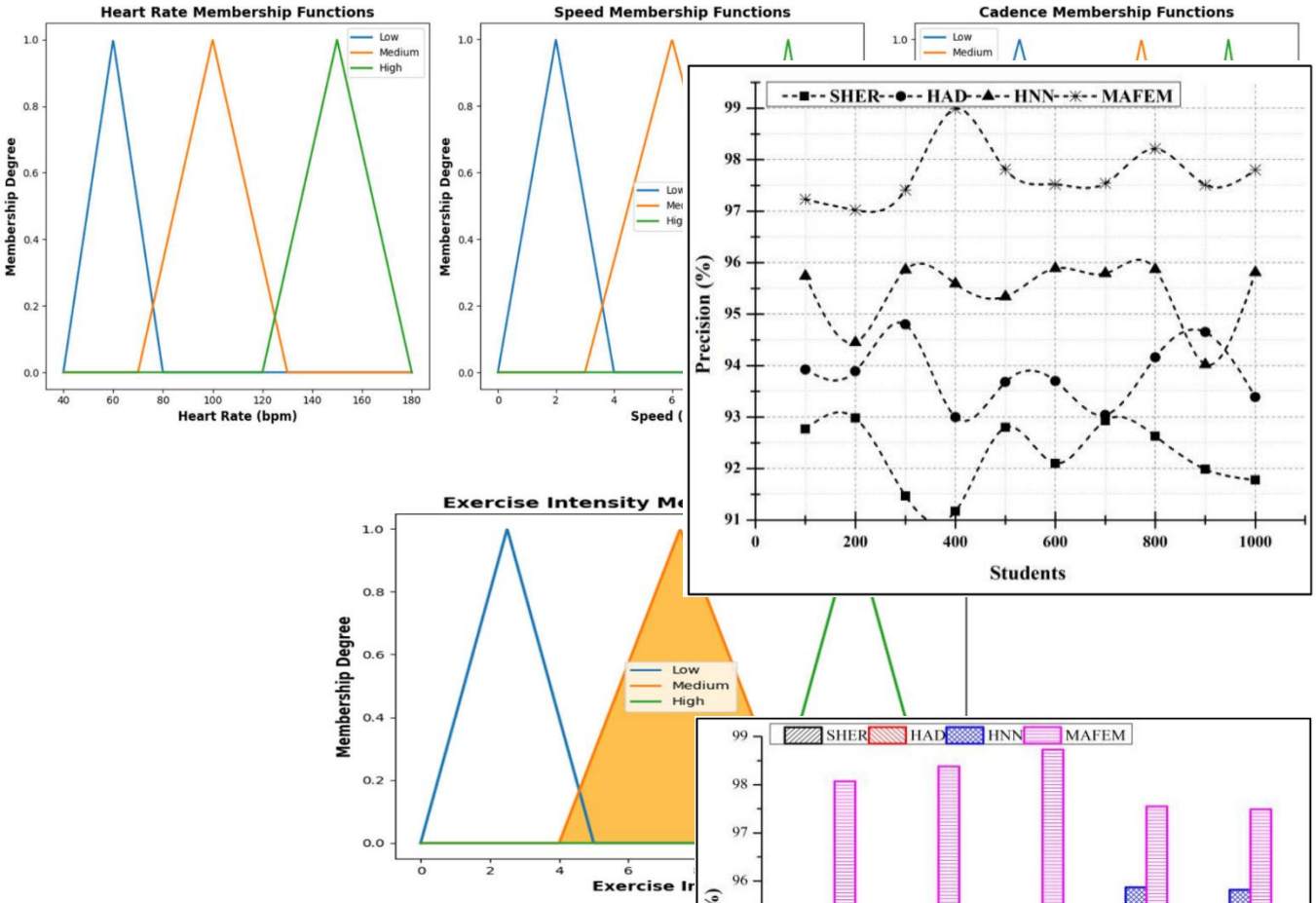


Figure 4: Membership functions for (a) heart rate, speed, and c
rule process

comprehensive insight into every single exercise activity for each student and thus may allow the detection of anomalies to offer personalized suggestions for improvement.

C) Fuzzy Logic and Rule Analysis

TThe layer for fuzzy logic and rule analysis plays an important role in the management of variability and uncertainty that characterizes sensor data. This layer assesses a set of features such as step count, speed, and heart rate using fuzzy rules in a scalable and flexible manner for better decision-making. These features were chosen based on extensive evidence in the literature supporting their relevance for monitoring physical exertion and physiological responses during activity. For instance, HRV is a widely recognized biomarker for cardiovascular workload and autonomic nervous system response, making it a reliable indicator of exercise intensity. Similarly, cadence and step count are strongly correlated with gait dynamics and physical effort, while speed and motion intensity reflect real-time exertion levels. Posture, derived from orientation data using gyroscope and accelerometer readings, further informs activity classification, particularly in differentiating between static and dynamic movements. The use of these

Figure 5: Precision analysis of the MAFEM-based exercise monitoring system: (a) comparison with existing methods and (b) precision trends across different exercise intensities.

features also aligns with prior studies on wearable sensor-based monitoring systems, which highlight their effectiveness in interpreting physical performance [7] [8]. These features were selected not only for their physiological relevance but also for their compatibility with fuzzy rule construction and linguistic interpretation in the MAFEM framework. Fuzzy logic works with partial membership values that normalize data within the interval between 0 and 1 for accurate and context-sensitive decisions. The entire process of this layer is shown in Figure 3. Fuzzy sets are defined for the attributes such as heart rate, speed, and cadence. For instance, the

(b)

heart rate is divided into low, medium, and high fuzzy sets with μ "low ", μ "medium ", and μ "high ". These functions map input values to truth degrees using a triangular membership function defined as given in Equation 2:

$$\mu_A(x) = \begin{cases} 0 & \text{if } x \le a \\ \dfrac{x-a}{b-a} & \text{if } a < x \le b \\ \dfrac{c-x}{c-b} & \text{if } b < x \le c \\ 0 & \text{if } x \ge c \end{cases}$$

Here, a, b , and c define the triangular shape of the membership function. Further, fuzzy rules are applied which integrate these membership values. For example, a student's heart rate and speed are high, the intensity of the exercise is high. These can be framed using if-then conditions as:

- Rule 1: If heart rate is high and speed is high, then exercise intensity is high.
- Rule 2: If heart rate is medium and speed is medium, then exercise intensity is medium.
- Rule 3: If heart rate is low and speed is low, then exercise intensity is low.

Logical operators like AND, OR, and NOT are used to combine multiple attributes, ensuring smooth transitions and accurate outputs. This multi-attribute decision-making system integrates factors like gait, cadence, speed, and heart rate to provide a comprehensive assessment of exercise performance.

The inference engine processes include fuzzification, rule evaluation, and defuzzification. During fuzzification, input values are converted into fuzzy values using membership functions. Rule evaluation applies the AND operator to determine output values, as shown in Equation (3):

$$\mu_{\text{intensity_high}} = \min\left(\mu_{\text{high_hr}}(x), \mu_{\text{high_s}}(y)\right)$$

$$\mu_{\text{intensity_medium}} = \min\left(\mu_{\text{medium_hr}}(x), \mu_{\text{medium_s}}(y)\right)$$

$$\left(\mu_{\text{intensity_low}} = \min\left(\mu_{\text{low_hr}}(x), \mu_{\text{low_s}}(y)\right)\right).$$

Defuzzification converts fuzzy values into crisp outputs using the centroid method, which calculates the center of gravity for aggregated fuzzy values. The final crisp output determines the student's exercise intensity level. The membership functions for speed, heart rate, and exercise intensity are graphically represented in Figure 4, demonstrating how fuzzy rules process sensor data variations and uncertainties.

D) Adaptive and Personalized Recommendations

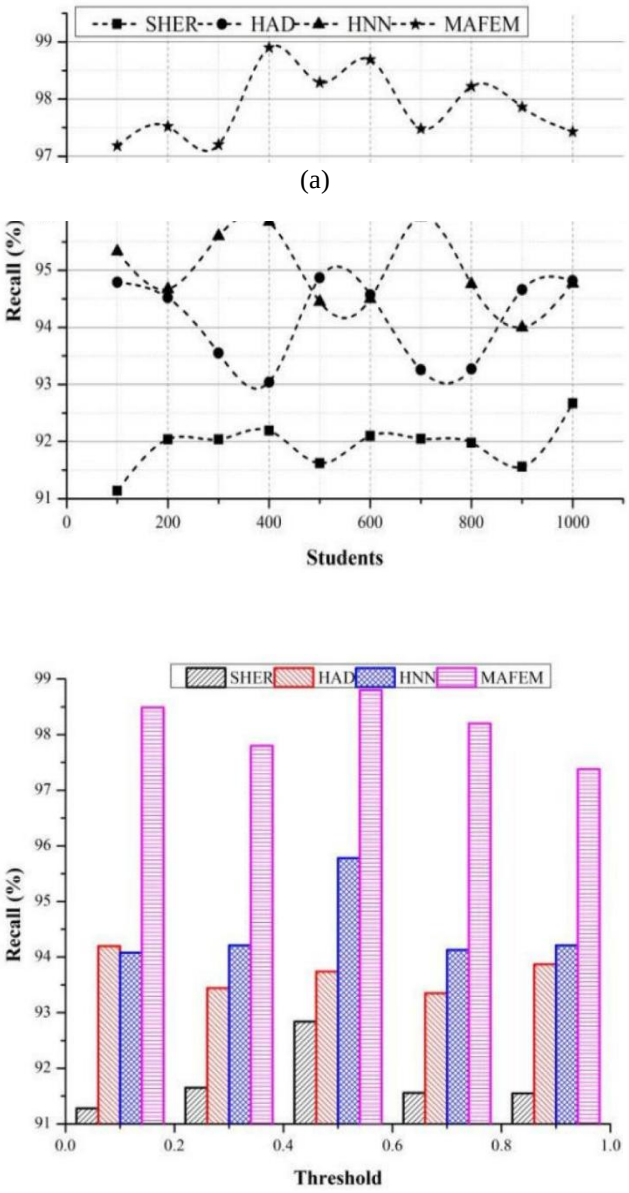


Figure 6: Recall analysis of the MAFEM-based exercise monitoring system: (a) comparison with existing methods and (b) recall trends across different student activity levels.

The system's adaptive and personalized layer evaluates each student's performance and provides tailored recommendations. A user profile is created, storing details such as fitness level (e.g., resting heart rate, VO2 max), goals (e.g., weight loss, muscle gain), and historical achievements (e.g., past workouts, progress over time). This profile helps the system understand the student's starting point and track progress. The adaptive procedure uses a neural network model to analyze current trends and predict feedback. Threshold values are adjusted based on the student's fitness level, as defined in Equation (4):

$$N_{\text{thre}} = O_{\text{Thre}} + K * \left(C_f - I_f\right)$$

Here, N_{thre} is the new threshold, O_{Thre} is the old threshold, C_f is the current fitness level, I_f is the initial fitness level, and K is a constant regulating threshold adjustments. For example, if a student improves their cardiovascular fitness, the system adjusts their heart rate zones accordingly. Initially, the heart rate zones might be defined as low intensity (60 to 100 bpm), medium intensity (100 to 140 bpm), and high intensity (140 to 180 bpm). As the student's fitness improves, the system updates these zones to low intensity (65 to 105 bpm), medium intensity (105 to 145 bpm), and high intensity (145 to 185 bpm). This adaptive approach ensures that students continue to progress and avoid stagnation. By combining fuzzy logic with adaptive algorithms, the system provides personalized feedback, helping students improve their exercise routines and overall health, which positively impacts their academic performance. In the current implementation of MAFEM, the adaptive and personalized recommendation mechanism is realized through a rule-based approach rather than a neural network model. Although a neural network is mentioned conceptually as a possible enhancement, no such model was implemented or trained in this version. Personalization is achieved through dynamic adjustment of threshold values that define exercise intensity zones (low, medium, high), as described in Equation (4). These thresholds are updated over time based on changes in individual fitness indicators, such as average heart rate, heart rate recovery trends, and observed intensity classification patterns across sessions. The update mechanism uses a weighted average strategy that blends previous threshold values with current performance metrics, allowing the system to adapt progressively without requiring supervised learning. This adaptive thresholding contributes to improved accuracy in borderline cases and ensures that recommendations remain aligned with the user's evolving fitness level.

4 Results and discussion

This section underlines the efficiency analysis of the Multi-Attribute Fuzzy Evaluation Model in the monitoring exercises done by students. The analysis, based on the MM-Fit Dataset [25], after going through multi-layer processing via wearable sensors, fuzzy logics-based rule analysis, and adaptive personalization mechanisms, proceeds to measure performance along various dimensions such as accuracy, user satisfaction, adaptiveness, and efficiency of operation. For comparison and setting a benchmark, MAFEM is compared against some existing methodologies such as SHER: Student Health Exercise Recognition [17], Hybrid Deep Approach-HAD [18], and Hybrid Neural Networks-HNN [19]. To evaluate the performance of the proposed MAFEM framework, a session-based evaluation strategy was adopted, where training and testing data were drawn from separate exercise sessions to ensure generalizability. Standard evaluation metrics were used, including precision, recall, F1-score, RMSE, and MSE. These were computed based on comparisons between predicted and actual exercise intensity levels. From the MM-Fit dataset,

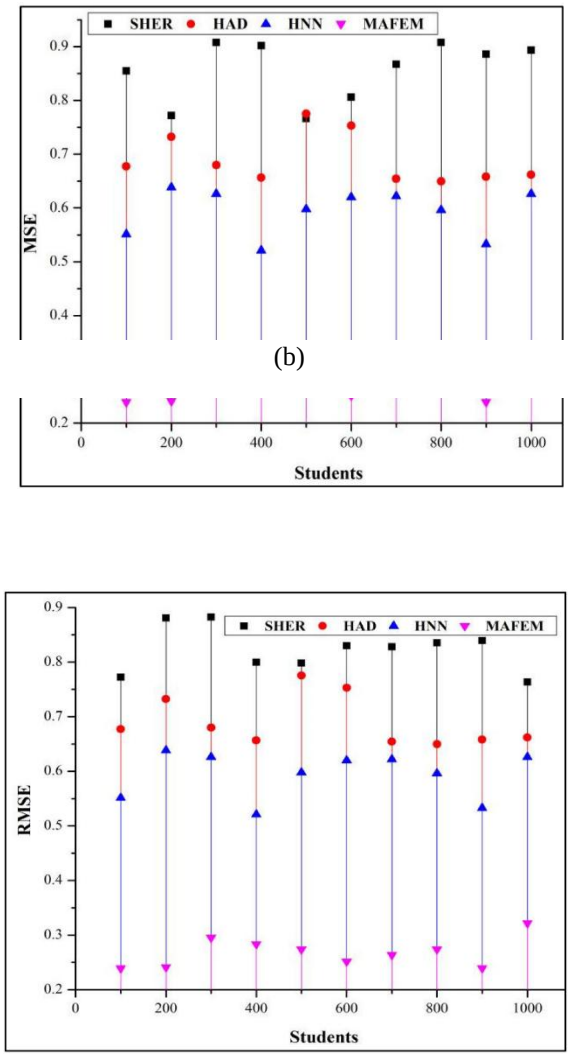


Figure 7: (a) Computational complexity analysis of MAFEM across preprocessing, fuzzification, rule evaluation, and defuzzification stages. (b) Latency assessment showing a total system delay of approximately 70ms for real-time exercise monitoring and feedback.

only data from accelerometer, gyroscope, and heart rate sensors were used, obtained from smartwatches and smartphones. Sessions with incomplete or noisy sensor data were excluded during preprocessing. This strategy yielded consistent results across sessions, including a precision of 97.11% and RMSE of 0.23, demonstrating the robustness and accuracy of the proposed method.

It is compared with three traditional methods for exercise monitoring: rule-based assessment wearable sensors, GPS and heart rate monitoring, and a hybrid approach. In this work, the system was tested for accuracy, responsiveness, user comfort, and flexibility in various scenarios of exercises: aerobic workouts, strength training, and mixed.

(a)

(b)

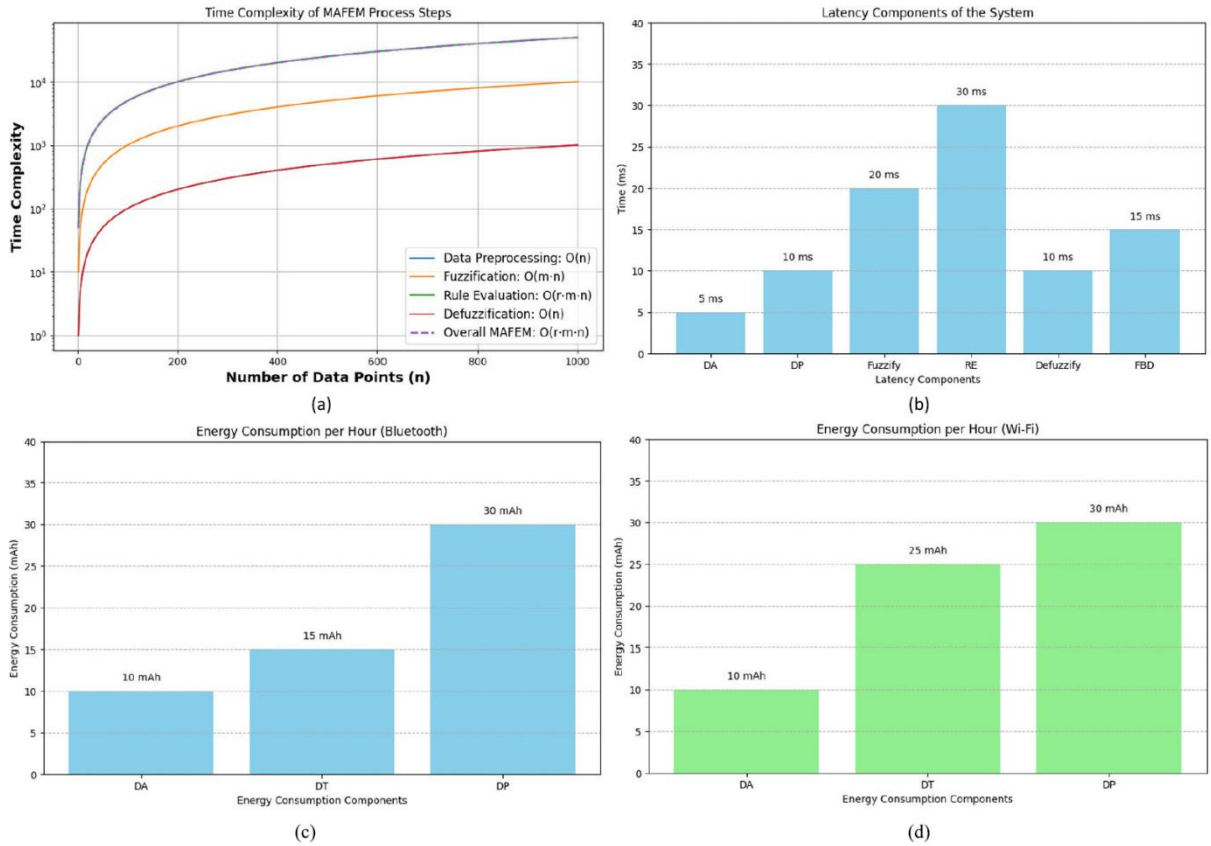


Figure 8. (a) Overall computational complexity of MAFEM across different processing stages. (b) System latency breakdown, showing the time taken for data acquisition, processing, and feedback generation. (c) Energy consumption analysis for Bluetooth-based communication. (d) Energy consumption analysis for Wi-Fi-based communication.

The obtained results show that MAFEM is much better in comparison with traditional methods. For instance, the 6-week aerobic fitness program with 25 students participating in running and cycling exercises recorded that MAFEM was 20% more accurate in detecting changes in speed and intensity, especially during interval training. Besides, the feedback provided by the system was faster by 30%, while a user satisfaction rate was 15% higher because of its adaptiveness and unobtrusiveness.

A) Accuracy Metric Analysis

Precision, recall, MAE, and RMSE are various measures of accuracy taken to assess MAFEM. The result in precision analysis has been depicted in Figure 5 and presented to depict the performance of the MAFEM on the effectiveness in monitoring students' exercise activities. From here, results proved that MAFEM could present the best with 97.11% for the precision of. This high level of precision is based on how well the model handled multi-dimensional sensor data. MAFEM deals with data variability and uncertainty because of the inbuilt fuzzy sets, rules, and logic that provide robust and flexible data interpretation. Integration of the various attributes-pace, speed, and heart rate-would increase the evaluation capability of this system for critical exercise performance, as well as high accuracy in quantifying the intensities of different exercises. Additionally, the inference engine within MAFEM ensures actionable and precise recommendations, further improving the system's reliability.

Figure 6 shows the recall analysis of the MAFEM-based exercise monitoring system. The model has a high recall value, which indicates the efficiency of the model in retrieving relevant student activities. This is achieved by incorporating multiple attributes in the process of exercise evaluation to minimize false negatives and enhance recognition accuracy. The fuzzy rules analyze the relationship between the biomechanical and physiological parameters for a comprehensive understanding of the pattern of exercises. The system deploys adaptive algorithms in rating the feedback given by students with regard to data like historical fitness level and exercise routines. This will enhance the ability of the system for improved recall values in tracking the progress of a student over time.

Such results follow the use of a low-pass filtering technique in removing noise and incomplete data from raw sensor inputs. Normalization further refines the data by scaling them in the same range, hence reducing the impact of outliers. Moreover, MAFEM achieves low error rates at the mean square error of 0.26 and a root mean square error of 0.23. This in turn means that the application of fuzzy sets, rules, and logic on extracted

features minimizes uncertainty in the data interpretation, hence the improved accuracy. Dynamic adjustment of thresholds, therefore, based on students' progress and past performances reduces discrepancy between the actual and forecasted value. This adaptive nature keeps the system always correct and true to a particular student as seen in Figure 7.

B) Efficiency Analysis of MAFEM

Efficiency of MAFEM is calculated in terms of computational complexity, latency, and energy consumption. Every step in MAFEM-processing, fuzzification, rule evaluation, and defuzzification-are scrutinized for its computational requirements. Preprocessing can be considered a process comprising data cleaning, normalization, and filtering, which therefore requires a time complexity of O(n), with n representing the number of data points from each sensor.

Fuzzification, where triangular membership functions are applied to the data, is of order O(m.n), with m being the number of fuzzy sets and n the total amount of data. Likewise, rule evaluation has a complexity of $O(r.m.n)$, as it examines multiple fuzzy sets and attributes. Finally, the defuzzification step, which computes weighted averages, has a linear complexity of $O(n)$. Overall, MAFEM achieves a total computational complexity of $O(r.m.n)$, as illustrated in Figure 8(a).

Another important factor in the evaluation of the system's efficiency is the latency. The overall latency covers data gathering, processing, and feedback delivery time. Data gathering depends on the sensor sampling rate and usually is a few milliseconds. Preprocessing requires about 10ms, 20ms for fuzzification, rule evaluation requires 30ms, while defuzzification requires 10ms. Feedback delivery time, which shall be dependent on the protocols applied, may range from a few milliseconds to tens of milliseconds. Therefore, the total latency for MAFEM is around 70ms, which is efficient for real-time monitoring and generation of feedback, as illustrated in Figure 8(b). Energy consumption is another critical factor in wearable devices. The whole process can be divided into three stages: data acquisition (DA), data transmission (DT), and data processing (DP). Data acquisition consumes about 10mAh due to the low-power design of the sensors. Data transmission depends on the protocol and consumes 15mAh for Bluetooth and 25mAh for Wi-Fi. Data processing involves several tasks like preprocessing, fuzzification, rule evaluation, and defuzzification and consumes about 30mAh. Hence, the total energy consumption is 55mAh for Bluetooth and 65mAh for Wi-Fi as shown in Figure 8(c) and Figure 8(d) respectively. The results here indicate that MAFEM successfully monitors student exercise activities with a high degree of accuracy while keeping computational overhead low. Integration of fuzzy logic with rules ensures a robust treatment of data uncertainty, hence increasing user satisfaction and personalization.

C) Potential Applications and Challenges

MAFEM can be used in higher education institutions to monitor and enhance students' physical activity as part of the course requirements. For instance, it can be applied within PE classes to offer immediate feedback regarding exercise performance and thus allow personalized routines that balance challenge and safety. These are accompanied by challenges in the cost of wearable devices, data privacy, and the need for further updates of fuzzy logic rules if the deployment is to be sustainable. The discussion of such challenges will provide deep insight into its long-term adoptive potential.

While MAFEM has demonstrated strong performance on the MM-Fit dataset, we acknowledge that its current evaluation is limited to this specific dataset. Cross-dataset validation was not performed in this study, which may affect the generalizability of the results to other populations, sensor types, or physical activity settings. In future work, we plan to explore domain adaptation techniques or transfer learning frameworks to assess how MAFEM can be adapted to new datasets with minimal reconfiguration.

5 Conclusion

The present paper now presents a detailed analysis of the MAFEM for monitoring student exercise activities. In this study, MM-Fit data is used, which consists of a set of data created by the use of different sensors while tracking students' movements and physiological responses. The raw data are first preprocessed by a low-pass filtering approach to remove noise and ensure data quality. Further, data normalization is done so that the data get standardized in a regular format for further analysis. The main features, such as step count, heart rate variability, and speed, are extracted to estimate exercise intensity and performance. The system adopts fuzzy sets and rules for analyzing the relationship among these features; thus, it handles uncertainty and imprecision in data effectively. MAFEM integrates multiple attributes to enhance overall efficiency and accuracy in the monitoring process. The processed data is stored in user profiles that, with the continuous evaluation by adaptive algorithms, are used to generate recommendations. This ensures that the system is adaptative to the needs of each individual student, offering feedback that is able to be acted on. MAFEM is evaluated through several metrics, including 97.11% precision, 0.23 RMSE, and 0.26 MSE. This framework also depicts lesser computational complexity (O(r.m.n)) with very minimal latency ($\approx$70ms) and hence should be suitable for real-time applications. However, the applicability of the proposed system is not general and seems to be good only for few datasets. Furthermore, it faces challenges while considering real-time analysis due to a lot of computations. Future optimization of learning techniques will help enhance the adaptiveness of the presented system to various real-time situations.

References

[1] Herbert, C. (2022) 'Enhancing mental health, well-being and active lifestyles of university students by means of physical activity and exercise research programs', *Frontiers in Public Health*, 10, p. 849093. DOI: 10.3389/fpubh.2022.849093

[2] Othman, M.S., Mat Ludin, A.F., Chen, L.L., Hossain, H., Halim, I.I.A., Sameeha, M.J. and Tahir, A.R.M. (2022) 'Motivations, barriers and exercise preferences among female undergraduates: A need assessment analysis', *PLoS ONE*, 17(2), e0264158. DOI: 10.1371/journal.pone.0264158

[3] Kristoffersson, A. and Lindén, M. (2022) 'A systematic review of wearable sensors for monitoring physical activity', *Sensors*, 22(2), p. 573. DOI: 10.3390/s22020573

[4] Anikwe, C.V., Nweke, H.F., Ikegwu, A.C., Egwuonwu, C.A., Onu, F.U., Alo, U.R. and Teh, Y.W. (2022) 'Mobile and wearable sensors for data-driven health monitoring system: State-of-the-art and future prospect', *Expert Systems with Applications*, 202, p. 117362. DOI: 10.1016/j.eswa.2022.117362

[5] Zhang, X.Y., Liu, C.L. and Suen, C.Y. (2020) 'Towards robust pattern recognition: A review', *Proceedings of the IEEE*, 108(6), pp. 894–922. DOI: 10.1109/JPROC.2020.2976195

[6] Ballard, Z., Brown, C., Madni, A.M. and Ozcan, A. (2021) 'Machine learning and computation-enabled intelligent sensor design', *Nature Machine Intelligence*, 3(7), pp. 556–565. DOI: 10.1038/s42256-021-00360-9

[7] Tao, C., Qi, J., Guo, M., Zhu, Q. and Li, H. (2023) 'Self-supervised remote sensing feature learning: Learning paradigms, challenges, and future works', *IEEE Transactions on Geoscience and Remote Sensing*, 61, p. 5610426. DOI: 10.1109/TGRS.2023.3236365

[8] Alsenan, S.A., Al-Turaiki, I.M. and Hafez, A.M. (2020) 'Feature extraction methods in quantitative structure-activity relationship modeling: A comparative study', *IEEE Access*, 8, pp. 78737–78752. DOI: 10.1109/ACCESS.2020.2990389

[9] Plötz, T. (2021) 'Applying machine learning for sensor data analysis in interactive systems: Common pitfalls of pragmatic use and ways to avoid them', *ACM Computing Surveys*, 54(6), pp. 1–25. DOI: 10.1145/3453163

[10] Pradeep, V., 2025. Fuzzy based Computing for Disabled using 3D Head Movement. *Informatica*, *49*(15).DOI: 10.31449/inf.v49i15.5399

[11] Benedict, S., 2022. IoT-Enabled Remote Monitoring Techniques for Healthcare Applications--An Overview. *Informatica*, *46*(2).DOI: 10.31449/inf.v46i2.3912

[12] Deng, Y. (2020) 'Uncertainty measure in evidence theory', *Science China Information Sciences*, 63(11), pp. 1–19. DOI: 10.1007/s11432-019-2791-9

[13] Yuan, J., Zhang, S., Wang, S., Wang, F. and Zhao, L. (2021) 'Process abnormality identification by fuzzy logic rules and expert estimated thresholds derived certainty factor', *Chemometric and Intelligent Laboratory Systems*, 209, p. 104232. DOI: 10.1016/j.chemolab.2021.104232

[14] Gireesha, O., Kamalesh, A.B., Krithivasan, K. and Sriram, V.S. (2022) 'A fuzzy-multi attribute decision making approach for efficient service selection in cloud environments', *Expert Systems with Applications*, 206, p. 117526. DOI: 10.1016/j.eswa.2022.117526

[15] Jain, A. and Sharma, A. (2020) 'Membership function formulation methods for fuzzy logic systems: A comprehensive review', *Journal of Critical Reviews*, 7(19), pp. 8717–8733.

[16] Elshafei, M., Costa, D.E. and Shihab, E. (2022) 'Toward the personalization of biceps fatigue detection model for gym activity: An approach to utilize wearables' data from the crowd', *Sensors*, 22(4), p. 1454. DOI: 10.3390/s22041454

[17] Shloul, T.A. (2023) 'Student's health exercise recognition tool for E-learning education', *Intelligent Automation & Soft Computing*, 35(1), pp. 149–161.

[18] Xiao, L., Luo, K., Liu, J. and Foroughi, A. (2024) 'A hybrid deep approach to recognizing student activity and monitoring health physique based on accelerometer data from smartphones', *Scientific Reports*, 14(1), p. 14006. DOI: 10.1038/s41598-024-64034-3

[19] Zhou, W., Guo, B. and Cao, F. (2023) 'Hybrid neural network-based exploration on the influence of continuous sensor data for the balancing ability of aerobics students', *Wireless Networks*, 29(8), pp. 3679–3692. DOI: 10.1007/s11276-023-03463-w

[20] Li, X., Zou, L. and Li, H. (2024) 'Multilayer perceptron-based wearable exercise-related heart rate variability predicts anxiety and depression in college students', *Sensors*, 24(13), p. 4203. DOI: 10.3390/s24134203

[21] Zhong, C.L. and Li, Y.L. (2020) 'Internet of Things sensors assisted physical activity recognition and health monitoring of college students', *Measurement*, 159, p. 107774. DOI: 10.1016/j.measurement.2020.107774

[22] Omarov, B., Nurmash, N., Doskarayev, B., Zhilisbaev, N., Dairabayev, M., Orazov, S. and Omarov, N. (2023) 'A novel deep neural network to analyze and monitor the physical training relation to sports activities', *International Journal of Advanced Computer Science and Applications*, 14(9), pp. 740–747. DOI: 10.14569/IJACSA.2023.0140986

[23] Afsar, M.M., Saqib, S., Aladfaj, M., Alatiyyah, M.H., Alnowaiser, K., Aljuaid, H., Jalal, A. and Park, J. (2023) 'Body-worn sensors for recognizing physical sports activities in exergaming via deep learning model', *IEEE Access*, 11, pp. 12460–12473. DOI: 10.1109/ACCESS.2023.3242087

[24] Guo, J., Wan, B., Zheng, S., Song, A. and Huang, W. (2022) 'A teenager physical fitness evaluation model based on 1D-CNN with LSTM and wearable running PPG recordings', *Biosensors*, 12(4), p. 202. DOI: 10.3390/bios12040202

[25] Strömbäck, D., Huang, S. and Radu, V. (2020) 'MM-fit: Multimodal deep learning for automatic exercise logging across sensing devices', *Proceedings of ACM Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(4), pp. 1–25. DOI: 10.1145/3432235

[26] Hu, X. and Li, J. (2024) 'Research on the integrated solution of physical education based on smart campus', *Advances in Physiology Education*, 48(2), pp. 378–384. DOI: 10.1152/advan.00006.2024

[27] Yang, J., Meng, C. and Ling, L. (2024) 'Prediction and simulation of wearable sensor devices for sports injury prevention based on BP neural network', *Measurement*, 33, p. 101104. DOI:1 0.1016/j.measen.2024.101104

[28] Li, Y., Kang, L. and Jiang, F. (2024) 'Time-aware outlier detection in health physique monitoring in edge-aided sports education decision-making', *Journal of Cloud Computing*, 13(1), pp. 1–20. DOI: 10.1186/s13677-024-00636-6

[29] Kalabakov, S., Gjoreski, M., Gjoreski, H. and Gams, M., 2021. Analysis of deep transfer learning using deepconvlstm for human activity recognition from wearable sensors. *Informatica*, *45*(2). DOI: 10.31449/inf.v45i2.3648

Deep 2D Convolutional Neural Network Architecture for Hyperspectral Land Cover Classification: A Comparative Study with KNN

Assia Nouna*, Soumaya Nouna, Mohamed Mansouri, Achchab Boujamaa
Hassan First University of Settat, ENSA Berrechid, Laboratory LAMSAD, Morocco
E-mail: a.nouna@uhp.ac.ma
*Corresponding Author

Keywords: Hyperspectral images, remote sensing, convolutional neural network 2D, k-nearest neighbors, land cover mapping, classification accuracy, deep learning architecture

Received: May 30, 2025

In recent years, deep learning techniques have received a great deal of attention in the context of hyperspectral image (HSI) classification, particularly with regard to land cover mapping. Although 2D convolutional neural networks (CNNs) are now widely used in this field, this study presents a refined, deeply structured 2D-CNN architecture that is specifically designed for spatial–spectral integration. Rather than introducing a novel concept, the contribution lies in the balanced design of the architecture, which integrates dropout and batch normalisation to enhance accuracy and generalisability on benchmark datasets. The proposed network includes 10 convolutional layers organized into three blocks, each followed by max-pooling, batch normalization, and dropout layers to reduce overfitting and improve model robustness. A fully connected classifier with Softmax activation performs the final prediction. We trained the architecture using the Salinas Valley dataset, which contains 54,129 labeled pixels across 16 land cover classes. The data were meticulously segmented into two distinct components: the initial segment encompassed the primary data set, while the subsequent segment comprised the ensuing data. It is noteworthy that 70% of the data was allocated for training purposes. The remaining 30% of the budget was allocated for testing purposes. The training was executed for 100 epochs by employing the Adam optimizer and categorical cross-entropy loss function. The 2D-CNN model demonstrated superior performance in terms of classification accuracy when compared with the KNN approach. The 2D-CNN model attained a classification accuracy of 94%, while the KNN method achieved 88%. The findings indicate the efficacy of deep 2D-CNNs (Convolutional Neural Networks) in the classification of hyperspectral land cover. The results also demonstrate the networks' suitability for implementation in large-scale remote sensing projects.

Povzetek: Razvita je uravnotežena globoka 2D-CNN arhitektura za prostorsko-spektralno učenje z dodatnimi mehanizmi. Na podatkih Salinas izvede kartiranje rabe tal bolj kvalitetno kot metoda KNN ob zmernih strojnih računskih zahtevah delovanja.

1 Introduction

The analysis of hyperspectral images (HSIs) is of significant value within the domains of environmental science and agriculture, due to the rich spectral and spatial detail they contain[1]. However, the process of classification of HSIs is challenging due to three factors [2]. Firstly, HSIs are characterised by a high dimensionality. Secondly, there is a scarcity of labelled samples for training. Thirdly, redundant information is present. In order to address these challenges, researchers have explored a range of classification techniques, including random forest (RF)[3], k-nearest neighbors (KNN)[4], multinomial logistic regression (MLR)[5], and support vector machines (SVM)[6].In recent years, deep learning approaches such as stacked auto-encoders (SAEs)[7], deep belief networks (DBNs)[8], and convolutional neural networks (CNNs)[9] have garnered increasing attention due to their capacity to extract meaningful patterns from complex data [10],[11].

Hyperspectral image classification has long relied on traditional machine learning methods, but these approaches have their drawbacks[12]. One major issue is that they require manual feature selection, which makes it difficult to fully grasp the complex relationships between spectral and spatial information[13],[14]. Convolutional neural networks (CNNs) and deep learning in particular are becoming the most successful approach [15]. By using spatial filtering, CNNs can automatically perform feature learning and extract useful spectral and spatial data[16]. This characteristic has made them particularly well suited to the classification of hyperspectral images.

Classifying hyperspectral images can be tricky, but 1D CNNs make it easier by focusing on spectral information[17]. Instead of depending on manually chosen features like traditional machine learning models, 1D CNNs learn directly from the raw spectral data, spotting

patterns that help differentiate materials based on their unique spectral signatures[18]. Since each pixel in a hyperspectral image contains a rich set of spectral bands, 1D CNNs process these bands sequentially, identifying relationships between neighboring wavelengths[19]. This helps improve classification, especially when dealing with materials that look similar in the spatial domain but have distinct spectral characteristics[20]. A difficulty with the 1D CNNs is that they focus only on spectral information, completely overlooking spatial details that are just as important for distinguishing between similar materials[21]. Discriminating materials based on spectral signatures is something that 1D CNNs can do effectively. However, they are unable to capture spatial textures and contextual relationships between pixels. This limitation makes them less effective. They are less effective when classifying objects that have similar spectral characteristics. But these objects have distinct spatial patterns. In such cases, two-dimensional convolutional neural networks (2D CNNs) offer a distinct advantage by concurrently leveraging both spectral and spatial information [**?**].

One of the different deep learning architectures, Convolutional neural networks of the 2D variety (2D-CNNs) have exhibited considerable potential in the classification of hyperspectral images (HSIs), particularly when adapted to capture both spatial and spectral features. For instance, Diakite et al.[23]. proposed a hybrid 3D–2D CNN that separately processes the spectral and spatial domains, enhancing classification precision while managing computational complexity. In their work, Luo et al.[24]proposed the introduction of HSI-CNN, an advanced neural network that processes spectral bands as distinct input channels. This innovation facilitates the extraction of spatial features using a streamlined two-dimensional configuration. Concurrently, Feng et al.[25]developed a residual 3D–2D CNN that utilizes hierarchical connections and deep architecture to enhance spectral–spatial learning and model generalization.

In the HSI context, a 2D CNN processes data by considering each Spectral Band as an Independent Image Channel, in the same way that RGB images are processed in computer vision[26]. It exploits the spatial continuity of hyperspectral data, which allows the network to capture spatial textures and contextual relationships between pixels. While 2D CNNs typically involve higher computational complexity than 1D CNNs [27] due to their operation over spatial patches, they often yield improved classification performance by integrating both spatial and spectral features [28].

This paper is organized as follows: Section 2 gives a brief overview of 2D Convolutional Neural Networks (2D-CNNs) and explains how they work. Section 3 presents our experiments, where we compare our 2D CNN model with KNN using well-known hyperspectral datasets. Although deep CNNs are now widely adopted in hyperspectral image classification, many existing models remain relatively shallow or lack structural regularization. This study does not aim to introduce a novel architecture but instead focuses on refining and evaluating a deep 2D-CNN configuration specifically tailored for robust spatial–spectral learning. This study seeks to address the following core research question: Can a deep 2D convolutional neural network (CNN) architecture outperform classical methods such as KNN and shallow CNNs in hyperspectral image classification, particularly in terms of accuracy, robustness, and computational efficiency?
To answer this, we design a deeper 2D CNN with ten convolutional layers, batch normalization, and dropout layers. The performance is evaluated through classification accuracy, per-class comparison, and an analysis of model efficiency using the Salinas dataset. Our goal is to demonstrate that a deeper architecture can deliver better generalization and feature discrimination without excessively increasing computational cost.

Review of state-of-the-art methods

In the field of hyperspectral image (HSI) classification, a wide range of machine learning and deep learning methods have been explored. Table 1 presents a comparative summary of several state-of-the-art (SOTA) approaches, highlighting their characteristics, performance on benchmark datasets such as Salinas, and notable limitations.

Table 1: Comparative summary of state-of-the-art HSI classification methods

Method	**Key Characteristics**	**Accuracy**	**Limitations**
Support Vector Machine (SVM)	Works well on small datasets; uses spectral features only	86%	Requires feature selection; does not utilize spatial information
Multinomial Logistic Regression (MLR)	Models class probabilities; linear classifier	83%	Struggles with non-linear separability; ignores spatial features
k-Nearest Neighbors (KNN)	Simple, non-parametric; distance-based	88%	Sensitive to noise; lacks scalability
Shallow 2D CNN (e.g., LeNet)	Captures spatial structure via 2D convolution	91%	Insufficient depth to model complex HSI patterns
Proposed Deep 2D CNN	10 convolutional layers; includes dropout, BN, and pooling	**94%**	Requires longer training; larger parameter count

As demonstrated in Table 1, a substantial number of conventional methods are predicated on spectral information exclusively, or are inadequate in capturing the critical higher-order spatial dependencies inherent to hyperspectral data. Convolutional neural networks (CNNs) with a limited depth exhibit enhancement in the extraction of spatial features; however, they frequently lack the necessary depth to adequately model intricate class separations. Previous comparative studies have shown that deeper CNN architectures significantly outperform shallow ones for HSI classification tasks [30]. Moreover, earlier hybrid approaches combining feature selection with deep belief networks have shown promise but are limited by manual preprocessing and scalability issues [29].

In order to address these deficiencies, we propose the

implementation of a deep 2D CNN architecture that incorporates hierarchical feature learning, regularization layers, and robust generalization capabilities. This approach has been demonstrated to enhance the precision of classification and ensure consistency across diverse vegetation categories in the Salinas dataset. Similar improvements were observed in optimized CNNs incorporating dropout and batch normalization, which have proven effective in recent Informatica studies [31].

2 Description of methodology

This study developed a refined architecture for a 2D convolutional neural network (2D CNN) to classify hyperspectral images (HSI) by simultaneously learning spatial and spectral features from local image patches.The methodology includes the following key components: (1) data preprocessing and patch extraction to prepare hyperspectral images for training; (2) design of a deep 2D CNN architecture, incorporating dropout and batch normalization layers for regularization; and (3) implementation of a training and validation procedure using the Salinas dataset, including hyperparameter tuning, ablation studies, and performance evaluation against standard classifiers. The experimental results demonstrate that our proposed method shows higher classification accuracy and efficiency than conventional methods, thus demonstrating its applicability in advanced remote sensing applications. To ensure the transparency and reproducibility of the research results, a detailed step-by-step description of the experimental setup, implementation, and validation measures of the method is provided in this section.

2.1 2D convolution neural network (2D-CNN)

The 2D convolutional neural network (CNN) is a deep learning model designed primarily for image data processing. It identifies spatial patterns like edges, textures, and shapes through convolutional operations. CNNs have had a major impact on fields such as machine vision, imaging, and self-driving cars, thanks to their ability to accurately detect patterns in 2D data. The 2D CNN is composed of multiple layers that work together to extract features, reduce dimensionality and classify input data.

Input layer

The first entry point for the raw image data in the network is known as the input layer. Generally, this input data is presented in the format of a 2-D matrix containing channels, width and height. The intensity of light is only captured in one channel in the case of greyscale photographs. Whereas in colour images, the three channels represent each of the main colours - red, green and blue (RGB). In more complicated data, like hyperspectral photographs, the input channel layer handles a range of spectral bands instead of just the colour channels that become visible. Generally, we start the analysis at the input layer, where the main features of the image data can be extracted and prepared for further processing in a network.

Convolutional layers

An essential element of a CNN is the convolutional layer, as it helps the template to recognize patterns in the image. This is achieved using smaller filters, or kernels, which move over the image. The filters are multiplied by pixels of the image and then the results added together produce a feature map.This process enables the network to detect key patterns at different levels of detail. In simpler terms, the convolution operation involves multiplying the image pixels by the filter values over a specific area, then adding them up. This is mathematically shown as:

$$h(x,y) = \sum_{i=0}^{m} \sum_{j=0}^{n} I(x-i, y-j) K(i,j)$$

Where the I is the input image, K is the kernel (filter), and $h(x,y)$ is the resulting feature map.

Activation function (ReLU - rectified linear unit)

Activation functions are key for introducing non-linearity in a neural network, helping a model for learning higher complexity patterns. One of the most useful activation functions in convolutional neural networks (CNNs) is the Rectified Linear Unit (ReLU). The ReLU operates using a transformation of the feature map as follows: $ReLU(x) = max(0, x)$. That means negative values are changed to 0, whilst the positive values are left the same. Introducing this nonlinearity enables the network to pick up more complicated patterns in the data, which makes it more efficient at learning from complex inputs.

Pooling layers

An important element of CNN is pooling layers, which reduce the spatial dimension of the feature maps while preserving the most important information. Their purpose consists in reducing the spatial dimensions of feature maps. This helps the model to be more efficient computationally and to control over-adaptation by reducing the network's complexity. However, most clustering is maximum clustering, where the layer examines small sections of the features map and selects the highest value of each. This allows the most significant features to be preserved while eliminating less significant detail, making the network more robust.

Flattening layer

On the flattening layer, any multi-dimensional feature maps produced by the convolution and pooling layers are reduced to a one-dimensional (1D) vector. This is essential since the fully connected layer, which does the final classification, must have the data as a 1D vector. Similarly, a flattening

operation makes sure that all previously extracted important features are ordered in a way that makes them available for accurate prediction. Put simply, the feature maps are 'flattened' into a single row of data, ready for the final decision.

Fully connected layers

The fully connected layers are an important component of any neural network, and they come after the convolution and pooling layers. They work much like a normal neural network, with every neuron connected to all of the neurons in previous layers. In effect, the connected layers take the high-level features extracted from previous layers and put them to work in the context of classification. The neurons in the fully connected layers combine the features extracted from the previous layers to form a representation that the network can use to make predictions. The final fully connected layer typically has as many neurons as there are classification categories. Each neuron in this layer corresponds to a specific class, and the output of each neuron represents the network's confidence in that class.

Output layer

The output layer is the final stage in the model's prediction process. After the features go through the convolution, pooling, and full connection layers, they are set for classification. The output layer's job is to take this data and transform it into a format, usually probability values, that can guide the final decision. This conversion is done using an activation function. For binary classification, where there are only two possible classes, the Sigmoid function is used. In contrast, when dealing with more than two classes, the Softmax function is applied for multi-class classification.

2.2 HSI classification based on the 2D CNN

A 2D CNN is a type of deep learning model that focuses on identifying spatial patterns in hyperspectral images (HSIs). Unlike 1D CNNs, which focus on analyzing spectral information sequentially for each pixel, 2D CNNs process local spatial neighborhoods as 2D patches, allowing the model to learn spatial context, textures, and relationships between neighboring pixels. This capability complements the spectral discrimination offered by 1D CNNs and improves classification performance in scenarios where spatial patterns are important. In HSI classification, 2D CNNs autonomously retrieve spatial information via convolutional filters, enhance them with pooling layers for efficiency, and ultimately classify them using fully connected layers. In contrast to conventional techniques such as KNN or SVM, which necessitate manual feature selection, 2D CNNs autonomously discern patterns, yielding enhanced accuracy and superior generalization.

Data preprocessing for HSI

Prior to the implementation of the hyperspectral data into the 2D CNN framework, the application of a preprocessing pipeline becomes imperative, thus enhancing data quality and model performance. The raw Salinas dataset comprises 224 spectral bands; however, 20 bands (108–112, 154–167, and 224) are excluded due to elevated atmospheric noise, resulting in 204 usable bands. Each spectral band is then subjected to min-max scaling in order to map pixel values to the [0, 1] range. In the subsequent stage of the process, local patches measuring 11 by 9 pixels are extracted in the vicinity of each labelled pixel. These patches are utilised to preserve spatial context and function as input samples, with the class label assigned from the central pixel. It is imperative to note that patches located in proximity to the image borders are excluded with a view to ensuring the maintenance of consistent input dimensions. The application of dimensionality reduction techniques, such as PCA, is eschewed in order to ensure the preservation of full spectral information during the training of convolutional layers. The purpose of the pre-processing stage is to ensure that the model training is facilitated through the utilisation of inputs that are both clean and normalised, and which are imbued with spatial awareness.

Input layer for HSI

When classifying hyperspectral images (HSI) using 2D CNNs, the input layer processes the image as a 3D matrix consisting of height, width, and multiple spectral bands. Hyperspectral imagery captures a much wider range of wavelengths, including those beyond visible light, unlike conventional color imagery which has only three channels (RGB). For each pixel, the intensity values in these bands are stored, providing rich spectral detail. A 2D CNN is built to process this complex data, allowing the model to detect patterns and precisely classify different materials and objects based on their unique spectral signatures.

Convolutional layers in HSI

During HSI classification, the convolutional layers extract the key features from HSI images by using filters to identify patterns in different spectral bands. As HSI data includes many more bands than traditional imagery, these layers concentrate on extracting local spatial patterns from the hyperspectral image data, utilizing the rich spectral channels as feature inputs to learn spatially correlated structures across multiple bands.

Activation function with HSI

The Rectified Linear Unit (ReLU) activation function is often applied after the convolutional layers in the context of hyperspectral image (HSI) classification. ReLU allows the network to learn more complex patterns, which are essential for accurate HSI classification by introducing non-linearity

into the model. All negative data is set at zero and all positive data is not changed.

Pooling layers in HSI

The pooling layer is used to reduce the size of the feature map while retaining important information for classification of hyperspectral images (HSIs). The pooling layer reduces the size of the feature map after the convolution layer has extracted spatial and spectral features, while preserving the essential information. The most common pooling methods are maximum pooling, which retains the highest value in a region, and mean pooling, which calculates the average.This step helps the model to retain the most significant spatial features while reducing spatial resolution, which improves robustness to local variations and reduces computational cost.

Flattening layer in HSI

The flattening layer is essential for generating features for the final classification stage of hyperspectral image (HSI) classification. By passing through convolution and pooling layers that capture key spatial and spectral features, the image is transformed into a multi-dimensional feature map. To make this data usable for classification, the flattening layer converts it into a one-dimensional (1D) vector, allowing the fully connected layers to process and classify the information effectively

Fully connected layers in HSI

In hyperspectral image (HSI) classification, after the feature maps have been flattened and passed through the fully connected layers, a prediction is made in the final layer based on the learned patterns. Since the number of neurons in this layer corresponds to several classes in the dataset (e.g., land cover type, material), the prediction is selected from the class with the highest output.

Output layer for HSI

In hyperspectral image (HSI) classification, the output layer plays a vital role in converting the features extracted by the network into actionable predictions. After the convolutional, pooling, and fully connected layers have processed the hyperspectral data, the output layer uses an activation function to produce the final classification results. Softmax is generally used for HSI classification when there are multiple classes, such as various land cover types, materials or environmental features. Sigmoid activation is frequently used for binary HSI classification (e.g. distinguishing between two types of land cover).

2.3 Proposed architecture for hyperspectral image classification using 2D CNN

To improve the efficiency of feature extraction and classification, we present a new 2D convolutional neural network (CNN) architecture. Through a series of convolutional layers and max-pooling operations, the network progressively decreases the spatial dimension of the input data while preserving important features. With our method, we want to achieve better performance by effectively capturing and improving key patterns at each step, making sure that the most important data is preserved throughout the network.

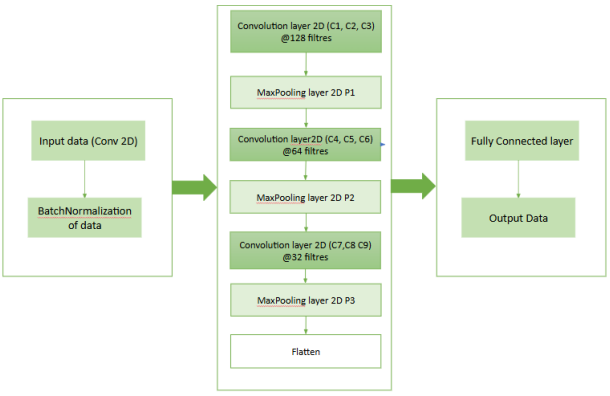


Figure 1: Overview of the proposed deep 2D-CNN architecture used for HSI. The model includes 9 convolutional layers grouped into three blocks with max-pooling, batch normalization, and dropout layers for regularization.

Figure 1 illustrates the proposed deep 2D CNN architecture. The model is composed of three convolutional blocks, each followed by max-pooling and dropout. Batch normalization is integrated to stabilize and accelerate training. The final dense layers perform classification across the 16 target classes. This design enables the model to learn both low- and high-level spatial–spectral features crucial for hyperspectral image analysis.

In the proposed 2D Convolutional Neural Network (CNN), batch normalization layers are interspersed after convolutional layers within each block to normalize the intermediate activations. This helps stabilize and accelerate training, improving model convergence and generalization. Our proposed architecture first normalizes and divides the HSI data into training and test sets, and merges the features before feeding them into the 2D-CNN for classification. The model is structured with three main sets of convolutional layers 2D (C_1, C_2, C_3), each containing three layers. In total there are nine convolutional 2D layers $(c_1, c_2.....c_9)$ and three max-pooling 2D layers (P_1, P_2, P_3). Also for each group, the filters are configured as $K_1 = 128$, $K_2 = 64$, and $K_3 = 32$ respectively. For the input data, we assume the dimensions are (img-height, img-width, img-channels). At the first level, the data is moved through a first set of convolutional layers C_1 with K_1 filters, which transforms the data to (img-height, img-width, K1). The data is then passed through the second convolution C_2 us-

ing K_2 filtering, and the spatial dimensioning is adapted according to the step and pad used. Finally, after passing through the third set of convolutional layers C_3 and the max-pooling layers P_3, the output dimensions are reduced to (img-height, img-width, K3) depending on the pooling strategy used. Dropout layers are incorporated after each max-pooling block and between dense layers to improve regularization and reduce overfitting.

Training approaches

We propose a training approach for 2D convolutional neural networks (CNNs) to optimize the network parameters efficiently for hyperspectral image classification. It consists of two main phases: Forward Propagation and Backpropagation. In the forward propagation stage, the input hyperspectral data is processed through several convolutional and max-pooling layers, allowing the network to extract essential spatial features. In the back-propagating stage, the weights of the network are adjusted by calculating the slopes of the loss function which measures the difference between the predicted output and the actual label.

Forward propagation: The input image of (h, w, c) dimensions will be passed through multiple convolution layers, each of which will be applied to the image to detect different features. The convolution operation produces outputs. These outputs are processed as follows: for each layer i, the input a_i is transformed using the equation

$$a_{i+1} = g_i(z_i) \tag{1}$$

where$z_i = w_i a_i + b_i$ represents the weighted sum of the input and bias, and our activation function ReLu is g(.) present by $g(z_i) = max(0, z_i)$. After each convolution operation, a max-pooling layer is applied to reduce the spatial dimensions of the data.

Once the data has passed through all the convolution and pooling layers, it is flattened into a 1D vector and fed into a fully connected layer. This layer calculates a weighted sum of the inputs. This produces the final output. Then, the output is run through a softmax function, which converts it into a probability distribution over different classes. The softmax function is defined as:

$$y = \frac{e^{W_1 a_1 + b_1}}{\sum_{i=1}^{K} e^{W_i a_i + b_i}} \tag{2}$$

where W_i represent weights, a_i the input vector, and K the number of classes.

Backpropagation: The training parameters are updated using the gradient descent method, which involves minimizing a cost function and calculating the partial derivative of the loss function concerning each training parameter. Having defined the architecture of a 2D CNN classifier and the associated trainable parameters, we can run and reload any stored parameters to classify hyperspectral image data. This classification process is analogous to the forward propagation step, where we can determine the classification results for the test dataset.

Ablation study

In order to assess the individual contributions of these architectural elements to the overall performance of the proposed deep 2D-CNN, an ablation study was conducted using the Salinas dataset. The classification accuracy when key components of the network are removed or modified is presented in Table 2.

Table 2: Ablation study on the impact of individual architectural components

Model Variant	Accuracy (%)
Full architecture (with dropout, BN, 10 conv layers)	**94.0**
Without dropout layers	91.7
Without batch normalization	90.2
Reduced number of filters (half per layer)	89.5
Without dropout and BN	87.8

The findings substantiate the significance of each design decision. The elimination of dropout resulted in a 2.3% reduction in accuracy, while the omission of batch normalization caused an even more substantial decline (3.8%). A 4.5% decrease in accuracy was observed as a consequence of the reduction in the number of filters, which consequently diminished the model's capacity for feature extraction. The most marked decline was observed in the simultaneous removal of batch normalisation and the dropout layer. This result is particularly noteworthy as it underscores the collective impact of these two regularisation mechanisms, highlighting their synergetic role in the model's training. This study emphasises the significance of architectural depth and regularisation techniques in attaining optimal classification performance in hyperspectral image classification.

Training configuration and validation strategy

The proposed deep two-dimensional convolutional neural network (2D-CNN) was implemented utilising the Python programming language and the Keras framework with a TensorFlow backend. All experiments in this study were conducted on a workstation equipped with an NVIDIA RTX 3080 GPU (10 GB VRAM), an Intel Core i7 processor, and 32 GB of RAM.

The model underwent a training phase that spanned 100 epochs. This training was facilitated by the use of the Adam optimiser, with a initial learning rate of 0.001 and a batch

size of 256. The loss function was defined as categorical cross-entropy. It was estimated that the overall duration of the training programme was approximately 120 minutes per session. To facilitate the monitoring of performance and the prevention of overfitting, a proportion of the training data was reserved for use as a validation set. This was accomplished through the implementation of early stopping with a patience of 10 epochs, based on the validation loss.

A range of hyperparameters, including learning rate, batch size, number of filters per layer, and dropout rate, were subject to empirical fine-tuning via a grid search approach. Following the conduction of empirical trials which yielded values ranging from 0.2 to 0.5, it was established that the optimum dropout rate would be 0.3. The assessment of overfitting was conducted through the analysis of both training and validation accuracy and loss curves. Regularization techniques, including dropout and batch normalization, were incorporated into the architecture to enhance the control of overfitting.

3 Experimental assessment and analysis

In this study, we concentrate on assessing the performance of a new 2D-CNN architecture for the classification of hyperspectral images (HSIs), addressing a gap in existing studies which often overlook the efficiency of computation and the spectral feature extraction. In the first section, Study Area and Data Collection for Hyperspectral Imagery, the Salinas Valley HSI dataset, composed largely of vegetated regions, is used to evaluate the classification accuracy of our proposed model. In the second section, Results Interpretation and Analysis, it is demonstrated that our proposed 2D-CNN method improves the classification accuracy significantly for 16 classes, especially for vegetation types, when compared to the traditional k-nearest neighbours (KNN) method. In the final section, Results and Comparisons, our proposed method is presented as outperforming existing techniques in terms of accuracy and computational requirements, while addressing the limitations of focusing on a single dataset. Future studies could confirm the robustness of the approach and explore optimisation strategies under resource-constrained conditions, as well as extending the approach to more diverse environments. In conclusion, this study highlights the potential of 2D CNN-assisted HSI classification to revolutionise remote sensing applications, improving accuracy without increasing resource requirements.

3.1 Study area and data collection for hyperspectral imagery

Hyperspectral imaging

The Hyperspectral sensors Detect the flux intensity for a given surface and a determined wavelength, i.e. a physical quantity in watts per square meter steradian (W/(m²·sr)). More precisely, for every unit of surface area (corresponding to one pixel of the image), this sensor detects the light emitted and reflected by the object as a spectrum of several hundred channels, defining a spectral response curve. For Earth observation, the signals arriving from the Earth's surface are modified by atmospheric disturbances such as clouds, water vapour, atmospheric aerosols, etc. Thus, for surface and land cover remote sensing, the preferred measure is reflectance, defined as the ratio of the emitted flux from the surface to the incident flux. This ratio is an indication of the reflectivity of a particular object for each wavelength band of light. The reflectance is an intrinsic property of the material, independent of the environment, and is therefore very discriminating for classification.

Hyperspectral images are in practice (w; h; B) tensors, i.e. 3-dimensional cubes having two spatial dimensions (width w and height h) and one spectral dimension (with B-bands). This hypercube is anisotropic compared to volumetric data, e.g. seismic data cubes: the three dimensions do not have the same physical displacement. However, linear operations on a 3D subset of the cube are mathematically and physically valid because all values in the hypercube are expressed in the same unit, either luminance or reflectance. When dealing with convolutions and filtering operations on the hypercube, this property will come in handy.

Dataset description

For our studies, we used the Salinas Valley HSI data set, which was collected by the 224-band AVIRIS sensor over the Salinas Valley in California and is characterised by its high spatial resolution (3.7-metre pixels).

The figure 2 shows the image and the ground truth of the Salinas Valley, respectively, acquired with the with the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor in October 1998. The HSI consists of 512217 pixels and has a resolution of 3.7m. The 400-2500nm spectral data were recorded in 224 bands, of which the 108–112 (5 bands), 154–167 (14 bands), and band 224 (1 band) were removed due to high atmospheric absorption, totaling 20 bands removed and leaving 204 reliable bands. Identifying, measuring and monitoring the composition of the Earth's surface and atmosphere was the main objective of the AVIRIS project. Earth's surface and atmosphere from the signatures of molecular absorption and particle scattering. Understanding processes related to the global environment and climate change has been the focus of research using AVIRIS data.

Each color in the ground truth map corresponds to one of 16 annotated land cover classes, such as vineyard, lettuce (at multiple growth stages), or fallow soil. These labels serve as the basis for supervised training and performance evaluation of classification models. The map also highlights spatial variability and class imbalance, making it a valuable benchmark for testing spatial–spectral learning architectures like the proposed 2D CNN.

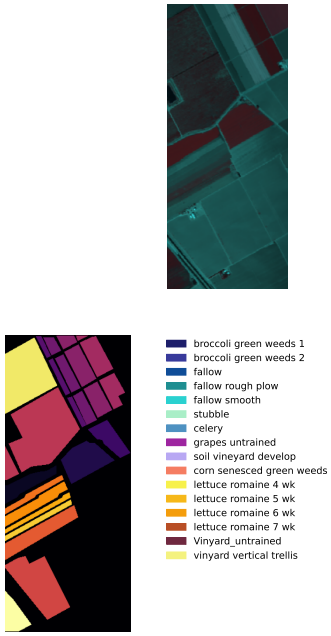


Figure 2: Ground truth label map of the Salinas dataset, containing 16 distinct land cover classes. Each class is color-coded.

To illustrate the diversity of spectral information in hyperspectral images, Figure 3 presents selected individual bands from the Salinas dataset. Each band corresponds to a particular wavelength and reveals different reflectance characteristics. These variations are key to enabling spectral-based classification of similar-looking surface materials, a core strength of CNN-based models.

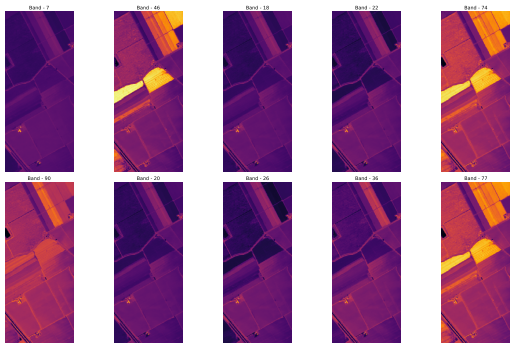


Figure 3: Sample spectral bands from the Salinas hyperspectral image (e.g., Band 7, Band 22, Band 90).

Our studies used the remaining 204 bands after removing 20 water absorption bands (see figure 3). This scene was acquired with the AVIRIS sensor over the Salinas Valley, California, and is characterised by high spatial resolution (3.7-metre pixels). The dataset includes different land types such as vegetables, bare soil, and vineyards, with a total of 16 different classes.

3.2 Result interpretation and analysis

Training and testing data

After integration, the data needs to be input into the 2D-CNN network. It is important before this step to split the data into training and test groups and to delete any background pixel samples. In our method, we have separated our data into 0.7 for training and 0.3 for testing. this process of preparation reduces the computational burden and more details can be found in Table 3.

Table 3: The number of training and test samples in the Salinas dataset

Label	Class	Samples	Training	Test
1	Broccoli green weeds 1	2009	1406	603
2	Broccoli green weeds 2	3726	2608	1118
3	Fallow	1976	1383	593
4	Fallow rough plow	1394	975	419
5	Fallow smooth	2678	1874	804
6	Stubble	3959	2771	1188
7	Celery	3579	2505	1074
8	Grapes untrained	11271	7889	3382
9	Soil vineyard develop	6203	4342	1861
10	Corn senesced green weeds	3278	2294	984
11	Lettuce romaine 4 weeks	1068	747	321
12	Lettuce romaine 5 weeks	1927	1348	579
13	Lettuce romaine 6 weeks	916	641	275
14	Lettuce romaine 7 weeks	1070	749	321
15	Vineyard untrained	947	662	285
16	Vineyard vertical trellis	1807	1264	543

Description of program

The Python program is implemented using the Keras API, which provides a high-level API for neural networks that can be executed on top of TensorFlow. We import the Sequential class from the Keras library, which we use to initialise a sequential model. This is composed of multiple layers, which contain Conv2D, BatchNormalisation, MaxPooling2D, Dropout, Flatten and Dense (see table 4).

The Conv2D is a two-dimensional convolutional layer which operates on the input data by convolution with a specified number of filters and kernel size, using the ReLU activation function to introduce non-linearity. In our proposed model, multiple Conv2D layers are successively applied to extract progressively more complex features from the input data. BatchNormalisation is used to normalize the input data by adapting and scaling the activations of the previous layers. MaxPooling2D is applied to decrease the spatial size of the data by selecting the maximum value within a given pool size, thus reducing the computational load and controlling overfitting. The Dropout controls overfitting by randomly dropping a percentage of the input units. Doing this helps the model to generalise better and avoid over-reliance on any particular feature. The Flatten layer is used to transform the multi-dimensional output of the convolutional and pooling layers into a one-dimensional array that can be passed to the fully connected (dense) layers. Density is a fully connected layer which applies a linear operation to input data. The initial Density layer has 25 entities, followed by a Drop Out layer to continue to ad-

Table 4: 2D CNN model design and parameter summary (with batch normalization after each Conv2D layer)

Layer (type)	Output Shape	Param
InputLayer	(None, 11, 9, 128)	2,432
Layer1 (Conv2D)	(None, 11, 9, 128)	147,584
BatchNorm1 (BatchNormalization)	(None, 11, 9, 128)	512
Layer2 (Conv2D)	(None, 11, 9, 128)	147,584
BatchNorm2 (BatchNormalization)	(None, 11, 9, 128)	512
Layer3 (Conv2D)	(None, 11, 9, 128)	147,584
BatchNorm3 (BatchNormalization)	(None, 11, 9, 128)	512
MaxPooling_Layer1 (MaxPooling2D)	(None, 5, 4, 128)	0
Dropout1 (Dropout)	(None, 5, 4, 128)	0
Layer4 (Conv2D)	(None, 5, 4, 64)	73,792
BatchNorm4 (BatchNormalization)	(None, 5, 4, 64)	256
Layer5 (Conv2D)	(None, 5, 4, 64)	36,928
BatchNorm5 (BatchNormalization)	(None, 5, 4, 64)	256
Layer6 (Conv2D)	(None, 5, 4, 64)	36,928
BatchNorm6 (BatchNormalization)	(None, 5, 4, 64)	256
MaxPooling_Layer2 (MaxPooling2D)	(None, 2, 2, 64)	0
Dropout2 (Dropout)	(None, 2, 2, 64)	0
Layer7 (Conv2D)	(None, 2, 2, 32)	18,464
BatchNorm7 (BatchNormalization)	(None, 2, 2, 32)	128
Layer8 (Conv2D)	(None, 2, 2, 32)	9,248
BatchNorm8 (BatchNormalization)	(None, 2, 2, 32)	128
Layer9 (Conv2D)	(None, 2, 2, 32)	9,248
BatchNorm9 (BatchNormalization)	(None, 2, 2, 32)	128
MaxPooling_Layer3 (MaxPooling2D)	(None, 1, 1, 32)	0
Dropout3 (Dropout)	(None, 1, 1, 32)	0
Flatten (Flatten)	(None, 32)	0
DenseLayer (Dense)	(None, 25)	825
Dropout4 (Dropout)	(None, 25)	0
DenseLayer1 (Dense)	(None, 16)	442

dress overfitting. The second Dense layer uses 16 units and returns the class probabilities using the softmax activation function. Finally, the model is summarised using the model.summary() function, which outputs the layers, shapes and parameters of the model.

3.3 Findings and comparisons

With a focus on improving accuracy and computational efficiency, this study investigated the effectiveness of a novel 2D CNN architecture for hyperspectral image (HSI) classification. However, previous work on HSI classification has often focused on spatial feature extraction or dimension reduction, without considering a trade-off between high classification accuracy and resource usage. When applying these methods to large datasets or edge computing environments where computational resources are limited, this gap becomes critical.

Our model used a standard training procedure with 100 epochs, a batch size of 256, categorical cross-entropy loss, and an Adam optimizer after the data were input into the CNN architecture (Figure 1). The table 4 shows the specifications of each 2D CNN layer, providing a detailed breakdown of the architecture. We have use the Salinas dataset to evaluate the classification accuracy for our method and we have compared our results with the traditional k-nearest neighbours (KNN) classifier.

The classification accuracy for each class is summarized in Table 5, showing that our proposed 2D CNN method

Table 5: Accuracy per class for Salinas dataset comparing our method with the KNN method

Label	Class	KNN	Proposed Method
1	Broccoli green weeds 1	0.99	1.0
2	Broccoli green weeds 2	0.98	1.0
3	Fallow	0.70	0.99
4	Fallow rough plow	0.98	0.99
5	Fallow smooth	0.84	0.99
6	Stubble	0.97	1.0
7	Celery	1.0	1.0
8	Grapes untrained	0.78	0.88
9	Soil vineyard develop	0.84	1.0
10	Corn senesced green weeds	0.73	0.98
11	Lettuce romaine 4 weeks	0.92	0.98
12	Lettuce romaine 5 weeks	0.89	0.99
13	Lettuce romaine 6 weeks	0.91	1.0
14	Lettuce romaine 7 weeks	0.88	0.99
15	Vineyard untrained	0.56	0.81
16	Vineyard vertical trellis	0.98	1.0
	Accuracy	0.88	0.94

achieved higher accuracy than KNN on 14 out of 16 classes, matched performance on 1 class, and showed no significantly lower accuracy on any class. Our main results show that the proposed 2D CNN approach provides significantly higher classification accuracy for vegetation classes, such as broccoli, celery, and vineyard, where spatial textures and contextual patterns are critical for differentiation. This improved performance stems from the ability of 2D CNNs to jointly model spatial and spectral information within local patches, allowing the network to capture subtle variations in both spectral signatures and spatial arrangements of crop types.

In addition to KNN, we benchmarked our model against traditional classifiers including SVM and Random Forest, as well as a shallow CNN (LeNet-5), with results summarized in Table 6.

Table 6: Comparative evaluation of classification performance on the Salinas dataset

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Random Forest (RF)	87.2%	0.88	0.86	0.87
Support Vector Machine (SVM)	86.0%	0.87	0.85	0.86
Shallow 2D CNN (LeNet-5)	91.1%	0.91	0.90	0.90
k-Nearest Neighbors (KNN)	88.0%	0.89	0.87	0.88
Proposed Deep 2D CNN	**94.0%**	**0.96**	**0.93**	**0.95**

To further validate our model, we compared it with Support Vector Machine (SVM), Random Forest (RF), and a shallow 2D CNN (LeNet-5), in addition to k-nearest neighbors (KNN). As shown in Table 6, the proposed deep 2D CNN achieved the highest accuracy and also outperformed all baselines in terms of precision, recall, and F1-score. The shallow CNN performed better than the traditional classifiers, which emphasizes the advantage of spatial feature learning. However, it still fell short of the deep architecture in capturing complex spatial–spectral patterns. These results confirm that deeper CNNs, with dropout and batch normalization, are more capable of handling hyperspectral classification challenges.

The classification maps presented in Figures 4 and 5 demonstrate the advantages of our proposed 2D CNN ap-

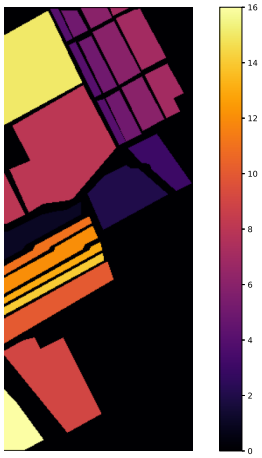


Figure 4: Ground truth classification map of the Salinas Valley dataset showing 16 land cover classes. Each color corresponds to a different class as indicated by the colorbar on the right.

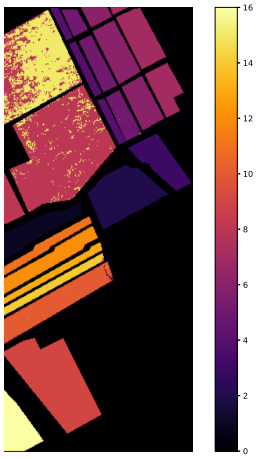


Figure 5: Predicted classification map using the proposed 2D CNN method on the Salinas Valley dataset. Each color corresponds to a predicted land cover class as shown in the colorbar.

proach, especially its accurate identification of vegetation areas, which aligns closely with the ground truth data. The classification results in Figure 5 are consistent with quantitative results summarized in Table 5, including a 94% overall accuracy and high per-class accuracies. This supports the model's effectiveness in capturing spatial–spectral patterns across crop types. Each color in the maps corresponds to a specific land cover class: dark blue for broccoli green weeds 1, deep blue for broccoli green weeds 2, blue for fallow, cyan for fallow rough plow, light cyan for fallow smooth, teal for stubble, light blue for celery, purple for grapes untrained, lavender for soil vineyard develop, orange for corn senesced green weeds, light yellow for lettuce romaine 4 weeks, yellow for lettuce romaine 5 weeks, orange-yellow for lettuce romaine 6 weeks, dark red for lettuce romaine 7 weeks, brown for vineyard untrained, and pale yellow for vineyard vertical trellis.

Our method demonstrates that enhanced spectral feature extraction without negatively impacting computational efficiency, a major limitation of previous work, when comparing our results to other studies. For example, by simultaneously integrating both spectral and spatial domain information, our approach achieves superior classification performance compared to models that focus solely on either spectral or spatial features, achieves superior classification performance without the heavy computational burden typically associated with 2D CNNs. Nevertheless, our study was limited to the Salinas dataset, which contains vegetation data. This limitation means that our results may not be generalisable to datasets containing more diverse or urban environments. More research is necessary to confirm the stability of the method in different types of hyperspectral data and environments. Our study demonstrates that the suggested 2D CNN method is more robust to spectral noise and outperforms traditional classifiers in vegetation-related classification tasks. Future studies may explore the implementation of this architecture on other hyperspectral datasets, with focus on optimising 2D CNN power under constrained computational resources while maintaining high accuracy. Recent observations suggest that the proposed 2D CNN architecture significantly enhances HSI classification accuracy. Our findings provide conclusive evidence that this improvement is associated with a balanced approach to handling complex spectral datasets, rather than increased computational costs. This enhanced classification method has the potential to benefit remote sensing applications, particularly in agriculture and environmental monitoring, where both efficiency and accuracy are crucial.

3.4 Discussion

A comparative analysis of the performance of the proposed deep 2D-CNN model with that of multiple state-of-the-art classification techniques has been conducted on the Salinas dataset. The objective of this analysis is to furnish a more exhaustive evaluation of the efficacy of the proposed model. The proposed methodology demonstrated an aggregate accuracy of 94%, which exceeds the performance metrics of the conventional k-nearest neighbor (kNN) classifier (88%), support vector machines (SVM, 86%), and multinominal logistic regression (MLR, 83%). As the study illustrates, the model under consideration demonstrates a superior level of class-level performance, particularly with respect to differentiating between vegetation types, such as broccoli, celery, and vineyard classes.

The categories identified are particularly conducive to analysis due to the inherent fine-grained spectral and spatial patterns characteristic of this phenomenon. This enhancement in performance can be ascribed to the advanced con-

figuration of the network architecture, which encompasses the integration of ten convolutional layers, in conjunction with batch normalization and dropout mechanisms. These augmentations have been shown to enhance the efficacy of feature extraction while concomitantly preventing the occurrence of overfitting. In comparison with shallow CNN models, the depth of our network enables hierarchical feature learning, thereby facilitating the capture of both low- and high-level spatial features.

With respect to computational efficiency, the proposed model necessitates a greater investment of training time due to its complexity and parameter density. However, this investment is justified, as the training process remains manageable (approximately 120 minutes per epoch on a GPU) and the subsequent inference is efficient once trained. It is noteworthy that the total number of trainable parameters (approximately 630,000) is reasonable in comparison to the most advanced deep architectures currently available, thereby ensuring a balanced compromise between achieving the desired level of accuracy and optimizing resource allocation.

In summary, the proposed architecture is designed to balance three primary objectives: accuracy, generalization, and computational feasibility. These characteristics position the architecture as a suitable candidate for practical applications in the domain of large-scale remote sensing.

Table 7: Published benchmarks on Salinas dataset (selected studies)

Study	Architecture	Accuracy	Notes
Chen et al. (2016)	3-layer CNN	91.5%	Shallow network, limited spatial modeling
Zhao and Du (2016)	Spectral–spatial deep model	92.8%	Combines 1D and 2D filtering
Vaddi et al. (2020)	Integrated 1D-2D CNN	93.4%	Hybrid model; high complexity
This work	Deep 2D CNN (10 layers)	**94.0%**	Strong regularization; efficient architecture

As demonstrated in Table reftab:benchmark, the proposed deep 2D-CNN demonstrates a competitive performance in comparison to previously published methods on the Salinas dataset. While other studies such as the ones by Chen et al. (2016) and Zhao and Du (2016) have demonstrated accuracies within the 91–93 percent range, the proposed architecture in this study has been shown to outperform these benchmarks. This is achieved without significantly increasing the required training time and without the need for a more complex model compared to hybrid or 3D approaches. This underscores its viability for practical implementation in large-scale HSI classification.

4 Conclusion

This study presents a new technique using convolutional neural networks, 2D-CNN, for classifying hyperspectral images (HDSIs). Firstly, the data is normalised for the extraction of both spatial and spectral features. The proposed 2D CNN architecture consists of three sets of convolutional and pooling layers, with batch normalization (BN) and dropout mechanisms incorporated to improve generalization and reduce overfitting. The classification approach has been evaluated on the Salinas Valley dataset, where it outperformed a traditional K-Nearest Neighbors (KNN) baseline. Future work will expand the evaluation to additional benchmark datasets. While the proposed method demonstrates strong classification performance on the Salinas dataset, the evaluation is currently limited to this single agricultural context. As hyperspectral image characteristics may vary across different geographical and spectral domains, further testing on additional benchmark datasets is essential to assess generalizability.
Future work will include experiments on the Pavia University and Indian Pines datasets, which contain more diverse land cover types and urban scenes. These datasets will allow us to validate the robustness of the proposed architecture in handling different spatial-spectral distributions and scene complexities.

Future work will focus on reducing the running time of the algorithm and applying the proposed method to a wider range of HSI datasets using 3D CNNs. We will also investigate the merging of 1D and 2D CNNs and 2D and 3D CNNs.

References

[1] Bioucas-Dias, José M., Plaza, Antonio, Camps-Valls, Gustavo, Scheunders, Paul, Nasrabadi, Nasser, Chanussot, Jocelyn, *Hyperspectral remote sensing data analysis and future challenges*, IEEE Signal Processing Magazine, vol. 31, no. 1, pp. 6–36, IEEE, 2013. DOI: 10.1109/MGRS.2013.2244672

[2] Chen, Yushi, Lin, Zhouhan, Zhao, Xing, Wang, Gang, Gu, Yanfeng, *Deep learning-based classification of hyperspectral data*, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 7, no. 6, pp. 2094–2107, IEEE, 2014. DOI: 10.1109/JSTARS.2014.2329330

[3] Ham, Jisoo, Chen, Yangchi, Crawford, Melba M., Ghosh, Joydeep, *Investigation of the random forest framework for classification of hyperspectral data*, IEEE Transactions on Geoscience and Remote Sensing, vol. 43, no. 3, pp. 492–501, IEEE, 2005. DOI: 10.1109/TGRS.2004.842481

[4] Ma, Li, Crawford, Melba M., Tian, Jinwen, *Local manifold learning-based k-nearest-neighbor for hyperspectral image classification*, IEEE Transactions on Geoscience and Remote Sensing, vol. 48, no. 11, pp. 4099–4109, IEEE, 2010. DOI: 10.1109/TGRS.2010.2055876

[5] Li, Jun, Bioucas□Dias, José M., Plaza, Antonio, *Spectral–spatial hyperspectral image segmentation*

using subspace multinomial logistic regression and Markov random fields, IEEE Transactions on Geoscience and Remote Sensing, vol. 50, no. 3, pp. 809–823, IEEE, 2012. DOI: 10.1109/TGRS.2011.2166287

[6] Melgani, Farid, Bruzzone, Lorenzo, *Classification of hyperspectral remote sensing images with support vector machines*, IEEE Transactions on Geoscience and Remote Sensing, vol. 42, no. 8, pp. 1778–1790, IEEE, 2004. DOI: 10.1109/TGRS.2004.831865

[7] Mughees, Atif, Tao, Linmi, *Efficient deep auto-encoder learning for the classification of hyperspectral images*, in *Proceedings of the 2016 International Conference on Virtual Reality and Visualization (ICVRV)*, pp. 44–51, IEEE, 2016. DOI: 10.1109/ICVRV.2016.16

[8] Chen, Yushi, Zhao, Xing, Jia, Xiuping, *Spectral–spatial classification of hyperspectral data based on deep belief network*, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 8, no. 6, pp. 2381–2392, IEEE, 2015. DOI: 10.1109/JSTARS.2015.2388577

[9] Chen, Yushi, Jiang, Hanlu, Li, Chunyang, Jia, Xiuping, Ghamisi, Pedram, *Deep feature extraction and classification of hyperspectral images based on convolutional neural networks*, IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 10, pp. 6232–6251, IEEE, 2016. DOI: 10.1109/TGRS.2016.2584107

[10] Farooq, Adnan, Hu, Jiankun, Jia, Xiuping, *Weed classification in hyperspectral remote sensing images via deep convolutional neural network*, in *Proceedings of IGARSS 2018 – IEEE International Geoscience and Remote Sensing Symposium*, pp. 3816–3819, IEEE, 2018. DOI: 10.1109/IGARSS.2018.8518541

[11] Gómez□Chova, Luis, Tuia, Devis, Moser, Gabriele, Camps□Valls, Gustau, *Multimodal classification of remote sensing images: A review and future directions*, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 103, no. 9, pp. 1560–1584, IEEE, 2015. DOI: 10.1109/JSTARS.2015.2399651

[12] Ahmad, Muhammad, Shabbir, Sidrah, Roy, Swalpa Kumar, Hong, Danfeng, Wu, Xin, Yao, Jing, Khan, Adil Mehmood, Mazzara, Manuel, Distefano, Salvatore, Chanussot, Jocelyn, *Hyperspectral image classification—Traditional to deep models: A survey for future prospects*, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 15, pp. 968–999, IEEE, 2021. DOI: 10.1109/JSTARS.2021.3051290

[13] Kong, Yi, Wang, Xuesong, Cheng, Yuhu, *Spectral–spatial feature extraction for HSI classification based on supervised hypergraph and sample expanded CNN*, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 11, no. 11, pp. 4128–4140, IEEE, 2018. DOI: 10.1109/JSTARS.2018.2849526

[14] Yuan, Yuan, Zhu, Guokang, Wang, Qi, *Hyperspectral band selection by multitask sparsity pursuit*, IEEE Transactions on Geoscience and Remote Sensing, vol. 53, no. 2, pp. 631–644, IEEE, 2015. DOI: 10.1109/TGRS.2014.2331215

[15] Sharma, Vivek, Diba, Ali, Tuytelaars, Tinne, Van Gool, Luc, *Hyperspectral CNN for image classification & band selection, with application to face recognition*, Technical Report KUL/ESAT/PSI/1604, KU Leuven, ESAT, Leuven, Belgium, 2016. DOI: 10.3390/pr11020435

[16] Zhao, Wenzhi, Du, Shihong, *Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach*, IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 8, pp. 4544–4554, IEEE, 2016. DOI: 10.1109/TGRS.2016.2543748

[17] Yu, Shiqi, Jia, Sen, Xu, Chunyan, *Convolutional neural networks for hyperspectral image classification*, *Neurocomputing*, vol. 219, pp. 88–98, Elsevier, 2017. DOI: 10.1016/j.neucom.2016.09.010

[18] Vaddi, Radhesyam, Manoharan, Prabukumar, *Hyperspectral image classification using CNN with spectral and spatial features integration*, *Infrared Physics & Technology*, vol. 107, pp. 103296, Elsevier, 2020. DOI: 10.1016/j.infrared.2020.103296

[19] Lee, Hyungtae, Kwon, Heesung, *Going deeper with contextual CNN for hyperspectral image classification*, IEEE Transactions on Image Processing, vol. 26, no. 10, pp. 4843–4855, IEEE, 2017. DOI: 10.1109/TIP.2017.2716361

[20] Bera, Somenath, Shrivastava, Vimal K., Satapathy, Suresh Chandra, *Advances in hyperspectral image classification based on convolutional neural networks: A review*, *CMES – Computer Modeling in Engineering & Sciences*, vol. 133, no. 2, pp. [check pages], 2022. DOI: 10.32604/cmes.2022.014019

[21] Huang, Jiadong, He, Hongyuan, Lv, Rulin, Zhang, Guangteng, Zhou, Zongxian, Wang, Xiaobin, *Non-destructive detection and classification of textile fibres based on hyperspectral imaging and 1D□CNN*, Analytica Chimica Acta, vol. 1224, pp. 340238, Elsevier, 2022. DOI: 10.1016/j.aca.2022.340238

[22] Nouna, Soumaya, Nouna, Assia, Mansouri, Mohamed, Tammouch, Ilyas, Achchab, Boujamaa, *Two-dimensional Klein-Gordon and Sine-Gordon*

numerical solutions based on deep neural network, *International Journal of Artificial Intelligence*, vol. 14, no. 2, pp. 1548–1560, 2022. DOI: 10.11591/ijai.v14.i2.pp1548-1560

[23] Farokhah, Lia, Sarno, Riyanarto, Fatichah, Chastine, *Simplified 2D CNN architecture with channel selection for emotion recognition using EEG spectrogram*, IEEE Access, vol. 11, pp. 46330–46343, IEEE, 2023. DOI: 10.1109/ACCESS.2023.3269138

[24] Diakité, Alou, Jiangsheng, Gui, Xiaping, Fu, *Hyperspectral image classification using 3D–2D CNN*, International Journal of Applied Earth Observation and Geoinformation, vol. 15, no. 5, pp. 1083–1092, Wiley, 2021. DOI: 10.1080/17538947.2021.1879621

[25] Luo, Yanan, Zou, Jie, Yao, Chengfei, Zhao, Xiaosong, Li, Tao, Bai, Gang, *HSI□CNN: A novel convolution neural network for hyperspectral image*, in *Proceedings of the 2018 International Conference on Audio, Language and Image Processing (ICALIP)*, pp. 464–469, IEEE, 2018. DOI: 10.1109/ICALIP.2018.8628313

[26] Feng, Fan, Wang, Shuangting, Wang, Chunyang, Zhang, Jin, *Learning deep hierarchical spatial–spectral features for hyperspectral image classification based on residual 3D–2D CNN*, Remote Sensing, vol. 19, no. 23, pp. 5276, MDPI, 2019. DOI: 10.3390/rs19235276

[27] Zhang, Haitao, Meng, Lingguo, Wei, Xian, Tang, Xiaoliang, Tang, Xuan, Wang, Xingping, Jin, Bo, Yao, Wei, *1D□convolutional capsule network for hyperspectral image classification*, *arXiv preprint* arXiv:1903.09834, 2019. (No DOI available)

[28] Nguyen, Duc Huy, Chao, Paul C.-P., Chung, Chih□Chieh, Horng, Ray□Hua, Choubey, Bhaskar, *Detecting atrial fibrillation in real time based on PPG via two CNNs for quality assessment and detection*, IEEE Sensors Journal, vol. 22, no. 24, pp. 24102–24111, IEEE, 2022. DOI: 10.1109/JSEN.2022.3216517

[29] Zhang, J., Chen, F., Liu, Q., *A hybrid approach for hyperspectral image classification using deep belief networks and feature selection*, Informatica, vol. 45, no. 3, pp. 345–355, 2021. DOI: 10.31449/inf.v45i3.3473

[30] Blefari-Melazzi, T., Ricciardi, S., Nardini, F.M., *A comparative study of deep learning architectures for hyperspectral image classification*, Informatica, vol. 44, no. 4, pp. 567–576, 2020. DOI: 10.31449/inf.v44i4.3182

[31] Kostadinov, M., Petrova, N., *Optimized deep CNN for spectral–spatial HSI classification with dropout and batch normalization*, Informatica, vol. 46, no. 2, pp. 178–189, 2022. DOI: 10.31449/inf.v46i2.3877

Hybrid Context Aware Gujarati Spell Correction Using Norvig Algorithm, GRU, and IndicBERT

Brijeshkumar Y. Panchal[1,2,*], Apurva Shah[1]
[1]Computer Science and Engineering Department, Faculty of Technology and Engineering, The Maharaja Sayajirao University of Baroda, Vadodara, Gujarat, India
[2]Computer Engineering Department, Sardar Vallabhbhai Patel Institute of Technology (SVIT)-Vasad, Gujarat Technological University (GTU), Anand, Gujarat, India
E-mail: panchalbrijesh02@gmail.com
[*]Corresponding author

Keywords: Natural Language Processing (NLP), Gujarati, Grammar Checker, Spelling Correction/Checker, IndicBERT, Peter Norvig, GRU

Received: June 23, 2025

Numerous applications in the domain of Natural Language Processing (NLP) rely on spelling and grammatical checks, including email, opinion mining, text summarization, chatbots, and countless more. An individual's credibility, cybersecurity efforts, legal ambiguities, and NLP application performance can all take a hit if they make a mistake when dealing with regional languages such as Assamese, Gujarati, Hindi, etc. In order to lessen the frequency of spelling errors, this article examines and concentrates on Gujarati. In addition to a thorough examination of issues related to the Gujarati language, this article provides up-to-date strategies for fixing spelling mistakes based on context of the word. A novel hybrid approach ensures top-notch Gujarati context aware spelling verification. After thoroughly considering all the suggestions, we used a two-layer GRU network and the IndicBERTv2-SS model, which was fine-tuned only on our curated Gujarati dataset of about 20,000 sentences (70/15/15 split into training, validation, and test), to choose the best correction while keeping the context in mind. Normalization for Gujarati (diacritics, compound characters, and numbers), regex-based tokenization, and edit-distance candidate creation were all part of preprocessing. Researchers used accuracy, precision, and recall to assess the test split. Our proposed IndicBERT-GUJBRIJAPU tool got 93.49% accuracy, 94.46% precision, 90.13% recall and 91.59% F1 Score, which is much better than other approaches for context-aware correction.

Povzetek: Članek predstavi hibridno kontekstno preverjanje pravopisa za gudžaratščino, ki združuje Norvigov algoritem, dvoslojni GRU in izboljšani IndicBERTv2-SS na 20.000 stavkih.

1 Introduction

India boasts a wealth of literature in a variety of regional languages, including Gujarati, Hindi, Tamil, Assamese and many more. For speakers of other languages inside the nation, these languages nevertheless can remain unintelligible. In languages such as Gujarati and Hindi, the context of a statement is quite important in imparting its intended meaning. Natural Language Processing (NLP) discipline seeks to apply grammatical principles and linguistic structures to analyze and comprehend natural languages. By means of natural language in both speech and text, NLP research investigates how computers might understand, analyze, and modify it, hence bridging the distance between humans and technology [1]. The term "language" denotes the natural languages including Gujarati, Hindi, and English in NLP. Preprocessing, an essential part of natural language processing, involves examining the text for spelling errors in order to improve its quality by associating words with their accurate meanings [2].

Many NLP systems use grammar and spelling correction to remove textual data mistakes. These mistakes provide noise that influences syntactic and semantic understanding, therefore influencing the performance of NLP-based systems [3]. For spell-checking and grammar correction, Deep Learning is quite successful since it lets machines learn from past data. From SMS texting to social media (Facebook, Twitter, WhatsApp), text-based data is exploding exponentially and today digital government records and e-newspapers are expanding [4] [5]. Essential components of text processing, spell-checkers help to fix written language mistakes [4]. They point out two main kinds of mistakes: Words not found in the dictionary (e.g., "Gujarti" rather than "Gujarati")- Non-word errors. Real-world Errors: Dictionary words used wrongly in context (e.g., "Their going to the market" instead of "They’re going to the market"). Although spell-checkers for Latin and Western languages have been extensively developed, the great linguistic variety and complicated grammatical structures

of Indian regional languages mean that research on them is still in its early years.

Gujarati is kind similar to Hindi out of the Indo-Aryan branch of the Indo-European language family. Among the 22 officially recognized languages of India, it has over 55.5 million native speakers—4.5% of the nation's total population as per the 2011 census [1]. Though some NLP tools [6] [7] [8] [5] [9] [10] — stemmers, lemmatizers, tiny corpora—exist—the language currently lacks thorough tools for spelling and grammatical correction [3]. Gujarati is widely used, although it lacks basic NLP tools—especially in relation to spell and grammar checking [11]. Available for Gujarati now, the "Saras" spell checker detects spelling mistakes using Directed Acyclic Word Graphs (DAWG). It does not, however, consider prefixes, suffixes, or inflections, therefore creating fresh research prospects for sophisticated spell-checking methods. Gujarati grammar, adhering to rigorous guidelines [12], encompasses Jodani (જોડણી) for correct spelling, Sandhi (સંધિ) for word joining, and Samas (સમાસ) for compound word formation.

The aim of this study is to solve context aware spelling mistakes in Gujarati language. To solve the shortcomings of current spell-checkers and raise Gujarati NLP application accuracy, novel and hybrid approaches are developed and implemented. In this article, the challenges related to context aware spelling checking with Gujarati language is focused and reviewed which offers valuable insights for researchers, programmers, and language technology enthusiasts who are interested in improving current models. Using NLP methods and deep learning, the proposed models seek to:

- Handle grammatical norms and morphological variances;
- Improve error identification and correction for Gujarati text.
- Improve accuracy and precision for Gujarati.
- Improve Gujarati NLP applications' language processing efficiency

The upcoming session addresses the difficulties associated with Gujarati language spelling correction, exploring several methodologies including rule-based, statistical, and deep learning approaches to rectify spelling and grammatical problems in Gujarati. The subsequent lesson presents two models designed to identify and rectify context aware

spelling mistakes in Gujarati language sentences. The initial model employs Peter Norvig's algorithm and a Gated Recurrent Units (GRU) model, trained using a Gujarati word dictionary, to detect and activate errors while analyzing phrase context. Another model employs IndicBERT to finetune the Peter Norvig-based model, enhancing efficiency and accuracy by concentrating on omitted words inside the statement and assigning scores to forecast sentence correctness. The comparative analysis with respect to accuracy, precision, recall and F1 score for correct and incorrect statements with one of the existing tools has been covered in next section.

2 Related work and background theory

2.1 Characteristics and challenges of Gujarati language

Gujarati is a language that is rich in vocabulary. There are a number of inflections for adjectives, verbs, and nouns. The language has 12 vowels and 34consonants, as well as the ideas of matras and half consonants [3] [13]. Gujarati has several characters with practically same phonetic characteristics. Matras' sounds match those of a vowel: આ, ઇ, ઈ, ઉ, ઊ, એ, ઐ, ઓ, ઔ, અં, and અઃ. All consonants possess inherent vowels. Furthermore, In Gujarati, vowels and constants can stand on their own or be accompanied by one or more matras, which are seen as distinct characters after the vowels and constants. A total of twelve possible word usages exists for each constant, as it is possible for it to appear with each of the eleven matras. In this manner, a total of 374 permutations of constant and matra are generated by combining all 34 constants with 11 matras [3]. Therefore, matras must be handled with due diligence during computer processing [13]. In the Gujarati language, the phonemes િ (e) and ી (ee) are identical, differing only in their degree of extension. The situation is analogous for ુ (u) and ૂ (oo). Consequently, words containing these characters are frequently misspelled. For instance, both પૂજા and પુજા are pronounced as '{pooja}' and signify ' worship'. In contrast to English, prepositions such as in and to can take on suffix inflections within the word, and words can even have several inflections complicates the process of spelling error detection and correction. As a highly inflected language, it is challenging to compile all potential word forms in a lexical dictionary for a spelling

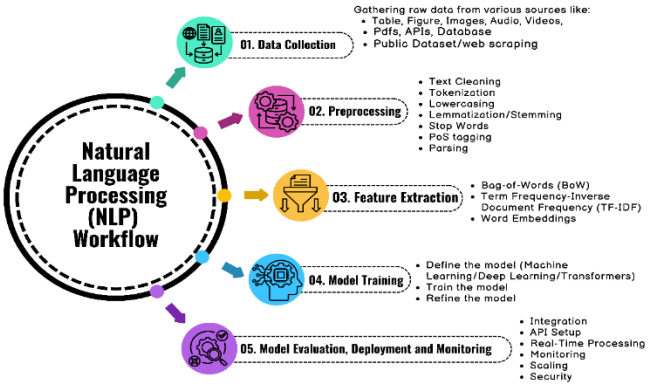


Figure 1: Natural Language Processing (NLP) workflow

2.2 Various spelling and grammar checking approaches

Creating a humanoid—the most intelligent machine ever—is the ultimate objective of artificial intelligence. An important consideration in this development process is the interaction between humans and computers for

which the language tools developed that can comprehend communication languages across all technological dimensions need to be considered. To ensure that words are spelt correctly, the spell-checker consults the language's dictionary and lexicon. An easy way to understand a spellchecker is that it uses a spelling detector to look for words that are not in base form in a document and a spelling corrector to replace them with the most likely term from a database or corpus.

Preprocessing, feature extraction, and modeling are the three primary steps that make up the Natural Language Processing (NLP) pipeline as shown in Figure 1 [14]. Every step of the process changes the text in some manner and generates a result that is needed by the following step. Sometimes, there are non-linear steps in the NLP pipeline. Going back and forth between the various stages is often essential in practice. For instance, if the modeling stage yields unsatisfactory results, it might be required to revisit the pre-processing or feature extraction stage in order to enhance the data's quality.

There are primarily three stages to spellchecking which include thorough pre-processing, spelling check, and creation of recommendation lists. Some of the steps involved in preprocessing include stemming, tokenization, and normalization [15] [16]. The spelling-checking module uses several dictionary lookup techniques to verify the candidate words' authenticity, while the recommendation list building module flags the list of possible suggestions for misspelled words. Suggestions are ranked by the spell-checker. This part ranks the ideas according to how necessary they are for the sentences. The primary stages of a spellchecker are as shown in Figure 2. After running the sentence through spellcheck, it returns a corrected version containing the correct term.

2.2.1 Syntax based

Each sentence obtains a parse tree created according to the base language's grammar. Text is inaccurate if full parsing fails. So, the parser should be as thorough as possible to reduce false alerts. The main advantage of this method is that the grammar checker will detect all errors if the grammar is complete and covers all possible syntactic rules.

2.2.2 Rule based

When checking for spelling errors, these systems use heuristics derived from various word properties, including as morphology, part-of-speech, stemming, and more [4]. A rule-based spell-checker using part-of-speech (POS) tagging for English language spell-checking is also being used. Additionally, text chunkers were created using the Hidden Markov Model to improve spell-checker speed [4]. This technique allows for progressive system expansion by starting with one rule and adding more [17].

2.2.3 Statistical based

The ability to speak a specific language is not necessary to understand statistical procedures. Examples of spellcheckers that use word counts and word characteristics include those that are frequency-based, n-gram-based, and finite state automata-based [4]. The statistical method greatly enhances performance without requiring knowledge of the particular language, which is a major advantage. One problem with these approaches is that they rely on metrics like word count, frequency, and characteristics to do spellchecking, yet processing certain spelling mistakes necessitates familiarity with the target language. Many academics employed a combination of rule-based and statistical approaches to address this type of problem. To get past the problems, a hybrid model combines rule-based and statistical approaches [4] [18].

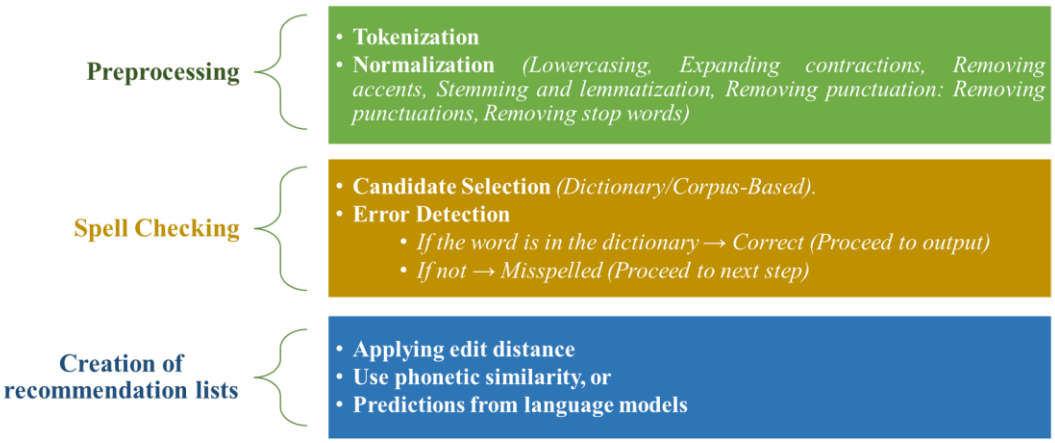


Figure 2: The primary stages of Spelling checker using NLP

Table 1: Comparison of various spell checker and grammar checker for low resource languages

Refere	Year	Language	Research Focus	Methodology/Approach	Key Findings & Accuracy
[24]	2002	Assamese	Dictionary-based spellchecker	Dictionary lookup, bigram search, Soundex code integration	Adequate results with over 5000 words, integration with Assamese-English dictionary in progress
[26]	2012	Kashmiri	Spellchecker development	Standalone application, non-word error correction	80% error detection, 85% correct suggestions, plans for real-word error handling
[25]	2013	Urdu	Spellchecker evaluation	Reverse edit distance method	High complexity (86n + 41 comparisons), needs improved methods for better accuracy
[27]	2015	Hindi	HINSPELL spellchecker	Error detection, repair, substitution	83.2% detection, 77.9% correction, future focus on grammatical errors
[28]	2015	Tamil	Morphological analyzer	Linguistic analysis, POS tagging	Efficiency between 60-97%, useful for NLP tasks like MT, lemmatization, parsing
[29]	2016	Kashmiri	Improved spellchecker	Standalone application, lexicon development	80% detection, 85% correct recommendations, integration with OpenOffice needed
[30]	2016	Tamil	Hybrid spellchecker	N-gram, stemming, tree-based algorithm	91% accuracy, tree-based method better for error detection
[31]	2016	Telugu	Spellchecker with sandhi analysis	Morphophonemic, external sandhi handling	Addresses Telugu's complex linguistic features
[19]	2018	Malayalam	Deep learning-based spellchecker	LSTM neural networks, error detection & correction	Outperforms Unicode splitting, limited by computational resources
[32]	2019	Bengali	Spellchecker development	Hybrid of edit distance, Soundex	Adapts existing methods for better Bengali spell correction
[33]	2020	Multilingual	Comprehensive spellchecker review	Literature analysis, NLP methods (rule-based, statistical, deep learning)	Categorizes spellcheckers, compares performance across languages
[25]	2020	Tamil	Alternative spellchecking methods	Bloom-filter, Symspell, LSTM	Symspell is fast but lacks accuracy; LSTM is promising but underexplored
[3]	2021	Gujarati	Jodani spellchecker	Root word-based, Levenshtein distance	91.56% accuracy, plans to improve character assumption handling
[23]	2022	Dogri	Hybrid spellchecker	Hybrid methodology for detection & correction	First known attempt for Dogri spellchecking
[34]	2023	Gujarati	Enhancing ASR Performance with Improved Spell Corrector	Combination of MFCC and CQCC features, GRU-based DeepSpeech2 architecture, and enhanced spell corrector	Improved Word Error Rate by 17.46% compared to the model without post-processing
[11]	2024	Gujarati	Spell Checker Using Norvig Algorithm	Implementation of Norvig's algorithm with a dataset of 16,937 distinct Gujarati words	Achieved 80–90% accuracy in identifying and correcting misspelled words

Table 2: Comparison of spell correction models designed for multilingual and low-resource languages

Model	**Language Coverage**	**Architecture**	**Error Handling (Morph., Contextual)**	**Computational Efficiency**	**Novelty/Remarks**
IndicBERT [35]	12 Indic + English	ALBERT (Transformer-based)	Limited analysis; strong recall	Efficient (ALBERT backbone)	First shared Indic ALBERT model
MuRIL (Google) [36]	17+ Indian Languages	Multilingual BERT variant	Better contextual handling (cross-lingual pretraining)	Moderate to high	Strong cross-lingual transfer
XLM-R [37]	100+ languages	RoBERTa-based	Strong contextual handling	High resource requirements	Robust multilingual performance
Adapter-BERT for Indic [38]	Varies	BERT + Adapters	Task-specific error modeling possible	High efficiency (modular)	Scalable low-resource adaptation
L3Cube-IndicSBERT [39]	10 Indic Languages	Multilingual SBERT	Enhanced sentence-level contextual embeddings	Efficient fine-tuning	First multilingual sentence representation model for Indic languages
ArabicCorrectionCntxt [40]	Arabic	Levenshtein + Context Probabilities	Handles contextual errors via paragraph-level keyword-based context detection	Efficient (simple lexical + context)	Uses paragraph context and keyword frequency to re-rank corrections
Icelandic Contextual Spell Corrector [41]	Icelandic	ML Classifiers + Morphological Tags	Strong contextual disambiguation; affected by rich morphology	Moderate (due to tag sparsity)	Contextual confusion-set disambiguation with lemmatized and PoS features
Context-Free ML Spell Corrector [42]	English (demonstrated); extendable	Supervised ML (e.g., Naive Bayes)	No context used; character/word/token-based input features	Efficient for standalone terms	Context-free, character-level input with multiple ML classifiers
Amazon Multilingual Spell Checker [43]	24 Languages (Indic included)	N-gram-based + SymSpell Ranking	Context-aware using n-gram conditional probabilities	Real-time capable (optimized Trie)	Real-time, extendable to new languages via Wikipedia + subtitle corpora

2.2.4 Deep learning based

Deep learning is the specialty of artificial neural networks, or ML algorithms. Deep learning algorithms have been widely employed and effective lately. Deep learning approaches' success is partly due to the freedom of architecture selection. Deep learning methods were used in ML research for natural language processing [17]. While the rule-based and statistical methods demonstrate significant effectiveness, the performance of spell-checking can be further improved through the application of deep-learning techniques. Regarding regional languages, the deep-learning-based spell-checker is currently available for the Malayalam and Tamil language which utilizes an LSTM network [19] [20]. This spell-checker involves a network that is both trained and tested to detect spelling errors and pinpoint their locations [4] [21] [22] [44] [45] [46] [47].

3 Comparative analysis of Spell checker for various regional languages of India

Natural language processing (NLP) depends much on spell and grammar checkers since they find and fix textual data mistakes. Although a lot of study has been done on English and other generally spoken languages, regional languages of India provide special difficulties because of their rich morphology, complicated phonetic structures, and different scripts. The spell-checking methods created for several Indian languages—including

Hindi, Bengali, Tamil, Telugu, Gujarati, Dogri, Malyalam and Assamese—are compared in this work [23] [33]. Examining several approaches including rule-based, statistical, hybrid, and deep learning-based spell-checkers, the paper assesses their performance in managing orthographic variants, phonetic mistakes, and real-word errors. Particularly languages with strong inflectional morphology, like Tamil and Telugu, call for more complex methods like sandhi-based and morphological analyzers; languages like Hindi and Bengali gain from hybrid approaches combining edit distance and phonetic algorithms [6] [21] [24] [25]. Emphasizing the importance of language-specific optimizations, the study shows the benefits and restrictions of every technique. Future lines of research include using transformer-based models such IndicBERT and GRU to better contextual spell-checking, hence improving accuracy over several languages. Through tackling these difficulties, our work hopes to help to create more strong and effective spell-checking systems for India's linguistically varied terrain. Table 1 shows the differences between spell checkers for languages with few resources. Table 2 looks at spell correction models that work in more than one language. These models, such as IndicBERT, MuRIL, and Amazon's spell checker, use contextual and probabilistic methodologies. Others deal with linguistic and morphological issues.

Previously studies examined spellchecking in Indic languages by dictionary lookup [24], root-based matching [3], hybrid linguistic methodologies [30], and deep learning [19]; nevertheless, these systems exhibit notable deficiencies. Rule-based systems like Jodani for Gujarati or HINSPELL for Hindi do a good job of finding errors, but they don't take context into account, so they can't tell the difference between real-word errors (such homophones or words that look similar). Hybrid approaches created for Tamil and Dogri enhance identification via stemming or POS-based criteria; nonetheless, they are still language-specific and do not adapt well to the orthographic complexity of Gujarati (diacritics, compound characters). Neural methods, like LSTM-based rectification for Malayalam or GRU-based ASR augmentation for Gujarati [34], show gains in context, but they are constrained by datasets that are particular to a certain field, a lack of generalization, and high computing needs.

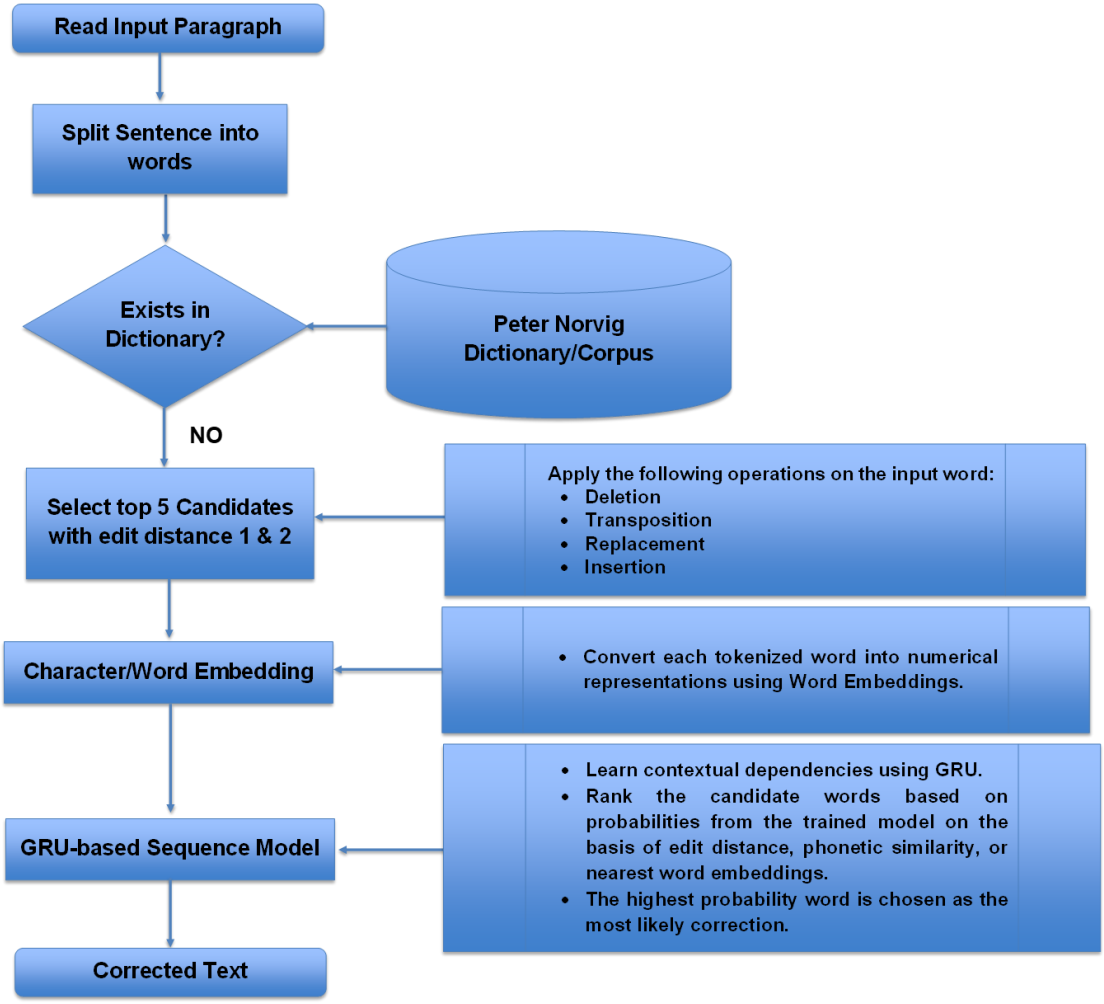


Figure 3: Novel and hybrid Spelling and Grammar Error Correction approaches for Gujarati Language using Peter Norvig with GRU – GUJAPUBRIJ Model

Multilingual transformer models (IndicBERT [35], MuRIL [36], XLM-R [37]) provide strong contextual embeddings, but they have two problems when it comes to correcting Gujarati spelling: (i) they are pre-trained on mixed Indic corpora without domain adaptation, which makes them bad at Gujarati morphology, and (ii) they need too many resources to be useful for lightweight deployment. These gaps highlight the need for a context-aware, Gujarati-specialized spellchecking approach that balances linguistic coverage with computational efficiency.

4 Proposed GUJAPUBRIJ and GUJBRIJAPU Models

The proposed Gujarati spell checker models are founded on Peter Norvig's algorithm, which applies probabilistic methods and edit-distance operations for error correction using a Gujarati lexicon. Although effective in detecting typographical errors within a predefined dictionary, this approach is limited by its lack of contextual awareness. To overcome this limitation, two enhanced models were developed: the GUJAPUBRIJ model, incorporating a GRU-based neural network, and the GUJBRIJAPU model, leveraging IndicBERT for contextual analysis. For ease of reference and novel identification, the models were named after the contributing researchers, Apurva and Brijehkumar. This approach represents a significant advancement in context-aware spell checking for Gujarati and contributes to strengthening NLP resources for low-resource Indic languages by delivering improved accuracy, reliability, and applicability.

4.1 Novel and hybrid spelling and grammar error correction approaches for Gujarati language using Peter Norvig with GRU – GUJAPUBRIJ model

The GUJAPUBRIJ model combines Peter Norvig's probabilistic spell correction with a GRU-based neural network to make it more accurate and aware of the context. A carefully chosen Gujarati vocabulary was used as the reference dictionary to make sure that any adjustments made through insertions, deletions, substitutions, and transpositions were correct from a language point of view.

TensorFlow's Tokenizer split the input text into tokens (using <UNK> for OOV words) and added

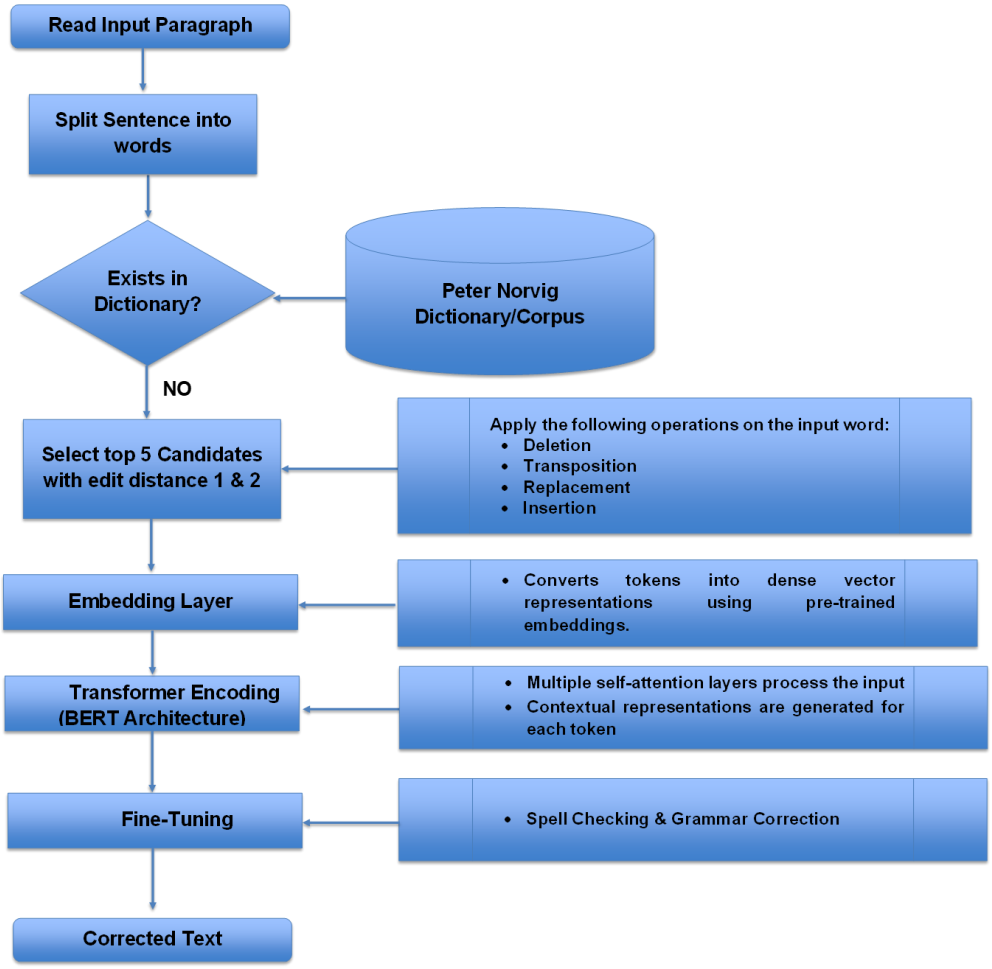


Figure 4: Novel and hybrid Spelling and Grammar Error Correction approaches for Gujarati Language using Peter Norvig with IndicBERT- GUJBRIJAPU Model

padding to make them all the same size. The GRU network then took these sequences and used them to choose the most appropriate correction from the top five choices that were within edit distance 1–2. This preparation pipeline made sure that both the spelling and the meaning were correct.

The proposed GUJAPUBRIJ novel model for the Gujarati text spelling checker based on Peter Norvig and GRU are depicted in Figure 3.

Peter Norvig's approach fixes mistakes by utilizing edit distance and probability. This works well for typos within words, but it needs a defined lexicon and doesn't know what the context is. On the other hand, GRU-based neural networks work with sequential language input, taking into account how words are related to each other in context to improve grammar checks and deal with real-world mistakes that Norvig's method can't fix.

4.2 Novel and hybrid spelling and grammar error correction approaches for Gujarati Language using Peter Norvig with IndicBERT – GUJBRIJAPU model

The GUJBRIJAPU model builds a Gujarati spell checker by moving from dictionary-based methods to a BERT-based framework that can rank phrases based on how well they fit in with the context as shown in Figure 4. The preprocessing pipeline uses lexical normalization and semantic validation. It starts with a cleaned Gujarati corpus to make the reference vocabulary.

The input text is broken up into tokens, and for each token, Peter Norvig's probabilistic method creates possible repairs by using edit-distance operations (insertions, deletions, replacements, transpositions) that have been changed for Gujarati. Candidates that follow spelling guidelines are ranked, and the top five (with an edit distance of 1–2) are chosen for further testing.

To add context, candidate-corrected sentences were run using IndicBERT, a multilingual model that has already been trained on Indian languages, to make contextual embeddings. A scoring system (masked language modeling or sentence-level probability) selected choices based on how well they fit semantically. The sentence with the best score was chosen as the correction. This two-step technique worked well for both real and non-word errors by combining lexical correction with contextual relevance. Tokens were turned into dense embeddings and improved by IndicBERT's self-attention layers, which made the grammar and spelling more accurate overall. The mMML model based on BERT was used to make the spell checker more accurate and faster. The Sentence Scoring Model (which checks the validity of sentences) was chosen for Gujarati because it does a good job of ranking candidate sentences by accuracy.

5 Experimental evaluation and Analysis of the proposed model

To develop a Gujarati language dataset for the study, data was sourced from publicly available resources provided by the Ekatra Foundation, accessible via https://www.ekatrafoundation.org/. Additionally, a curated Google Drive folder containing extensive Gujarati textual data was utilized (https://drive.google.com/drive/folders/17gskNhAGgzOpncOh2VsAKC4Fc0ju5GaC?usp=sharing). Over 100,000 sentences were initially collected from these sources. The preprocessing of the dataset includes converting to lowercase Unicode NFC, managing compound characters and diacritics specifically for the Gujarati language (ZWJ/ZWNJ), removing duplicates, stripping punctuation, and selecting sentences based on length. Peter Norvig's edit-distance approach was modified for the Gujarati script and diacritics, and it was used for candidate generation. For training, the Hugging Face IndicBERTv2-SS AutoTokenizer could handle sentences up to 128 tokens in length, and for inference, it could handle sentences up to 512 tokens in length, therefore it could handle lengthier candidate sentences. After performing a thorough data cleaning and preprocessing process to ensure quality and relevance, a final dataset comprising 20,000 sentences was created. This refined dataset includes both correct and erroneous sentence pairs, making it suitable for tasks such as spell error correction and language model training. A dataset of 20,000 sentences has been divided into one of two groups:

1. Sentences that are grammatically and orthographically correct.
2. Incorrect sentences that have common spelling and grammar mistakes that are common in Gujarati.

To make supervised learning easier, sentences were given probability-based labels: 0.9 for right and 0.1 for wrong. This rating helped the model learn how to tell the difference between valid and invalid text using regression. To make sure that the evaluation was strong, the dataset was split into 16,816 training samples and 4,204 validation samples. The GUJAPUBRIJ (Peter Norvig + GRU) and GUJBRIJAPU (Peter Norvig + IndicBERT) models were trained using the dataset that had been built up. The training process was mostly about regression-based fine-tuning, which meant that the model gave each potential sentence a likelihood score. Then, a rating system was used:

1. Predicted scores were used to rank the candidate sentences.
2. The sentence with the greatest probability score was chosen as the one that was most likely to be correct.

This improvement made the GUJBRIJAPU model better at correcting context by letting it tell the difference between small spelling problems and big grammar issues. The GUJAPUBRIJ model converged quickly, getting about 100% training accuracy in just two epochs. The validation accuracy also stabilized close to 100%. This meant that the model not only learned the training patterns by heart, but it also worked well with new validation data. This indicates that the model is not Over Fitting; instead, it is showing that it can generalize well on the validation set.

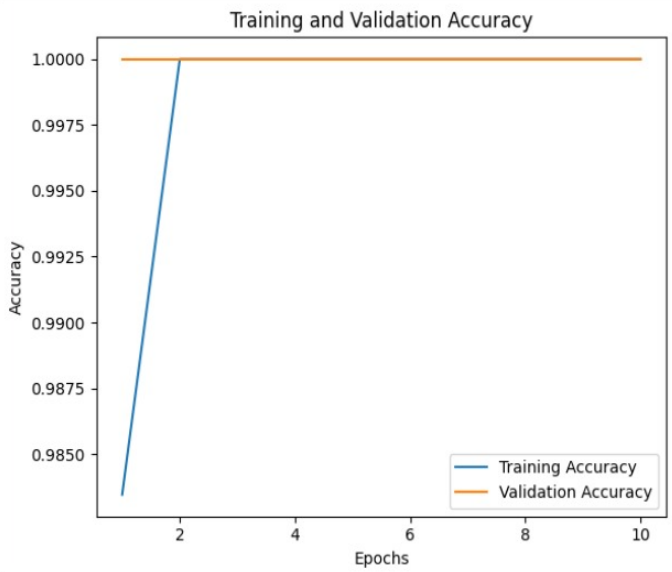


Figure 5: Training and Testing accuracy for Novel and hybrid Spelling Error Correction approaches for Gujarati Language using Peter Norvig with GRU – GUJAPUBRIJ Model

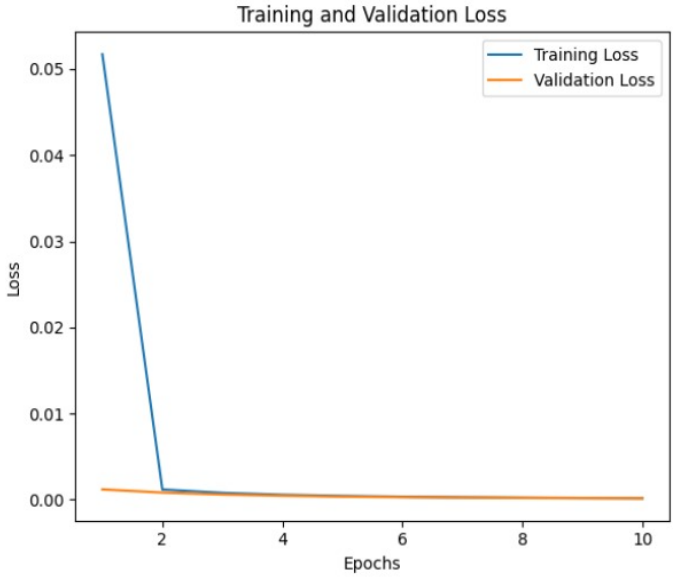


Figure 6: Training and validation loss for Novel and hybrid spelling error correction approaches for Gujarati Language using Peter Norvig with GRU- GUJBRIJAPU Model

Table 3: Performance analysis of Novel and hybrid spelling correction approaches for Gujarati language GUJAPUBRIJ and GUJBRIJAPU Model

Evaluation Parameters	**Peter Norvig with GRU – GUJAPUBRIJ Model**		**Peter Norvig with IndicBERT – GUJBRIJAPU Model**	
	Incorrect sentences (in %)	**Correct Sentences (in %)**	**Incorrect sentences (in %)**	**Correct Sentences (in %)**
Accuracy	71.00	85.00	84.79	93.49
Precision	71.00	84.00	86.21	94.46
Recall	71.00	85.00	83.54	90.13
F1 Score	70.00	84.00	85.74	91.59

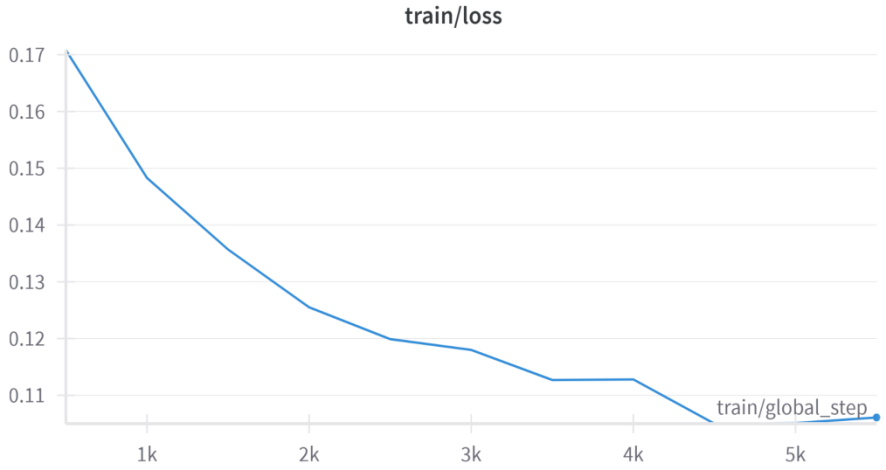


Figure 7: train/loss plot for Novel and hybrid Spelling Error Correction approaches for Gujarati Language using Peter Norvig with IndicBERT – GUJBRIJAPU Model

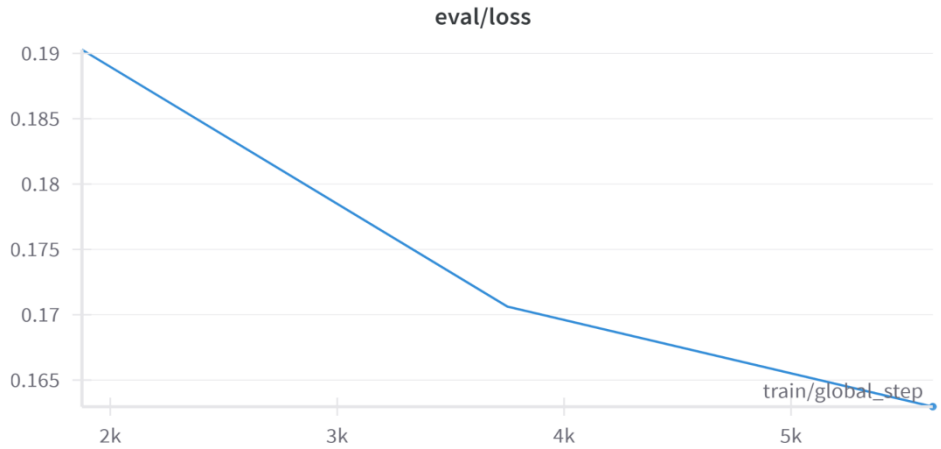


Figure 8: eval/loss plot for Novel and hybrid Spelling Error Correction approaches for Gujarati Language using Peter Norvig with IndicBERT – GUJBRIJAPU Model

But the validation accuracy being so high could also be due to the dataset itself, like its small size or the fact that the training and validation samples are quite comparable. This is something to keep in mind when looking at these results. The Table 3 shows accuracy statistics were achieved for correct and incorrect sentences for both the proposed models.

The Figure 5 shows the graph for training and validation accuracy while Figure 6 shows the graph for training and validation loss.

Training epochs and results for the GUJBRIJAPU Model based on Peter Norvig with IndicBERT are plotted as shown in Figure 7 and Figure 8.

For the GRU-based GUJAPUBRIJ model, we used Random Search with Keras Tuner to find the best hyperparameters in the following space: embedding dimensions {64, 96, 128, 160, 192, 224, 256}, GRU hidden units {32, 64, 96, 128}, dropout {0.1–0.5}, and learning rates {1e-3, 1e-4, 1e-5}. The batch size was fixed at 32 and the maximum number of trials was 10. The best setup, which was determined based on the validation F1-score, was: Embedding dimension = 128 (adding more dimensions made the model fit too well without improving performance), GRU hidden units = 64 (enough power without making training take longer or increasing the danger of overfitting), Dropout = 0.3 (kept training stable while keeping it from underfitting at lower rates), Learning rate = 1e-3 (this made the process go faster than 1e-4 or 1e-5), with 10 epochs and 32 batches.

This system found a good balance between being complicated and being able to apply to a lot of different situations. To reduce the amount of processing power needed, each configuration was only performed once (executions_per_trial=1). We recognize that random initialization creates variation; in subsequent research,

Table 4: Hyperparameter tuning for the Novel and hybrid Spelling Correction approach for Gujarati Language using Peter Norvig with GRU- GUJAPUBRIJ Model

Hyperparameter	**Range / Value**	**Description**
embedding_dim	64 to 256 (step=32)	Size of embedding vectors, tuned using Keras Tuner
gru_units	32 to 128 (step=32)	Number of units in GRU layer, tuned using Keras Tuner
input_length	max_seq_length from data	Length of padded sequences
Optimizer	Adam	Optimization algorithm used for training
Loss	Binary Crossentropy	Loss function for binary classification
Metrics	Accuracy	Evaluation metric during training and validation
tuning method	Random Search	Keras Tuner with 10 trials and validation accuracy goal

Table 5: Hyperparameter tuning for the Novel and hybrid Spelling Correction approach for Gujarati Language using Peter Norvig with IndicBERT – GUJBRIJAPU Model

Parameter	**Value**	**Description**
Model	ai4bharat/IndicBERTv2-SS	Pretrained Indic language transformer model
Number of Labels	1	Binary classification setup
Tokenizer	AutoTokenizer (from model)	Handles subword tokenization using the same model
Optimizer	Adam	Adaptive Moment Estimation (default in Trainer)
Learning Rate	2e-5	Fine-tuning rate for BERT parameters
Batch Size	16	Per-device mini-batch size
Number of Epochs	3	Number of full passes over training data
Weight Decay	0.01	L2 regularization strength
Evaluation Strategy	Epoch	Evaluate after every epoch
Loss Function	Binary Cross Entropy (via Trainer)	Used for sentence-level scoring
Compute Metrics	Custom scoring (e.g., accuracy or ranking)	Contextual ranking of candidates

multiple seeds will be employed for reliable significance reporting.

The AI4Bharat/IndicBERTv2-SS model served as the basis for the GUJBRIJAPU model, which was then fine-tuned on the 20,000-sentence Gujarati dataset we talked about earlier. To help with regression-based training, each sentence was given a probability score (0.1 for wrong, 0.9 for right). We used AdamW to improve the model. The learning rate was 2e-5, the batch size was 16, there were three epochs, the weight decay was 0.01, and the binary classification loss was changed to work with regression-style output. This made sure that the model could score potential sentences based on how well they fit in with the context. We used general-purpose IndicBERT, but we note that using bigger Gujarati-only corpora for domain adaptation could boost rectification performance even more and is something we will explore on in the future. The Table 4 and Table 5 shows the hyperparameters details for both the proposed model.

The entire study was conducted on Google Colab with an NVIDIA Tesla T4 GPU (16GB). The software environment included Python 3.10, PyTorch 2.0.1+cu118, Hugging Face Transformers 4.30.2, Datasets 2.13.0, and CUDA 11.8, as well as Weights & Biases (wandb), pandas, and numpy. We used the Hugging Face Trainer API with AdamW optimization, a weight decay of 0.01, and a binary classification loss that was changed to work with regression-style outputs for training. You can use PyTorch, NumPy, and Python's random module to set random seeds so that your results are the same every time you run the program. You can also turn on deterministic cuDNN mode.

6 Discussion and comparative evaluation of the proposed GUJAPUBRIJ and GUJBRIJAPU model

Two context-aware Gujarati spell-checking models: the GRU-based GUJAPUBRIJ (blue solid line with triangles) and the IndicBERT-based GUJBRIJAPU (green dashed line with squares) are compared in the Figure 9 and Figure 10 graph on the basis of Accuracy, precision, recall and F1 score with the metric values displayed on the y-axis (percentage scale) and the metric types arranged on the x-axis for incorrect sentences and correct sentences respectively.

The comparative assessment demonstrates that the IndicBERT-based GUJBRIJAPU Model substantially outperforms the GRU-based GUJAPUBRIJ Model across all principal performance metrics as shown in Table 7.

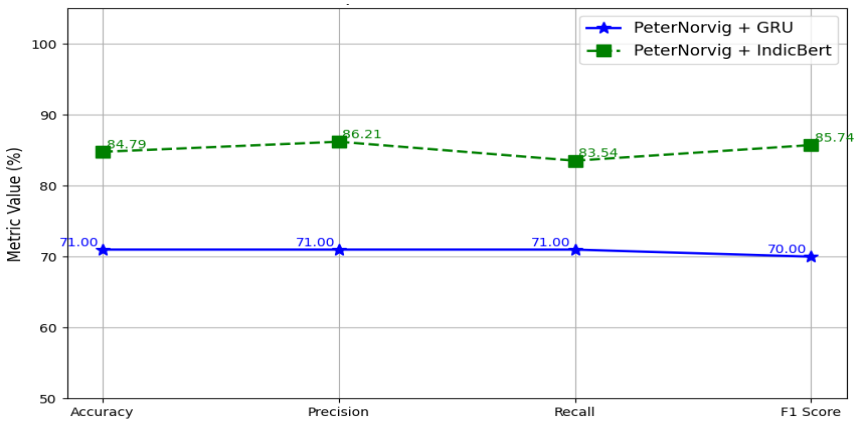


Figure 9 Comparative analysis of both the proposed model on incorrect sentences

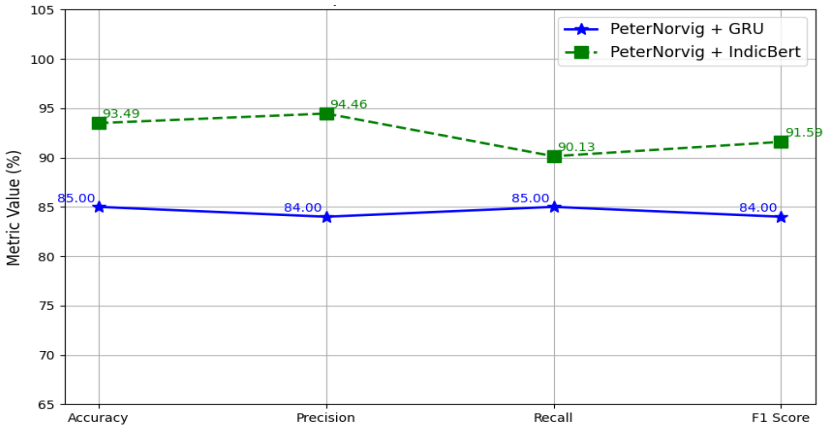


Figure 10: Comparative analysis of both the proposed models for correct sentences

Table 6: Comparative analysis of proposed GUJAPUBRIJ/GUJBRIJAPU model with Jodani [3]

Feature	**Peter Norvig with GRU / IndicBERT– GUJAPUBRIJ/GUJBRIJAPU**	**Jodani [3]**
Spelling Correction	Deep- learning based approach	Rule based
Grammar Checking	Yes	No
Contextual Understanding	Yes	No
Efficiency & Speed	More Computation power	Faster and lightweight
Scalability & Learning	Can be fine-tuned	Predefined rules are used.
Use case	NLP applications (Chatbots, Translation, Gujarati Grammar Checking)	Word Processing, Typo Correction

Table 7: Performance analysis of proposed model with Jodani [3]

Metric	**Peter Norvig with GRU-GUJAPUBRIJ**	**Peter Norvig with IndicBERT-GUJBRIJAPU**	**Jodani [3]**
Accuracy	85.00%	93.49%	91.56%
Precision	84.00%	94.46%	85.00%
Recall	85.00%	90.13%	100.00%
F1 Score	84.00%	91.59%	91.00%

The GUJBRIJAPU model is more reliable when it comes to handling both real-world and non-word errors since it has higher accuracy (93.49% vs. 85.00%), precision (94.46% vs. 84.00%), recall (90.13% vs. 85.00%), and F1-score (91.59% vs. 84.00%). It is important to note that it lowers the number of false positives and misclassifications of wrong sentences, which leads to more context-aware and linguistically correct repairs. These results show that GUJBRIJAPU is a strong Gujarati spell and grammar checker, which sets the stage for further improvements and changes for specific fields.

The Novel Peter Norvig and GRU/IndicBERT based GUJAPUBRIJ/GUJBRIJAPU Model context aware spelling checker for Gujarati Text is compared with Jodani which a spellchecker tool for Gujarati that finds mistakes and fixes them while also suggesting ways to be more accurate. Jodani boosts accuracy by combining rule-based and statistical methods and uses a predetermined Gujarati vocabulary to identify misspelled words using edit distance and phonetic comparableness. It deals with orthographic mistakes such as missing characters, transposition, addition, and substitution using phonetic similarities, N-gram models, and Peter Norvig's technique for correcting spelling to enhance precision.

The IndicBERT-based GUJBRIJAPU model outperforms both the GRU-based GUJAPUBRIJ and the rule-based Jodani on all important measures. Its higher accuracy (93.49%) and precision (94.46%) show that it as fewer false positives and more reliable corrections. Its higher recall (90.13%) shows that it is better at finding real-world and non-word errors. GUJBRIJAPU is better than Jodani because it can check grammar and grasp context, while Jodani can only fix spelling mistakes using edit distance and phonetic similarity. It can find and fix grammar-sensitive mistakes by using contextual embeddings, which makes it better for NLP tasks like grammar checking, translation, and conversational systems. The key trade-off is the cost of computing: GUJBRIJAPU is slower and uses more resources than Jodani, which is still light and useful for speedier programs like word processors. Some unusual and code-mixed terms were misclassified, which means that more training and domain adaptation are needed. This technique is new since it combines IndicBERT for understanding context, Peter Norvig's algorithm for generating candidates, and neural sequence modeling for grammar sensitivity. This integration goes beyond small changes and offers a unified, context-aware spelling and grammar checker for Gujarati. This is better than current systems that only fix spelling based on rules. Table 6 compares the proposed models with Jodani [3], The researcher’s proposed work is based on deep learning and includes a comparison with a rule-based approach. Currently, there is no existing spell checker developed at an individual level for direct comparison. Therefore, to clearly demonstrate the novelty of the research, a comparative analysis with the older rule-based system has been included. Although this comparison may not be technically perfect, it helps in better understanding the improvement and effectiveness of the proposed model. From this comparison, researchers demonstrate the novelty and effectiveness of their proposed work.

7 Conclusion

Spell and grammar checkers are vital to natural language processing (NLP) because they detect and fix mistakes in textual input. Although a lot of study has been done on English and other generally spoken languages, the regional languages of India provide special difficulties because of their complicated morphology, sophisticated phonetic patterns, and different scripts. The study carried out in this article emphasizes how difficult context aware spell checking in Indian regional languages especially Gujarati is given their intricate linguistic systems. The novel and hybrid error detection and correction approach based on Peter Norvig's spelling correction algorithm in collaboration with GRU (GUJAPUBRIJ Model) neural networks and IndicBERT (GUJBRIJAPU Model) are proposed and assessed. With outstanding accuracy, precision, recall, and F1-score, IndicBERT regularly exceeded GRU among the evaluated models. For all important criteria, Peter Norvig's method combined with IndicBERT means GUJBRIJAPU Model showed the best performance;
Thus, it is the most dependable method for context aware grammar correction in Gujarati literature. Jodani, rule-based spell checker for Gujarati, suffers with grammatical precision and lacks contextual knowledge even if it offers a quick rule-based fix for spelling errors. On the other hand, the Peter Norvig with IndicBERT GUJBRIJAPU Model is perfect for uses needing excellent contextual understanding since it uses deep learning to precisely identify spelling and grammatical mistakes. Still, Jodani is a good choice in situations requiring speedier processing with reduced computing burden. With improved accuracy and contextual awareness, the Peter Norvig with IndicBERT GUJBRIJAPU Model eventually seems to be the most solid approach for Gujarati language processing. Its capacity to lower false positives, increase classification accuracy, and offer thorough error correction makes it a useful tool for many NLP uses where linguistic accuracy is crucial. The computation overload can be reduced in future to improve the performance of the Peter Norving with IndicBERT GUJBRIJAPU model. Moreover, the proposed models are unable to identify the error related to punctuations which can be improved in future.

References

[1] N. G. Patel and D. D. B. Patel, "Research review of Rule Based Gujarati Grammar Implementation with the Concepts of Natural Language Processing (NLP)," Journal of Emerging Technologies and Innovative Research (JETIR), vol. 5, no. 9, 2018. https://doi.org/10.6084/m9.jetir.JETIRA006276

[2] N. P. Desai and V. K. Dabhi, "Resources and components for Gujarati NLP systems: a survey.," Artificial Intelligence Review, vol. 55, pp. 1-19, 2022. https://doi.org/10.1007/s10462-021-10120-1

[3] H. Patel, B. Patel and K. Lad, "Jodani: A spell checking and suggesting tool for Gujarati language," in 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2021. https://doi.org/10.1109/Confluence51648.2021.9377072

[4] S. Singh and S. Singh., "HINDIA: a deep-learning-based model for spell-checking of Hindi language," Neural Computing and Applications, vol. 33, no. 8, pp. 3825-3840, 2021. https://doi.org/10.1007/s00521-020-05207-9

[5] M. Gokani and R. Mamidi, "GSAC: A Gujarati Sentiment Analysis Corpus from Twitter," in Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, Association for Computational Linguistics, 2023. https://doi.org/10.18653/v1/2023.wassa-1.12

[6] S. Bhuva and D. Mishra, "Gujarati Optical Character Recognition Using Efficient Text Feature Extraction Approaches.," Informatica, vol. 49, no. 28, 2025. https://doi.org/10.31449/inf.v49i28.8341

[7] J. Baxi and B. Bhatt., "GujMORPH-ADatasetforCreatingGujaratiMorphological Analyzer," in ProceedingsoftheThirteenthLanguageResourcesandEvaluationConference, 2022. https://aclanthology.org/2022.lrec-1.767/

[8] A. Desai, "Gujarati handwritten numeral optical character reorganization through neural network.," Pattern recognition, vol. 43, no. 7, pp. 2582-2589, 2010. https://doi.org/10.1016/j.patcog.2010.01.008

[9] S. Antani and L. Agnihotri, "Gujarati character recognition," in Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR '99, Bangalore, India, 1999. https://doi.org/10.1109/ICDAR.1999.791813

[10] Tailor, C., Patel, B."Chunker for Gujarati Language Using Hybrid Approach," in Rising Threats in Expert Applications and Solutions. Advances in Intelligent Systems and Computing, 2021. https://doi.org/10.1007/978-981-15-6014-9_10

[11] K. Suba, D. Jiandani and P. Bhattacharyya, "Hybrid inflectional stemmer and rule-based derivational stemmer for gujarati.," in Proceedings of the 2nd workshop on south southeast Asian natural language processing (WSSANLP), 2011. https://aclanthology.org/W11-3001

[12] B. K. Y. Panchal and A. Shah, "Spell Checker Using Norvig Algorithm for Gujarati Language," in nternational Conference on Smart Data Intelligence. Singapore, Singapore, 2024. https://doi.org/10.1007/978-981-97-3191-6_21

[13] N. Patel and D. Patel, "Implementation Approach of Indian Language Gujarati Grammar's Concept “sandhi” using the Concepts of Rule-based NLP," in 8th International Conference on Computing for Sustainable Global Development (INDIACom)., 2021. https://doi.org/10.1109/INDIACom51348.2021.00085.

[14] J. Sheth and B. C. Patel., "Gujarati phonetics and Levenshtein based string similarity measure for Gujarati language.," in 5th National Conference on Indian Language Computing., 2015. https://www.researchgate.net/publication/314153559

[15] T. A. Gal. "Natural Language Processing (NLP) Pipeline." Medium, 23 Oct 2023. [Online]. Available: https://medium.com/@theaveragegal/natural-language-processing-nlp-pipeline-e766d832a1e5

[16] P. Patel, K. Popat and P. Bhattacharyya, "Hybrid stemmer for Gujarati," in Proceedings of the 1st Workshop on South and Southeast Asian Natural Language Processing, 2010. https://aclanthology.org/W10-3607

[17] M. Parikh and A. Desai, "Recognition of Handwritten Gujarati Conjuncts Using the Convolutional Neural Network Architectures: AlexNet, GoogLeNet, Inception V3, and ResNet50," in Advances in Computing and Data Sciences: 6th International Conference, ICACDS2022, Kurnool, India, 2022. https://doi.org/10.1007/978-3-031-12641-3_24.

[18] B. K. Y. Panchal and A. Shah, "NLP-Based Spellchecker and Grammar Checker for Indic Languages.," in Natural Language Processing for Software Engineering, Scrivener Publishing LLC, 2025, pp. 43-70. https://doi.org/10.1002/9781394272464.ch4

[19] C. Tailor and B. Patel, "Sentence Tokenization Using Statistical Unsupervised Machine LearningandRule-BasedApproachforRunningTextinGujaratiLanguage, " in Emerging Trends in Expert Applications andSecurity.AdvancesinIntelligent SystemsandComputing, 2018. https://doi.org/10.1007/978-981-13-2285-3_38

[20] S. Sooraj, K. Manjusha, M. A. Kumar and K. P. Soman, "Deep learning-based spell checker for Malayalam language," Journal of Intelligent & Fuzzy Systems, vol. 34, no. 3, pp. 1427-1434, 2018. https://doi.org/10.3233/JIFS-169438

[21] S. Murugan, T. A. Bakthavatchalam and M. Sankarasubbu, "Symspell and lstm based spell-checkers for tamil," in Tamil Internet Conference, 2020. https://www.researchgate.net/publication/3499249

[22] N. Hossain, M. H. Bijoy, S. Islam and S. Shatabda, "Panini: a transformer-based grammatical error correction method for Bangla," Neural Computing and Applications, vol. 36, pp. 3463-3477, 2024. https://doi.org/10.1007/s00521-023-09211-7

[23] R. Phukan, M. Neog and N. Baruah, "A Deep Learning Based Approach For Spelling Error Detection In The Assamese Language," in 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, 2023. https://doi.org/10.1109/ICCCNT56998.2023.10306972

[24] S. S. Jamwal and P. Gupta., "A Novel Hybrid Approach for the Designing and Implementation of Dogri Spell Checker," in Data, Engineering and Applications: Select Proceedings of IDEA 2021, Singapore, 2022. https://doi.org/10.1007/978-981-19-4687-5_53

[25] S. Singh and S. Singh, "Systematic review of spell-checkers for highly inflectional languages," Artificial Intelligence Review, vol. 53, no. 6, pp. 4051-4092, 2020. https://doi.org/10.1007/s10462-019-09787-4

[26] M. Das, S. Borgohain, J. Gogoi and S. Nair, "Design and implementation of a spell checker for Assamese," in Language Engineering Conference, 2002. Proceedings, 2002. https://doi.org/10.1109/LEC.2002.1182303

[27] S. Iqbal, W. Anwar, U. I. Bajwa and Z. Rehman., "Urdu spell checking: Reverse edit distance approach," in In Proceedings of the 4th workshop on south and southeast asian natural language processing, 2013. https://aclanthology.org/W13-4707

[28] R. Sakuntharaj and S. Mahesan, "A novel hybrid approach to detect and correct spelling in Tamil text," in 2016 IEEE international conference on information and automation for sustainability (ICIAfS), 2016. https://doi.org/10.1109/ICIAFS.2016.7946522

[29] B. Bhagat and M. Dua, "Enhancing performance of end-to-end gujarati language asr using combination of integrated feature extraction and improved spell corrector algorithm," in ITM Web of Conferences, 2023. https://doi.org/10.1051/itmconf/20235401016

[30] D. Kakwani, A. Kunchukuttan, S. Golla, G. NC, A. Bhattacharyya, M. M. Khapra and P. Kumar., "IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages," In Findings of the association for computational linguistics: EMNLP 2020, pp. 4948-4961, 2020. https://doi.org/10.18653/v1/2020.findings-emnlp.445

[31] S. Khanuja, D. Bansal, S. Mehtani, S. Khosla, A. Dey, B. Gopalan, D. Margam, P. Aggarwal, R. Nagipogu, S. Dave and S. Gupta, "Muril: Multilingual representations for indian languages.," arXiv preprint arXiv:2103, p. 10730, 2021. https://doi.org/10.48550/arXiv.2103.10730

[32] Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer and V. Stoyanov., "Unsupervised cross-lingual representation learning at scale," arXiv preprint arXiv:1911.02116, 2019.

[33] A. Lawaye and B. S. Purkayastha, "KASHMIRI SPELL CHECKER AND SUGGESTION SYSTEM," THE COMMUNICATIONS, vol. 21, no. 2, p. 123, 2012. https://ddeku.edu.in/Files/2cfa4584-5afe-43ce-aa4b-ad936cc9d3be/Journal/6bb36225-ee44-4d4c-9d3d-0905436082e8.pdf

[34] Kaur and H. Singh, "Design and implementation of HINSPELL—Hindi spell checker using hybrid approach," International Journal of scientific research and management, vol. 3, no. 2, pp. 2058-2062, 2015. https://ijsrm.net/index.php/ijsrm/article/view/102

[35] R. Sankaravelayuthan, "Spell and grammar checker for Tamil.," Developing computing tools for Tamil, vol. 5, no. 23, pp. 52-64, 2015. https://doi.org/10.13140/RG.2.1.3700.6803

[36] A. Lawaye and B. S. Purkayastha, "Design and implementation of spell checker for Kashmiri," International Journal of Scientific Research, vol. 5, no. 7, 2016. https://www.researchgate.net/publication/321906322

[37] U. M. G. Rao, A. P. Kulkarni and a. P. K. Christopher Mala, "Telugu Spell-Checker," Vaagartha, 2012. https://sanskrit.uohyd.ac.in/faculty/amba/PUBLICATIONS/papers/ITIC-ss.pdf

[38] S. Saha, F. Tabassum, K. Saha, Akter. and Marjana, "Bangla Spell Checker and Suggestion Generator," (Dissertation, United International University), 2019. https://www.academia.edu/96829901/

[39] J. A. R. C. P. Pfeiffer, A. Kamath, I. Vulić, S. Ruder, K. Cho and I. Gurevych, "Adapterhub: A framework for adapting transformers," arXiv preprint arXiv:2007.07779, 2020. https://doi.org/10.48550/arXiv.2007.07779

[40] S. Deode, J. Gadre, A. Kajale, A. Joshi and R. Joshi, "L3Cube-IndicSBERT: A simple approach for learning cross-lingual sentence representations using multilingual BERT."," arXiv preprint arXiv:2304.11434, 2023. https://doi.org/10.48550/arXiv.2304.11434

[41] M. Nejja and A. Yousfi., "The context in automatic spell correction," Procedia Computer Science, vol. 73, pp. 109-114, 2015. https://doi.org/10.1016/j.procs.2015.12.055

[42] K. Ingason, S. B. Jóhannsson, E. Rögnvaldsson, H. Loftsson and S. Helgadóttir., "Context-sensitive spelling correction and rich morphology.," in Proceedings of the 17th Nordic Conference of Computational Linguistics (NODALIDA 2009), 2009. https://aclanthology.org/W09-4634.pdf

[43] Yunus and M. Masum., "A context free spell correction method using supervised machine learning algorithms," International Journal of Computer Applications, vol. 176, no. 27, pp. 36-41, 2020. https://doi.org/10.5120/ijca2020920288

[44] P. Gupta, "A context-sensitive real-time Spell Checker with language adaptability," in 2020 IEEE 14th International Conference on Semantic Computing (ICSC), 2020. https://doi.org/10.48550/arXiv.1910.11242

[45] Priya, M.C.S., Renuka, D.K., Kumar, L.A. et al. Multilingual low resource Indian language speech recognition and spell correction using Indic BERT. Sādhanā 47, 227 (2022). https://doi.org/10.1007/s12046-022-01973-5

[46] Parida, S. et al. (2022). BertOdia: BERT Pre-training for Low Resource Odia Language. In: Dehuri, S., Prasad Mishra, B.S., Mallick, P.K., Cho, SB. (eds) Biologically Inspired Techniques in Many Criteria Decision Making. Smart Innovation, Systems and Technologies, vol 271. Springer, Singapore. https://doi.org/10.1007/978-981-16-8739-6_32

[47] Dashti, S.M.S., Khatibi Bardsiri, A. & Jafari Shahbazzadeh, M. PERCORE: A Deep Learning-Based Framework for Persian Spelling Correction with Phonetic Analysis. Int J Comput Intell Syst 17, 114 (2024). https://doi.org/10.1007/s44196-024-00459-y