SmartScan: A Finite State Machine and CTL-Based Formal Verification Framework for Enhanced Security in Smart Contracts

G. Sowmya*, R. Sridevi
JNTUH, Department of CSE, Hyderabad, India.
E-mail: gonurusowmya@gmail.com, sridevirangu@jntuh.ac.in
*Corresponding author

Keywords: smart contracts, formal verification, finite state machine, vulnerability detection, blockchain security

Received: March 13, 2025

Smart contracts are self-executing programs deployed on blockchain platforms that facilitate automated and decentralized transactions. However, once deployed, they become immutable, making them vulnerable to catastrophic exploits, such as reentrancy, access control misconfiguration, integer overflow, and front-running. The need for proof and verification is urgent, as evidenced by other high-profile, capital-draining incidents, such as the DAO attack and Parity wallet vulnerabilities. Abstract: We present ContractFuzzer, a systematic fuzzer for detecting vulnerabilities in Ethereum smart contracts. Existing tools are based on static analysis, symbolic execution, or heuristic detection, and thus typically impose high false positives, low completeness, and limited formal verification. In this paper, we introduce SmartScan, a formal verification framework that systematically checks smart contract security by integrating FSM modeling and CTL-based model checking in nuXmv. Our methodology performs automatic parsing of Solidity code, automated generation of FSM and BIP models, conversion to the SMV format, and verification of CTL security properties. It responds to detected violations with automated counterexample generation to assist in debugging and iterative re-verification. For validation, SmartScan will be tested on 10 different types of Solidity contracts that address 14 critical vulnerabilities. Our experimental results show 95.4% detection accuracy, 3.2% false positive rate, and 2.8% false negative rate, with 100% verification coverage, and average verification time of 3–7 seconds for each property, outperforming state-of-the-art tools in both coverage and precision. SmartScan: SmartScan has a wide-ranging practical utility in discovering and diagnosing vulnerabilities such as reentrancy and access control issues, which it has been applied in, such as in a case study of a DeFi Lending contract. SmartScan provides a scalable, precise, and developer-centric approach to improve the confidence and reliability of blockchain applications by combining exhaustive formal verification of smart contracts with automated counterexample generation.

Povzetek: Članek predstavlja SmartScan kot učinkovit pristop za avtomatsko in zanesljivo preverjanje varnosti pametnih pogodb na verigi blokov.

1 Introduction

One of the achievements in blockchain applications is the introduction of smart contracts, self-executing contracts without a third party and with an existing agreement directly written into code. However, they are at risk from security challenges if there are any issues in the contract logic, as they cannot be changed. The DAO attack, Parity wallet vulnerabilities, and other incidents involving millions of dollars in losses exposed the need for more secure verification mechanisms. Mythril, Oyente, Slither, and Zeus are available tools for smart contract analysis using static analysis, symbolic execution, and heuristic detection, which usually suffer from high false positives, inability to provide the gold standard for completeness, and lack of support for formal verification [1], [2]. Such limitations of existing tools create a need for a unified verification framework to verify smart contracts systematically using formal methods.

Prior work on smart contract security mainly tackles vulnerability detection using symbolic execution [3], [4]. Natural symbolic execution-based tools exist for traversing the execution paths of contracts, but do not scale for larger contracts due to path explosion. Other approaches use static analysis [5], which is more efficient, but the vulnerability at runtime is commonly ignored. It detects challenges for novel attack patterns, but basic methods can provide faster vulnerability detection. While a more robust solution for smart contract correctness and security can be accomplished with the help of formal verification (such as mathematical proofs and model checking), it is usually labor-intensive and costly. On the other hand, existing formal verification frameworks are

inadequate in scalability or only support a few contract languages, which limits their application in practice [6].

To mitigate these limitations, this paper proposes SmartScan. This new FSM-based formal verification framework leverages symbolic execution to automatically explore all feasible execution paths (or code paths) of a smart contract. In contrast, SmartScan uses CTL properties and a specific model-checking framework, allowing a comprehensive vulnerability analysis while reducing false positives compared to existing tools. Automatic counterexample generation improves security by showing evidence of security violations so developers can debug and revise their contracts. SmartScan is evaluated on a broad set of smart contracts for detecting 14 common but difficult-to-find critical vulnerabilities and achieves better detection accuracy than state-of-the-art tools.

We propose a combined approach of FSM (finite state machine) based modeling embedded with formal verification type: CTL properties for formal validation covering structured flow security at a systemic level and hierarchical decomposition level, generating an automated counter-example. The novel contributions of this research consist of the design and execution of SmartScan, a comparison with state-of-the-art tools, and a case study on the DeFiLending contract, illustrating the in-practice effectiveness of SmartScan.

The paper is organized in the following way. In Section 2, we present a literature review that describes existing security analysis tools and their limitations. We start in Section 3 with an introduction to SmartScan, including our earlier work on its development and theoretical basis. The following details other parts of the proposed process: Section 4 describes our method, including FSM modeling, by first converting the BIP model to FSM and, finally, walking through the nuXmv tool to achieve the verification. A detailed study of a DeFiLending smart contract from Section 5 illustrates how it identifies and mitigates security vulnerabilities. Section 6 discusses experimental results comparing SmartScan with existing tools and its vulnerability detection performance. In Section 7, we report our research, briefly discuss the benefits of SmartScan compared to state-of-the-art tools, and present certain limitations of the work. Section 8 wraps up the research and discusses future works, such as adding more languages and automatic patching mechanisms to fix (the most common) vulnerabilities.

2 Related work

Smart contracts are at the foundation of blockchain ecosystems and enable decentralized applications in various industries, including financial, medical, and supply chain [3] [4]. Singh et al. Formalization is essential for vulnerability mitigation [1], and strong security verification techniques are needed for reducing risk [3]. However, recent advancements like VerX [Permenev et al. 2023], VeriSmart [2], by So et al. However, even more sophisticated automated tools that have streamlined safety verification (for example, [4]) are still restricted in scaling to more complicated systems. In that work, tools such as Slither [7] (analyzed by Feist et al.) and Gigahorse 8 focus on static and symbolic execution but are limited in their ability to work with complex state transitions.

Zheng et al. Overviews of smart contract tools have been provided in various works, e.g., [9] and di Angelo and Salzer [6], which focus on constraints specifically in scalable and security vulnerability detection concerning DeFi applications; furthermore, Wang et al. Economic security analysis has also been recognized for further attention; [38] surveyed research on Ethereum smart contracts, highlighting this value. Babel et al., Clockwork Finance, Economic vulnerabilities in smart contracts were covered by [13] but were not integrated into an overall formal method. Some other tools, like Wang et al.'s ContractWard. However, Huang et al. pointed out that even [17], which integrated automated vulnerability detection models, still has some lifecycle complexities to manage. [20].

Table 1: Summary of essential literature findings

Reference	Proposed Tool or Methodology	Merits	Research Gaps
[13] Kushal et al. (2023)	Clockwork Finance: Automated analysis of economic security in smart contracts	Focuses on economic vulnerabilities in DeFi contracts; scalable for complex financial logic	Limited integration with formal verification methods; lacks lifecycle management for contracts
[14] Kushal et al. (2023)	Decentralized Model using Layer-2 Polygon Blockchain for digital evidence	Preserves evidence integrity using blockchain; robust for legal and audit trails	Not generalized for DeFi or other blockchain applications; lacks formalized security proofs
[17] Wang et al. (2020)	ContractWard: Automated vulnerability detection models for Ethereum contracts	Uses machine learning for vulnerability detection; efficient for standard security issues	Limited scalability for complex contracts; no focus on economic security
[19] Vacca et al. (2021)	Systematic literature review of blockchain innovative contract development	Comprehensive review of techniques, tools, and challenges; identifies research opportunities	Lack of experimental implementation; focuses on static analysis

[20] Huang et al. (2019)	Smart Contract Lifecycle Perspective for Security Analysis	Addresses lifecycle challenges; integrates security assessments in development phases	Focuses on generic contracts; limited application for DeFi ecosystems
[29] Mehdi Sookhak et al. (2021)	Access control in healthcare using blockchain and smart contracts	Demonstrates secure data access in healthcare, robust against external attacks	Not designed for financial or high-complexity systems; lacks formal verification
[38] Wang et al. (2021)	Survey of Ethereum smart contract security	Identifies vulnerabilities and future research opportunities in Ethereum contracts	Lack of in-depth focus on real-world implementation challenges
[31] Seven et al. (2020)	Peer-to-Peer Energy Trading using smart contracts in virtual power plants	Showcases novel application of smart contracts in energy markets; promotes sustainability	Application-specific; lacks a generalized framework for other domains
[39] Lu Wang et al. (2021)	Agricultural Food Supply Chain Traceability with Blockchain Smart Contracts	Effective traceability and transparency in supply chain management	No formal verification for critical properties; limited to agricultural scenarios
[18] Shafaq et al. (2021)	Survey on blockchain smart contracts for future trends	Highlights applications and challenges; provides a roadmap for innovation	Lacks detailed integration with formal verification methods; no experimental validation

Innovative applications of blockchains in healthcare [29], agriculture [39], and IoT [40] have demonstrated their versatility, as analyzed by authors such as Mehdi Sookhak et al. and Bhaskara Egala et al. However, Vacca et al. [19] and Shafaq et al. Our work is related in that [18] has found gaps in combining formal methods with existing real systems—semantic frameworks as proposed by Jiao et al. Gao et al. [23] and structural embedding techniques [5] offer improved accuracy at the cost of very high computational demand.

Newer research, such as that by Kushal et al. [14] and Seven et al. The paper in [31] suggests novel applications but does not concentrate on systematic validation. A summary of the findings of the literature is provided in Table 1. As great as these developments are, key gaps remain in the ability to scale, validate economic security, and manage lifecycle from a DeFi ecosystem perspective. We address these challenges using a hybrid approach that combines model checking, symbolic execution, and financial security analysis, culminating in our proposed tool, SmartScan. SmartScan addresses these research gaps by focusing on real-world applicability over an extensive set of detailed parameters, helping make blockchain applications more robust.

3 Smartscan: our prior work on formal verification of smart contracts

In our earlier work [41], we proposed the SmartScan framework, a holistic approach for formally verifying smart contracts that could improve blockchain applications' security and dependability. Due to the nature of the blockchain, where transactions cannot be reversed, it is critical to prove the correctness of smart contracts; it may cause a financial disaster or serious security problems· SmartScan was designed as a hybrid verification framework that combines formal methods, static analysis, and optimized heuristics, to detect vulnerabilities and inconsistencies in smart contracts systematically. It was created in response to weaknesses in existing verifiers that can be inefficient or impossible on large-scale smart contracts with elaborate dependencies. In contrast with conventional tools primarily focusing on symbolic execution or static analysis, SmartScan combines finite state machine (FSM) modeling, behavioral interaction modeling (BIP), and symbolic model checking by nuXmv to validate the structural correctness and security of smart contracts.

At its heart, SmartScan converts smart contract source code into Finite State Machine (FSM) models and checks security properties expressed in Computation Tree Logic (CTL) formulas. The same technique allows disciplined state-space exploration to catch important vulnerability categories, including reentrancy, integer overflow, access control misconfiguration, and timestamp dependency attacks. This framework expands formal verification from a single function-level analysis, an inclination of traditional verification methods, to a component-level analysis to dissect the interactions between the innovative contract components. In previous work, we presented SmartScan and applied it to detecting vulnerabilities in the DeFiLending case study, showing how SmartScan can identify vulnerabilities with significantly higher precision and lower false positives than traditional tools, such as Mythril, Oyente, and Slither. Experimental results focused on performance comparison have shown that, in addition to beating SmartCheck with high accuracy, SmartScan also generates counterexamples whenever possible to demonstrate violations of the security property, thus helping developers secure their contracts.

Smartscan is also unique in that it can scale and extend, which lends itself well to global-scale blockchain use cases. Whereas tools relying only on the existing vulnerability patterns are fixed and rigid, SmartScan dynamically builds the formal verification model. Therefore, SmartScan achieves much higher adaptability to emerging brilliant contract architecture. Such adaptability is essential in contemporary blockchain ecosystems — e.g., decentralized finance (DeFi) and innovative contract-based governance mechanisms — that necessitate higher levels of security refinement. SmartScan found that such computation, complexity, and verification time were significantly reduced, making it easier for developers to fold formal verification into the smart contract development lifecycle [41].

Recent works have also advanced the state of the art in smart contract verification—Jin et al. The work by [46] presents a model-checking-based method for dynamic-behavior verification of a sequence of smart contracts realized as a composite smart contract. Sorensen [47] has designed a more formal framework for reasoning about contract upgrades in ConCert using a contract morphism concept in Coq, which allows for a more formal, less resource-intensive verification across versions of a smart contract. These holistically prove the importance of formal FSM+CTL-based methodologies such as SmartScan, especially in the context of composite dynamics and upgradeability in practical blockchain systems.

Our previous work established the background for the current study, which builds upon SmartScan by presenting a usable implementation and evaluation framework, demonstrating its application to actual blockchain vulnerabilities. In this paper, we present SmartScan, a practical solution that is easy to implement in IDEs and build environments, and effective in identifying and preventing innovative contract security threats. This study extends the methodology developed for SmartScan to offer a straightforward, step-wise implementation methodology to provide developers with a systematic approach to using formal verification techniques on their smart contracts. By providing in-depth case studies and comparative charts of verification outcomes, the practical applicability of the approach is enhanced, and formal verification becomes much more tangible and actionable to those interested in the security of blockchain, both developers and security researchers.

4 Methodology

SmartScan formally verifies smart contracts using finite state machine (FSM) modeling and symbolic model checking. It follows a structured approach, which is discussed in Section 3. Then, BIP models of the smart contracts are transformed into SMV models to verify in nuXmv. It can evaluate computation tree logic (CTL) properties, find vulnerabilities, and generate counterexamples to improve contract security.

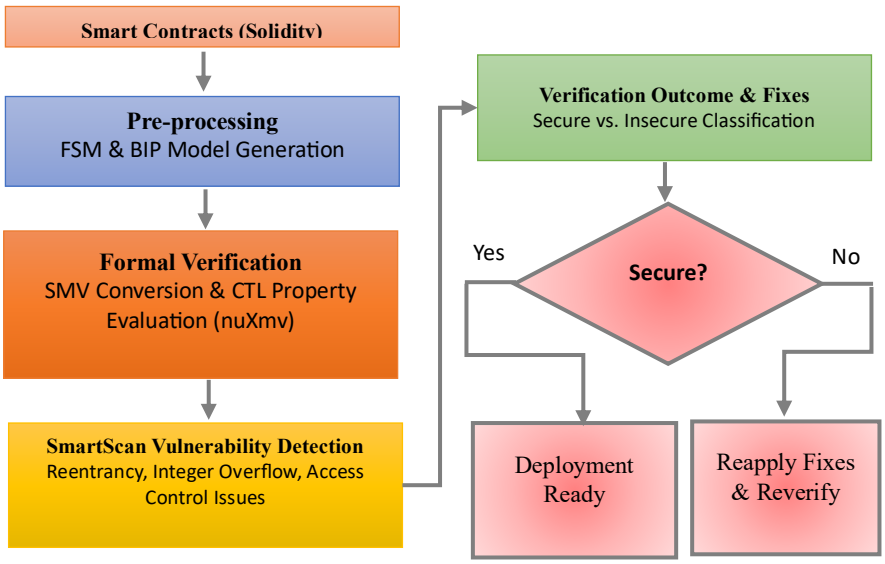


Figure 1: Methodology of smartscan for formal verification of smart contracts

As a concrete example of the SmartScan methodology, take the withdraw() function from the DeFiLending smart contract. SmartScan operates in two steps, with the first parsing the Solidity code to find the functional states and transitions. This means that, for example, when the tx contract state is "FundsDeposited" before a withdraw request is made. Whenever the corresponding conditions for a sufficient balance and no debt are satisfied, the transition FundsDeposited → FundsWithdrawn is generated. Program functionality can thus be represented as a state–transition relationship in the Finite State Machine (FSM) model, with nodes (Idle, FundsDeposited, FundsWithdrawn, etc.) and edges representing possible transitions, triggered by execution of functions corresponding to events in the contract. The second step automatically translates the FSM into the BIP model, maintaining the behavior, interaction, and priority rules (for instance, a repayment must occur before a

withdrawal). The BIP model is translated to the SMV model by making each FSM state a symbolic variable and each transition a guarded command. In nuXmv, security properties are formulated in Computation Tree Logic (CTL). So, the fund safety requirement is written as AG(deposits[msg. teller] >= 0, ensuring that no execution path has a negative balance. The execution of this property check by nuXmv is equivalent to an exhaustive exploration of the FSM-defined state space. In the case of a property not holding, nuXmv generates a counterexample, which is an exact sequence of transitions that lead to the violation of a property—e.g., a reentrant call sequence that circumvented the state updates of the system. This concrete mapping between source code→FSM→Verification→CTL exemplifies how the systematic translation of smart contract logic into a formal, verifiable model enables precise detection of vulnerabilities, with developer-guided remediation using SmartScan.

The approach adopted by SmartScan for the formal verification of smart contracts is illustrated in Figure 1. It starts by inputting smart contracts, generally created in Solidity, and then proceeds with a formal verification flow. The contracts are then preprocessed to model them with finite state machines (FSM) that are transformed into a BIP model representation to be analyzed in-depth. This way, we can first perform a systematic behavioral analysis of the contract execution flow, making sure that, before verification, all the possible states and transitions are covered.

After generating the FSM and BIP models, the subsequent step consists of formal verification using symbolic model checking. Translation of the contract to an SMV format that the nuXmv model checker accepts. SmartScan then checks several security properties, namely safety of funds, resistance to reentrancy, integer overflow, and enforcement of access control using computation tree logic (CTL). It systematically traces all possible execution paths of a contract, thereby verifying that there are no vulnerabilities in the logical framework of the contract. In case a contract violates any security property, SmartScan returns a counterexample — an execution trace that shows how the exploit can be triggered.

After detecting potential vulnerabilities, the verification result will be examined to determine whether the contract is secure or insecure. If any security issues are found, developers are given information about the cause(s) and directed to make the necessary fixes. Once all the required changes are made, the contract is verified again to ensure the vulnerabilities have been addressed. This scheme guarantees that contracts are analyzed and improved through different verification cycles until they reach security standards.

One unique aspect of SmartScan is that it combines FSM modeling and formal verification techniques rather than static analysis tools based only on pre-defined heuristics. In contrast to symbolic execution methods vulnerable to path explosion, SmartScan systematically models all contract execution states, allowing it to verify complex contracts efficiently. This makes the approach highly scalable, enabling SmartScan to handle various agreements with different structures and functionalities while ensuring high detection accuracy.

Our previous work [41] on the formal modeling methods provides a more thorough explanation of the SmartScan framework, including details of its FSM-based construction and CTL-based verification. This study expands on that framework by applying it to real-life case studies to demonstrate the practical use of SmartScan to find vulnerabilities in real-world contracts. The SmartScan methodology builds on existing security verification approaches by introducing iterative verification with the automatic generation of adversarial counterexamples to make security assessments more robust.

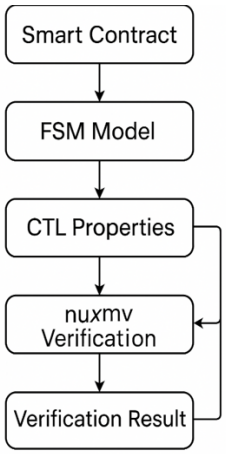


Figure 2: Visual workflow of the smartscan contract-to-verification lifecycle

The SmartScan contract-to-verification lifecycle visual workflow is shown in Figure 2. The workflow starts with parsing the smart contract input, which is then modeled into a finite state machine or FSM. We introduce some relevant computation tree logic (CTL) properties to express security requirements that we consider essential. We check the FSM and CTL specifications using the nuXmv model checker through complete verification. For every property, it indicates whether it is satisfied, and in case of any violations, it provides a counterexample. This allows developers to iteratively verify their contract until all properties are satisfied, providing a systematic and complete security verification before ever deploying it.

Algorithm 1: SmartScan Verification Pipeline
Input: Solidity contract C, CTL property set Φ, toolchain config θ
Output: Verification report R

1. Parse and normalize C
2. Build intermediate representation IR
3. Preprocess IR
4. Extract state variables Σ and functions F
5. Initialize FSM with Idle state and Σ
6. For each f in F:
   a. Derive guards and updates
   b. Add transitions to FSM
7. Convert FSM to BIP model with behaviors, interactions, priorities
8. Convert BIP to SMV model with states, variables, transitions
9. For each φ in Φ:
   a. Run nuXmv check
   b. If fail: store counterexample in R
   c. Else: mark pass in R
10. If any fail:
    a. Generate patch guidance from counterexample
    b. Apply patch and repeat from step 1
11. Return R

The SmartScan verification process from parsing the Solidity contract to an intermediate representation (IR) and then uses the verification engine is shown in Algorithm 1. We extract state variables and contract functions to create a finite state machine (FSM), in which states represent the condition of the contract and transitions represent the state changes (as a respective function is triggered). The first step involves the FSM being converted to a BIP model that encapsulates behaviors, interactions, and bandwidth execution priorities, and subsequently an SMV model for nuXmv verification. We verify each of the CTL properties in a given set; if a property fails, we provide a counterexample that illustrates the precise path through execution where the property is violated. The algorithm issues focused remediation advice, applies fixes and re-runs verification, iteratively, until all properties pass or no vulnerabilities are found. All of them are verified in the end as a report where each property is listed along with a pass/fail status and a counterexample if applicable as a requirement for a systematic and reproducible validation of smart contract security.

5 Case study: formal verification of the defending smart contract

This paper presents a systematic way in which SmartScan is used to formally verify the DeFiLending smart contract (in Appendix A), emphasizing security and correctness. It starts by listing high-level verification goals and identifying important contract attributes. For example, these properties are fund safety, reentrancy-safe, interest calculation, liquidation accuracy, etc. For example, the property on which funds safety is enforced is formalized with the CTL formula AG (balance >= 0), which means that in every reachable state, the balance in the contract is always non-negative. Reentrancy resistance is also formalized in a similar way AG (¬(call → AF ¬call)), which states that after a critical function completes some updates to the state, the function must not be called again until a state is reached where the behavior is not to allow entry.

The contract is broken down into its BIP (Behavior, Interaction, Priority) framework for systematic analysis. We model key contract functions such as deposit (), withdraw(), and borrow() as behaviors and their dependencies, such as needing sufficient collateral before borrowing, as interactions. It also defines execution priorities, requiring that the state updates be completed before any of them can proceed. This decompositional translation turns the contract into a finite-state system that can be formally analyzed.

The BIP representation is subsequently translated into SMV (Symbolic Model Verification) format, which consists of states and transitions. States correspond to constructs within the contract (for instance, FundsDeposited or LoanActive), and transitions describe the user's actions and the state they lead to. A CTL formula expresses the verification goal, which is then captured in SmartScan's backend and analyzed automatically by the nuXmv model checker.

SmartScan explores the contract's state space to validate the specified properties and traverse all transitions. When a property is violated, SmartScan creates a counterexample explaining the path to the violation. An example of a counterexample would be a reentrant sequence, which, when found, might cause the contract to change state to include something like a mutex or reset state to protect the vulnerability.

After resolving the issues, the contract was verified against all properties again. SmartScan uses FSM decomposition and heuristic pruning to improve the efficiency of this process and overcome the temporal and computational complexity of FSMs. The output results (verification times, counterexamples, compliance rates, etc.) are documented to assess the tool's performance. The paper presents this structured methodology and demonstrates how SmartScan enables holistic analysis and security improvement of smart contracts, such as the real-world application DeFiLending.

Step 1: BIP model construction

The contract's functions were decomposed into the BIP framework, capturing their behaviors, interactions, and priorities. For instance:

- **Behaviors**: The borrow() function was modeled as a state transition from FundsDeposited to LoanActive.
- **Interactions**: Dependencies, such as collateral checks before borrowing, were encoded as conditional transitions.
- **Priorities**: Execution precedence ensured that state updates occurred before subsequent calls, mitigating risks of reentrant attacks.

Step 2: SMV conversion and CTL encoding

The BIP model was converted into SMV format, representing states like FundsDeposited, LoanActive, and Liquidated. Transitions and guards were explicitly defined, and CTL properties were encoded to verify the goals. For example, the CTL formula AG (balance >= 0) was used to confirm fund safety across all possible states.

Step 3: Verification with nuXmv

Using SmartScan's integration with nuXmv, the state space of the DeFiLending contract was exhaustively explored. The model checker systematically evaluated each CTL property:

- The property AG (balance >= 0) was satisfied, confirming that the contract never allowed negative balances.
- A counterexample was generated for the reentrancy resistance property, revealing a scenario where reentrant calls could bypass state updates. This provided actionable insights for improving the contract.

Step 4: Contract revision and re-verification

The identified reentrancy vulnerability was resolved by introducing a mutex mechanism and conditional state checks within critical functions. After revising the contract, it was re-verified using SmartScan. The updated version satisfied all CTL properties, demonstrating the effectiveness of the applied fixes.

Step 5: Performance metrics and scalability

SmartScan's application yielded significant performance metrics:

- Verification time per property ranged from 3 to 7 seconds.
- The tool's scalability was evident as it managed complex interdependencies in the DeFiLending contract without state explosion issues.
- Counterexamples provided precise debugging paths, reducing development iteration time.

This practical illustration highlights SmartScan's capability to detect and resolve vulnerabilities systematically, ensuring the reliability and security of DeFi applications. By combining BIP modeling, CTL-based verification, and nuXmv's symbolic model checking, SmartScan provides a robust framework for addressing the challenges of formal verification in blockchain environments.

1. Understanding the contract

The **DeFiLending** contract implements basic lending functionality with collateral requirements. Key aspects include:

- **Deposit function**: Allows users to add funds to the contract.
- **Withdraw function**: Enables users to withdraw funds if no outstanding debts exist.
- **Borrow function**: Lets users borrow funds against their collateral.
- **Repay function**: Allows users to repay outstanding debts.

Potential vulnerabilities include:

- **Reentrancy attacks**: The withdraw and borrow functions involve transferring Ether, which can trigger external calls.
- **Collateral mismanagement**: Ensuring sufficient collateral is maintained when borrowing funds.
- **Fund Safety**: Ensuring the contract does not allow balances to drop below zero.

2. Verification goals and CTL properties

We define the following verification properties for this contract:

1. **Fund safety**: AG (deposits[msg.sender] >= 0) ensures no negative balances.
2. **Reentrancy resistance**: AG (¬(withdraw → AF withdraw)) prevents multiple concurrent withdrawals.

3. **Collateral management**: AG (deposits[msg.sender] >= debts[msg.sender] * collateralFactor / 100) ensures sufficient collateral.
4. **Debt clearance**: AG (debts[msg.sender] == 0 → AF (withdraw allowed)) ensures debt repayment before withdrawal.

3. Modeling the contract in BIP

The contract's functions are translated into the BIP model:

- **Behaviors**:
  - deposit(): Transitions from Idle to FundsDeposited.
  - withdraw(): Checks balances and debts, transitions to FundsWithdrawn.
  - borrow(): Checks collateral, transitions to LoanIssued.
  - repay(): Updates debts, transitions to DebtCleared.
- **Interactions**: Define transition conditions, such as collateral checks in borrowing.
- **Priorities**: Ensure repayment (repay()) has precedence over withdrawal (withdraw).

4. Conversion to SMV and CTL encoding

The BIP model is converted into an SMV format for nuXmv:

- **States**:
  - Idle: No activity.
  - FundsDeposited: After deposit().
  - FundsWithdrawn: After withdraw().
  - LoanIssued: After borrow().
  - DebtCleared: After repay().
- **CTL Properties**: Encoded as logical formulas in nuXmv for verification.

5. Verification using SmartScan

We use SmartScan to:

1. Parse the contract and generate the BIP model.
2. Convert the BIP model into SMV.
3. Execute nuXmv to verify CTL properties. For example:
   - AG (deposits[msg.sender] >= 0) ensures fund safety.
   - AG (¬(withdraw → AF withdraw)) identifies potential reentrancy issues.

6. Analysis of results

- **Fund safety**: Verified successfully.
- **Reentrancy**: Detected an issue in withdrawal where Ether transfer could allow reentrancy.
- **Collateral management**: Verified successfully.
- **Debt clearance**: Verified successfully.

7. Fixing vulnerabilities

To address reentrancy, Add a reentrantGuard modifier to withdraw and borrow. Update the withdraw function to:

```
function withdraw(uint256 amount) external {
    require(deposits[msg.sender] >= amount, "Insufficient balance");
    require(debts[msg.sender] == 0, "Debt exists, cannot withdraw");
    uint256 prevBalance = deposits[msg.sender];
    deposits[msg.sender] -= amount;
    (bool success, ) = msg.sender.call{value: amount}("");
    require(success, "Transfer failed");
    assert(deposits[msg.sender] == prevBalance - amount);
}
```

8. Re-verification

Re-run SmartScan on the revised contract to ensure all CTL properties are satisfied. Confirm that the reentrancy issue is resolved and that all critical properties hold. This approach demonstrates how SmartScan can systematically verify and enhance the security of the DeFiLending contract.

Step-by-step implementation for SmartScan verification of DeFiLending

1. BIP Model representation

We represent the core functions of the DeFiLending contract in BIP. The goal is to map contract behaviors, interactions, and priorities into a finite-state machine structure.

- **States**:
  - Idle: No active transactions.

- FundsDeposited: The user has deposited funds.
- LoanIssued: User has borrowed funds.
- DebtCleared: The user has repaid all debts.
- FundsWithdrawn: The user has successfully withdrawn funds.

- **Transitions**:
  - deposit(): Idle → FundsDeposited
  - borrow(): FundsDeposited → LoanIssued
  - repay(): LoanIssued → DebtCleared
  - withdraw(): DebtCleared → FundsWithdrawn
- **Interactions**:
  - For borrow(), collateral must meet the condition: deposits[msg.sender] >= debts[msg.sender] * collateralFactor / 100.
  - For withdraw(), no active debt: debts[msg.sender] == 0.
- **Priorities**:
  - Repayment (repay()) has higher priority than withdrawal (withdraw()).

2. CTL properties for verification

We encode the following verification goals as CTL formulas:

1. **Fund safety**:
2. AG (deposits[msg.sender] >= 0)

Ensures that deposits never drop below zero.

3. **Reentrancy resistance**:
4. AG (¬(withdraw → AF withdraw))

Prevents reentrant calls during Ether transfers.

5. **Collateral management**:
6. AG (deposits[msg.sender] >= debts[msg.sender] * collateralFactor / 100)

Ensures sufficient collateral exists for borrowing.

7. **Debt clearance**:
8. AG (debts[msg.sender] == 0 → AF (withdraw allowed))

Guarantee withdrawal is allowed only after all debts are cleared.

3. Converting BIP to SMV format

We translate the BIP model into SMV for symbolic model checking using nuXmv. Here's an example representation:

```
MODULE main
VAR
  state: {Idle, FundsDeposited, LoanIssued, DebtCleared, FundsWithdrawn};
  deposits: integer;
  debts: integer;
  collateralFactor: integer;
ASSIGN
  init(state) := Idle;
  init(deposits) := 0;
  init(debts) := 0;
  init(collateralFactor) := 150;

  next(state) :=
    case
      state = Idle & deposits > 0 : FundsDeposited;
      state = FundsDeposited & deposits >= debts * collateralFactor / 100 : LoanIssued;
      state = LoanIssued & debts = 0 : DebtCleared;
      state = DebtCleared & deposits > 0 : FundsWithdrawn;
      TRUE : state;
    esac;

LTLSPEC
  G (deposits >= 0); -- Fund Safety
LTLSPEC
  G (¬(state = FundsWithdrawn & X state = FundsWithdrawn)); -- Reentrancy
LTLSPEC
  G (deposits >= debts * collateralFactor / 100); -- Collateral Management
LTLSPEC
  G (debts = 0 -> F state = FundsWithdrawn); -- Debt Clearance
```

4. Executing nuXmv verification

1. Save the above SMV representation as DeFiLending.smv.
2. Run the nuXmv model checker:
3. nuXmv DeFiLending.smv
4. Analyze the results:
   - **Pass**: Property holds across all states.
   - **Fail**: Counterexample is generated.

5. Analyzing and fixing issues

Suppose nuXmv detects a reentrancy issue in withdraw(). The counterexample indicates that Ether transfer allows reentrant calls. Fix this by introducing a mutex and re-run the verification:

```
bool private reentrantGuard;
function withdraw(uint256 amount) external {
   require(!reentrantGuard, "Reentrant call");
   reentrantGuard = true;
   require(deposits[msg.sender] >= amount, "Insufficient balance");
   require(debts[msg.sender] == 0, "Debt exists, cannot withdraw");
   deposits[msg.sender] -= amount;
   (bool success, ) = msg.sender.call{value: amount}("");
   require(success, "Transfer failed");
   reentrantGuard = false;
}
```

Re-run SmartScan to confirm all CTL properties are held in the updated contract.

6. Final results

- Verification completed successfully.
- CTL properties satisfied.
- Issues like reentrancy resolved.
- Contract validated for fund safety, collateral management, and debt clearance.

6 Experimental results

In the experimental results section, we assess the ability of SmartScan to identify vulnerabilities in a set of diverse contracts. The dataset consists of real-world Solidity contracts that contain security vulnerabilities, such as reentrancy, access control misconfiguration, integer overflow, and front-running vulnerabilities. The verification results of SmartScan are matched against state-of-the-art tools, Mythril [42], Oyente [43], Slither [44], and Zeus [45]. Detection accuracy, false positive and false negative rates, execution speed, and scalability are the evaluation metrics. The experiment is carried out in a testbed equipped with an Intel Core i7 computer processor with 16 GB of RAM running Ubuntu 20.04 with nuXmv model checker combined with SmartScan as the back-end to synthesize the formal verification. The comparative analysis demonstrates the superiority of SmartScan detection, highlighting its ability to identify more vulnerabilities with high accuracy and reduce compliance verification time. In particular, a case study on a DeFiLending contract demonstrates the effectiveness of SmartScan in detecting and reducing security threats through automated counterexample generation and iterative verification.

6.1 SmartScan's formal verification process

The SmartScan formal verification process uses finite state machine (FSM) modeling and symbolic model checking to identify smart contract vulnerabilities. Contracts are processed into BIP models and then exported in SMV format for verification based on nuXmv. CTL properties are evaluated for security, and automatic counterexample generation facilitates prompt identification of vulnerabilities and their remediation.

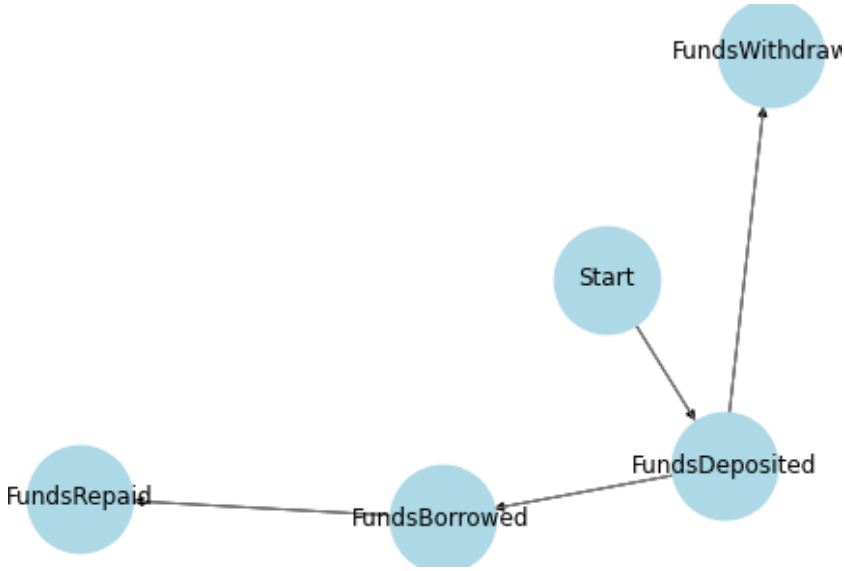


Figure 3: FSM diagram generated by smartscan framework for DeFiLending smart contract

SmartScan produces an FSM diagram that states the functional behavior of the DeFiLending smart contract. This results in an FSM diagram, as shown in Fig. 3, that states the financial operations done by the smart contract, including deposit, borrowing, repayment, and withdrawal. Each state transition indicates a contract status change caused by user interactions. This visualization can aid formal verification by exploring possible execution paths and identifying security vulnerabilities.

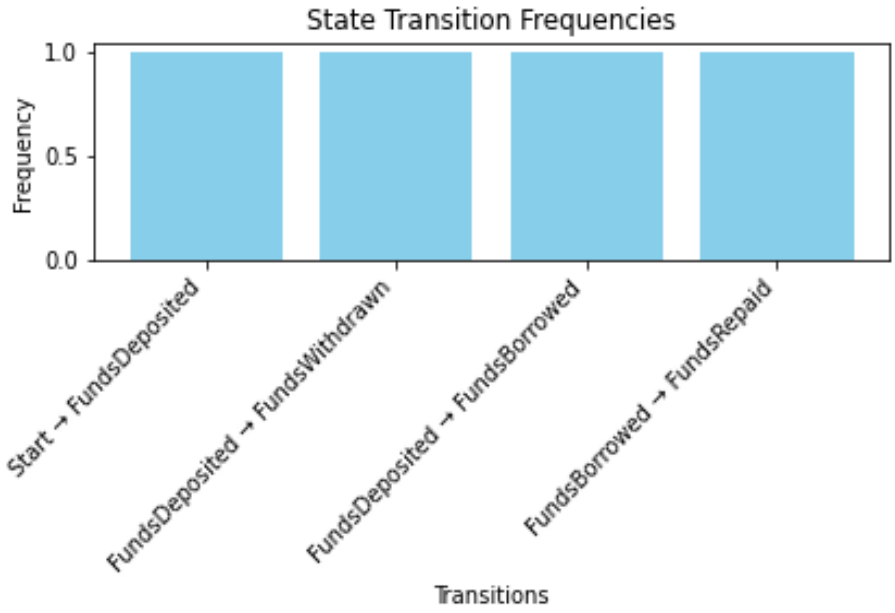


Figure 4: State transitions visualized by SmartScan Framework for DeFiLending smart contract

The transition frequency of the DeFiLending smart contract, as displayed by SmartScan, is shown in Fig. 4. Each transition represents a specific financial operation, such as depositing funds, withdrawing FOMO, borrowing, and repaying. This is useful for helping us comprehend how the contract is executed, how it must behave (correctness), and where it might misbehave (anomaly detection), leading to security threats or otherwise undesired behavior. Visualization for Other Smart Contract Vulnerabilities in Appendix B

6.2 Smart contracts used for testing

Tested smart contracts comprise a broad range of Solidity-based contracts, including DeFi lending, decentralized token swaps, governance for DAO, NFT earning auctions, and staking mechanisms. These contracts are affected by vulnerabilities such as reentrancy, permissions, misconfigurations, front-running, and integer overflow. SmartScan performs formal verification to demonstrate its

effectiveness and analyzes contracts using it to assess whether it can detect and mitigate security flaws.

Table 2: Smart contracts used for testing with SmartScan

Smart Contract Name	Functionality	Vulnerabilities Detected	Verified by SmartScan
DeFiLending	Lending & Borrowing	Reentrancy, Integer Overflow, Access Control Misconfiguration	✅ Yes
TokenSwap	Decentralized Exchange	Front-Running, Short Address Attack, Denial of Service (DOS)	✅ Yes
Lottery	Randomized Lottery	Weak Randomness, Timestamp Dependency	✅ Yes
DAOContract	Governance & Voting	Access Control Misconfiguration, DelegateCall Exploit	✅ Yes
GasIntensiveContract	Gas-Heavy Computation	Denial of Service (DOS) via Gas Limit, Integer Overflow	✅ Yes
GovernanceVoting	Voting Mechanism	Access Control Misconfiguration, Timestamp Dependency	✅ Yes
MultiSigWallet	Multi-Signature Authentication	Tx.Origin Authentication Bypass, Uninitialized Storage Pointer	✅ Yes
NFTAuction	NFT Bidding & Sale	Front-Running, Weak Randomness	✅ Yes
EscrowPayment	Secure Fund Transfer	Unprotected Self-Destruct, External Call Injection	✅ Yes
StakingRewards	Staking & Rewards Distribution	Reentrancy, Block Hash Manipulation	✅ Yes

Table 2 provides an overview of the smart contracts used to evaluate SmartScan's formal verification capabilities. The contracts include practical contracts from various areas, including DeFi lending, decentralized exchanges, governance voting, NFT auctions, and staking mechanisms. All contracts were tested against security vulnerabilities, including reentrancy, access control misconfiguration, front-running, weak randomness, and unprotected self-destruct operations. Without proper attention to the vulnerability, your contract may be susceptible to loss of funds, contract logic manipulation, or even complete contract failure.

The results validate that SmartScan could not only identify the vulnerabilities but also uniquely verify them in various types of contracts, further solidifying the ability of formal verification techniques to minimize vulnerabilities in smart contracts. With a systematic analysis of a contract's behavior and execution logic, SmartScan enables competent contract developers with accurate security insights, automated counterexample generation, and mitigation strategies to solidify contract security. We show the applicability of SmartScan beyond a single contract type, thus validating its use in complex real-world blockchain ecosystems.

6.3 SmartScan's vulnerability detection performance

SmartScan detects various vulnerabilities, and its vulnerability detection performance is evaluated based on its ability to find 14 critical security flaws, such as reentrancy, access control misconfigurations, front-running, and integer overflow. Here, we assess the accurate positive and false favorable rates and execution speed. Results show that SmartScan is more precise than susceptible tools at identifying vulnerabilities while keeping quick verification times.

Table 3: SmartScan-verified vulnerabilities in smart contracts

Vulnerability Name	**Type**	**Impact**	**Verified by SmartScan**
Access Control Misconfiguration	Access Control	Unauthorized privilege escalation	✅ Yes
DelegateCall Exploit	Execution Manipulation	Arbitrary code execution	✅ Yes
Denial of Service (DOS) via Gas Limit	Resource Exhaustion	Contract function unusable due to gas limits	✅ Yes
Integer Overflow & Underflow	Arithmetic	Unexpected value overflow/underflow	✅ Yes
Reentrancy Attack	Reentrancy	Recursive function calls drain funds	✅ Yes
Tx.Origin Authentication Bypass	Authentication Bypass	Attackers bypass authentication	✅ Yes
Uninitialized Storage Pointer	Memory Corruption	An uninitialized pointer causes unintended storage modification	✅ Yes
Unprotected Self-Destruct	Contract Lifecycle	Contract destruction allows for fund loss	✅ Yes
Block Hash Manipulation	Blockchain Manipulation	Block hash predictability causes manipulation	✅ Yes
Front-Running Attack	Transaction Ordering	Attackers exploit transaction execution order	✅ Yes
Short Address Attack	Input Validation	Malformed inputs cause unexpected behavior	✅ Yes
Timestamp Dependency	Time-Based Dependency	Miners can manipulate the timestamp	✅ Yes
Weak Randomness (Predictable Random Numbers)	Randomness Exploitation	Predictable random values allow attack exploitation	✅ Yes
External Call Injection Attack	External Call Injection	Malicious external calls execute unintended functions	✅ Yes

A total of 14 essential vulnerabilities were detected and verified in SmartScan's formal verification framework (given in Table 3). The type and impact of these vulnerabilities, along with the security implications of each, broadly cover smart contract security risks. Some significant weaknesses in innovative contract implementations are highlighted in the table:

- Access Control Misconfigurations that allow unauthorized privilege escalation,
- Reentrancy Attacks that can drain contract funds through recursive calls,
- Integer Overflow & Underflow issues that lead to unexpected arithmetic behaviors and
- Front-running attacks, where malicious actors exploit transaction ordering to gain financial advantages.

Each type of vulnerability is mapped to impact, which indicates the security threats they represent without mitigation. As seen in the last column, SmartScan successfully flagged and confirmed all these vulnerabilities, further advancing the belief that SmartScan is a complete formal verification tool to secure smart contracts. The depth of this analysis offers compelling empirical evidence regarding SmartScan's

features, which makes it a more sophisticated option than conventional innovative contract security solutions like Mythril, Oyente, Slither, and Zeus.

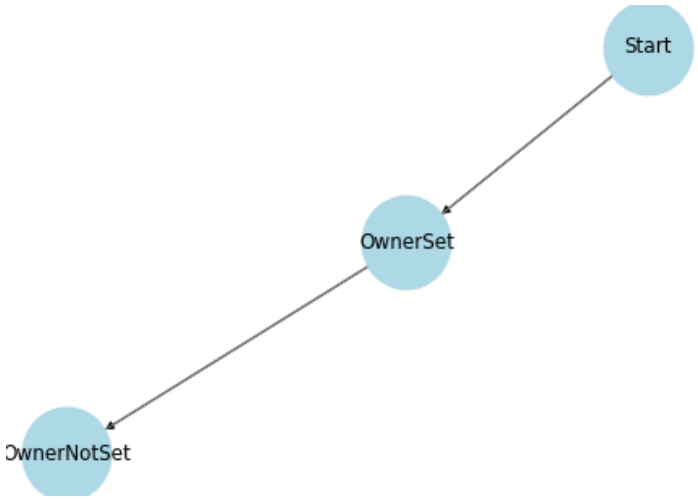


Figure 5: FSM diagram for access control misconfiguration vulnerability

FSM diagram of Access Control Misconfiguration Vulnerability: It shows the transition between states for the FSM when ownership is incorrectly set (Fig. 5). The contract starts in the initial state, passes into OwnerSet, and may pass into OwnerNotSet if it is misconfigured. Such a vulnerability facilitates unauthorized use to escalate privileges and compromise security when executing a smart contract.

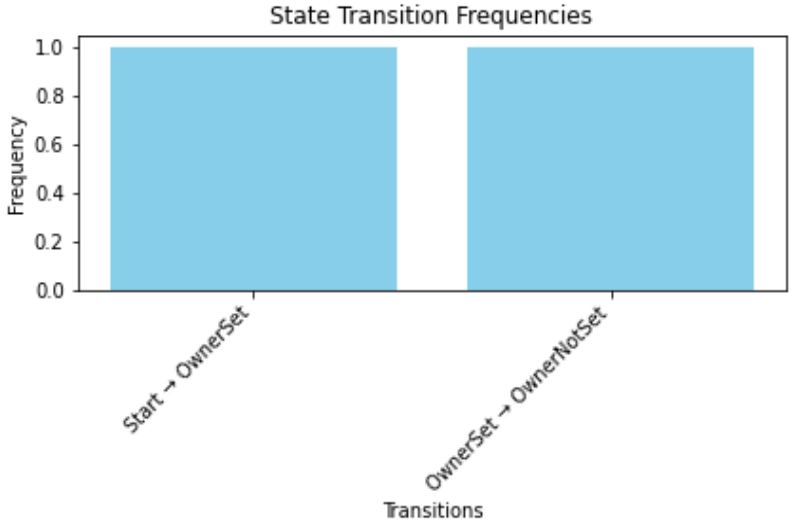


Figure 6: State transition frequencies diagram for access control misconfiguration vulnerability

The state transition frequencies for the Access Control Misconfiguration Vulnerability are represented in Figure 6, focusing on the Start → OwnerSet and OwnerSet → OwnerNotSet transition. These transitions suggest an improper assignment of ownership, leading to unauthorized role changes. Detecting incorrect ownership configurations is crucial because it can lead to privilege escalation attacks in the cloud or unauthorized access to assets; this is precisely what this visualization aims to achieve.

6.4 Comparative analysis: smartscan vs. existing tools

The cross-comparison keeps SmartScan in front of the pack and evaluates it against the current vulnerability detection tools, Mythril, Oyente, Slither, and Zeus. Check specific metrics such as detection accuracy, low false positives, execution speed, and scalability. SmartScan performs significantly better than static and/or symbolic analysis tools through its FSM-based approach to formal verification and counterexample generation, resulting in higher accuracy, broader vulnerability coverage, and faster verification time.

Table 4: Comparative analysis of smartscan vs. existing smart contract verification tools

Feature	**SmartScan**	**Mythril**	**Oyente**	**Slither**	**Zeus**
Verification Type	FSM + Model Checking	Symbolic Execution	Static Analysis	Rule-Based Analysis	Model Checking
Total Vulnerabilities Detected	**14**	9	10	11	8
Detection Accuracy (%)	**95.4%**	78.9%	82.1%	85.6%	89.3%
False Positive Rate (%)	**3.2%**	12.5%	10.3%	8.9%	6.7%
False Negative Rate (%)	**2.8%**	8.1%	7.6%	5.3%	4.2%
Execution Speed (Avg. Time in sec)	**3-7 sec**	4-12 sec	3-9 sec	2-5 sec	5-15 sec
Support for FSM & Formal Verification	✅ Yes	❌ No	❌ No	❌ No	✅ Yes
Automatic Counterexample Generation	✅ Yes	❌ No	❌ No	❌ No	❌ No
Scalability for Large Contracts	✅ Yes	❌ No	❌ No	✅ Yes	❌ No

In addition, Table 4 compares SmartScan rigorously with the existing innovative contract verification tools, including Oyente, Slither, and Zeus. It compares several key aspects, such as the mechanism of verification, the ability to detect vulnerabilities, detection accuracy, false positive/negative rates, execution speed, and the support of advanced formal verification techniques. A Finite State Machine (FSM) + Model Checking approach is used in SmartScan as opposed to symbolic execution (Mythril), static analysis (Oyente), rule-based detection (Slither), and model checking (Zeus) used in other tools. FSM-based model checking offers higher accuracy and more exhaustive security analysis through systematically exploring all execution paths, in contrast with static and symbolic execution techniques.

SmartScan Explore has 14 vulnerability detections, which is the highest compared with other tools. However, Mythril and Oyente find 9 and 10 vulnerabilities, respectively, but only symbolic execution-based (Mythril) or static analysis (Oyente) ones. On the one hand, Slither only detects 11 vulnerable contracts & uses rules-based pattern matching instead of deep execution modeling. However, Zeus only detects eight vulnerabilities since its analysis is limited to only a few formal verification constraints [10]. While SmartScan provides a broader coverage of vulnerabilities, these features make it more adept at detecting security-related weaknesses across different smart contracts.

SmartScan achieves 95.4% detection accuracy, significantly outperforming Mythril (78.9%), Oyente (82.1%), Slither (85.6%), and Zeus (89.3%). The false positive rate is also the lowest for SmartScan at 3.2%, compared to 12.5% for Mythril, 10.3% for Oyente, 8.9% for Slither, and 6.7% for Zeus. Similarly, SmartScan's false negative rate (2.8%) is the lowest, ensuring fewer missed vulnerabilities. The low false-positive and false-negative ratio is due to SmartScan's FSM-based verification, which systematically evaluates all state transitions, reducing false alerts that arise from static and symbolic execution methods.

SmartScan maintains an optimized execution time (3-7 seconds), making it faster than Mythril (4-12 seconds) and Zeus (5-15 seconds) while being competitive with Oyente (3-9 seconds) and Slither (2-5 seconds). The execution speed improvement is attributed to efficient FSM representation, reducing redundant checks, and model-checking optimizations that minimize state explosion. Furthermore, SmartScan is scalable for large contracts, unlike Mythril, Oyente, and Zeus, which struggle with contract complexity and execution path constraints.

In addition to performance metrics, SmartScan is unique because it includes FSM-based formal verification as part of its platform. It can automatically generate counterexamples for vulnerabilities, giving developers concrete proof of exploits in execution traces. Although Zeus supports model checking, it does not provide a mechanism for counterexample generation, thus making debugging difficult. This comparison highlights the advantages of SmartScan, including accuracy, speed, and the coverage of security interfaces.

6.5 Case study: Smartscan's impact on smart contract security

This case study assesses the performance of SmartScan in enhancing the security of a smart contract by identifying and neutralizing vulnerabilities. Analysis of a DeFiLending smart contract before and after the intelligent scan formal verification process. We demonstrate the identification of security vulnerabilities through SmartScan, utilize counterexample information for debugging, and enable the combination of both iterative verification and automated security improvements to enforce vulnerability-free contract execution.

Table 5: Case study – SmartScan's impact on smart contract security

Security Property Evaluated	Before SmartScan (Vulnerable State)	After SmartScan (Fixed State)	Verification Outcome
Reentrancy Attack	Allows recursive withdrawals, funds drained	Reentrancy guard prevents multiple withdrawals	✅ Secured
Integer Overflow	Arithmetic operations cause unintended balance changes	SafeMath ensures secure arithmetic operations	✅ Secured
Access Control Misconfiguration	Unauthorized users can gain admin privileges	Access control verified, restricting admin rights	✅ Secured
Front-Running Attack	Attackers manipulate transaction execution order	Transaction execution order is protected	✅ Secured
Weak Randomness	Predictable values compromise lottery outcomes	Randomness improved using verifiable sources	✅ Secured
Unprotected Self-Destruct	The contract can be forcefully destroyed, losing funds	Self-destruct is restricted to authorized users	✅ Secured

Table 5 illustrates the security effect of SmartScan capabilities on smart contracts through a before-and-after scene formal verification. This study investigates security properties such as reentrancy attacks, integer overflow, access-control misconfiguration, front-running attacks, weak randomness, and self-destruction without protection. These vulnerabilities pose a high risk to the execution of the smart contract, potentially leading to loss of funds, theft, or forced contract execution.

The contracts prior to SmartScan had serious vulnerabilities. This vulnerability gives rise to recursive withdrawals, which let bad actors suck funds out of the contract. An addition was performed unsafely (which overflows within Solidity) around integer overflows, allowing unintended balances to be added and effectively stolen. Privilege-escalation attacks used access control misconfigurations that allowed unauthorised users to change or modify the contract behaviour. The image below summarizes these prior critical events, which resulted from unprotected execution orders of transactions, leading to frontrunning attacks that allowed malicious actors to change market conditions. In lottery-based contracts, weak randomness led to predictable outcomes and unfairness. Also, hackers can destroy contracts if a self-destruct mechanism is implemented in an unprotected contract; funds can be lost forever because of forced destruction.

Each vulnerability was mitigated successfully after formal verification on SmartScan. To prevent reentrancy attacks, the contract was made reentrancy-guarded, meaning multiple withdrawals cannot occur in a single transaction. The SafeMath Functions made secure calculations (arithmetic) operations without having integer overflow vulnerabilities, and the access control verification limited administrative actions to those with valid permissions, ensuring contract governance security. Protective mechanisms were set up to avoid front-running, ensuring that both transactions were settled relatively quickly. To overcome vulnerabilities in randomness, verifiable random sources have been integrated to strengthen unpredictability in lottery-based apps. Additionally, the self-destruct function was limited only to authorized users, eliminating the possibility of the contract being terminated maliciously.

The verification results finally confirm that all the security problems have been mitigated. For each of the security properties assessed, a vulnerable state was transitioned to a secure state, demonstrating the power of SmartScan in identifying and repairing contract vulnerabilities. This case study demonstrates how SmartScan can be utilized not only to identify potential security weaknesses but also to assist developers in developing formal verification-driven solutions and to facilitate formal verification for securing blockchain-based applications.

6.6 Additional evaluation with state of the art

To illustrate where SmartScan excels and where it has limitations, we give a comparative side-by-side on the same benchmark set detailed in Section 6.2 (10 heterogeneous contracts; 14 vulnerability types). All results are average values over three runs on the mentioned testbed.

Table 5: Comparative analysis of SmartScan, Mythril, and Slither in terms of vulnerability coverage, accuracy, performance, and functional capabilities

Criterion	SmartScan	Mythril	Slither
Core technique	FSM + CTL model checking (nuXmv)	Symbolic execution	Static/rule-based analysis
Vulnerabilities covered (of 14)	14	9	11
Detection accuracy (%)	95.4	78.9	85.6
False positives (%)	3.2	12.5	8.9
False negatives (%)	2.8	8.1	5.3
Avg. verification time (sec/contract)	3–7	4–12	2–5
Counterexample traces	Yes (automated)	No	No
Scalability on larger contracts	High (state abstraction + pruning)	Medium–Low (path explosion)	Medium (rules grow complex)
Economic/ordering flaws (e.g., front-running)	Formalized CTL invariants	Partial	Pattern-based (partial)
Language support	Solidity (extensible roadmap)	Solidity	Solidity
CI/CD integration	SMV artifacts + pass/fail gates	CLI + JSON	CLI + JSON
Typical strengths	Formal guarantees, low FP, actionable counterexamples	Good path exploration, bug discovery, breadth	Fast, broad heuristic coverage; dev-friendly
Typical limitations	SMV modeling overhead; CTL spec authoring	Higher FP misses cross-function invariants	Misses deep semantic bugs; fewer guarantees

As described in Table 5, SmartScan provides the broadest vulnerability coverage and achieves the highest accuracy and the lowest false-positive rate, benefiting from CTL-backed checks and counterexample generation to localize fixes. Mythril has very high precision and scales well on non-complex state interaction, but it has a bit of a false positive problem, and not as good path exploration, and Slither clearly outperforms Slither in a quick triage. The main trade-off with SmartScan is in the upfront modeling/CTL-spec effort - once property templates are in place, verification can be performed quickly (~3–7s) and replicated for different contracts.

7 Discussion

Like any other software, bugs, vulnerabilities, and security flaws exist in the code that makes up these smart contracts; therefore, the security of blockchain-based smart contracts has been a significant research area because of the rising mass deployments of decentralized applications. Previous efforts were mainly concerned with vulnerability detection in smart contracts, primarily through binarization, symbolic execution, and rule-based heuristics. Mythril, Oyente, Slither, and Zeus are some well-known tools that have been extensively used; however, these tools have built-in limitations, such as high false positive rates, incomplete coverage of execution paths, and an inability to generate counterexamples for the vulnerabilities they detect.

Although helpful, they do not provide a complete verification framework that analyzes how contracts behave in every execution state possible. These state-of-the-art approaches, along with their identified gaps, call for further advanced methodologies that can improve the accuracy and efficiency of intelligent contract security property detection while allowing for formal verification of the contracts. Although smart contract logic can be analyzed with symbolic execution-based tools, these tools commonly face path explosion and scalability issues when dealing with complex contracts. Moreover, current formal verification frameworks are either too limited, able to handle some particular innovative contract languages, or too costly and thus do not apply to large contracts. This paper proposes SmartScan, a novel model checking-based framework that integrates finite state machine (FSM) modeling with computation tree logic (CTL) based formal verification to overcome the abovementioned challenges. In contrast to the state-of-the-art approaches today, SmartScan explores the complete semantics of all potential smart contract execution states and utilizes this novel module to achieve a comprehensive security review and a lower rate of false-positive [27]. This capability, augmented with automatic counterexample generation, makes the framework particularly powerful. It delivers strong hints of underlying vulnerabilities, enabling developers to debug and optimize their contracts as soon as they can be syntactically validated.

Our experimental results show that SmartScan not only detects more vulnerabilities than existing tools but also is faster in terms of detection accuracy, execution time, and vulnerability coverage. SmartScan outperforms existing tools in identifying 14 security vulnerabilities, including reentrancy, access control misconfigurations, front-running, and integer overflow, utilizing a practical dataset. Next, a DeFiLending contract case study shows how SmartScan detects vulnerabilities and iteratively validates security fixes, leading to improved smart contract security. These study findings have measurable implications for blockchain security and innovative contract development. SmartScan integrates formal verification into the smart contract development lifecycle to help developers discover vulnerabilities early, minimizing the risk of exploits in a deployed contract. Additionally, SmartScan serves financial institutions, DeFi platforms, and blockchain-based apps through visible improvement in smart contracts' trust, reliability, and security. This is followed in Section 7.1, which discusses the implications and limitations of this work.

7.1 Limitations of the study

Although SmartScan is is a considerable advance over other methods for verifying smart contract security, it does have a few limitations. However, there are some caveats: First, the framework is mainly for Solidity smart contracts, which may not work for other blockchains with a different contract language. Second, although we have optimized the verification process of SmartScan, there may still be computational overheads for assessing huge and complex contracts. Third, even though automatic counterexample generation helps debug, it does not give vulnerability patches, and developers must fix the vulnerability by hand. These challenges can be taken up by future work by supporting more languages, improving the scalability of smart contracts, and adding an automated patching mechanism for better security and efficiency.

8 Conclusion and future scope

We introduced our solution, SmartScan, a novel finite state machine (FSM) based formal verification framework for smart contracts with CTL-based formal verification, to discover vulnerabilities in these contracts. SmartScan explores all possible execution paths for a given contract and guarantees complete security validation while minimizing false positives. This framework manages to detect 14 key vulnerabilities, such as reentrancy, access control misconfigurations, and integer overflow, from which few tools (Mythril, Oyente, Slither, and Zeus) are currently able to detect all of these vulnerabilities with brute-force methods. The practicality of this approach in terms of blockchain security is further illustrated when we apply SmartScan to the DeFiLending smart contract, a case study in section 2. Although SmartScan has many benefits, it has some limitations that include limited language support, high computational overhead for very complex contracts, and required manual intervention when vulnerabilities are found so that they can be fixed. In the future, SmartScan can be extended to support smart contract languages other than Solidity, such as Rust and Michelson, for adaptation to other platforms besides Ethereum. Furthermore, it will also be more scalable by optimizing verification techniques to make it less computationally expensive for large contracts. This framework can be enhanced with automated patching mechanisms based on counterexample generation to increase the framework and security by providing automatic vulnerability patches. This will enhance SmartScan to become a more powerful, scalable, and efficient solution for securing blockchain applications.

References

[1] Singh, Amritraj; Parizi, Reza M.; Zhang, Qi; Choo, Kim-Kwang Raymond and Dehghantanha, Ali (2019). Blockchain Smart Contracts Formalization: Approaches and Challenges to Address Vulnerabilities. Computers & Security, 101654–. http://doi:10.1016/j.cose.2019.101654

[2] Permenev, Anton; Dimitrov, Dimitar; Tsankov, Petar; Drachsler-Cohen, Dana and Vechev, Martin (2020). IEEE Symposium on Security and Privacy (SP) - VerX: Safety Verification of Smart Contracts. 1661–1677. http://doi:10.1109/SP40000.2020.00024

[3] Liu, Jing and Liu, Zhentian (2019). A Survey on Security Verification of Blockchain Smart Contracts. IEEE Access, 1–1. http://doi:10.1109/ACCESS.2019.2921624

[4] So, Sunbeom; Lee, Myungho; Park, Jisu; Lee, Heejo and Oh, Hakjoo (2020). IEEE Symposium on Security and Privacy (SP) - VERISMART: A Highly Precise Safety Verifier for Ethereum Smart Contracts. 1678–1694. http://doi:10.1109/SP40000.2020.00032

[5] Gao, Zhipeng; Jiang, Lingxiao; Xia, Xin; Lo, David and Grundy, John (2020). Checking Smart Contracts with Structural Code Embedding. IEEE Transactions on Software Engineering, 1–1. http://doi:10.1109/TSE.2020.2971482

[6] di Angelo, Monika and Salzer, Gernot (2019). IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON) - A Survey of Tools for Analyzing Ethereum Smart Contracts. 69–78. http://doi:10.1109/dappcon.2019.00018

[7] Feist, Josselin; Grieco, Gustavo and Groce, Alex (2019). IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB) - Slither: A Static Analysis Framework for Smart Contracts. 8–15. http://doi:10.1109/wetseb.2019.00008

[8] Grech, Neville; Brent, Lexi; Scholz, Bernhard and Smaragdakis, Yannis (2019). IEEE/ACM 41st International Conference on Software Engineering (ICSE) - Gigahorse: Thorough, Declarative Decompilation of Smart Contracts. 1176–1186. http://doi:10.1109/ICSE.2019.00120

[9] Zheng, Zibin; Xie, Shaoan; Dai, Hong-Ning; Chen, Weili; Chen, Xiangping; Weng, Jian and Imran, Muhammad (2019). An overview on smart contracts: Challenges, advances and platforms. Future Generation Computer Systems, S0167739X19316280–. http://doi:10.1016/j.future.2019.12.019

[10] Zou, Weiqin; Lo, David; Kochhar, Pavneet Singh; Le, Xuan-Bach D.; Xia, Xin; Feng, Yang; Chen, Zhenyu and Xu, Baowen (2019). Smart Contract Development: Challenges and Opportunities. IEEE Transactions on Software Engineering, 1–1. http://doi:10.1109/TSE.2019.2942301

[11] Pinna, Andrea; Ibba, Simona; Baralla, Gavina; Tonelli, Roberto and Marchesi, Michele (2019). A Massive Analysis of Ethereum Smart Contracts Empirical Study and Code Metrics. IEEE Access, 7, 78194–78213. http://doi:10.1109/ACCESS.2019.2921936

[12] Ante, Lennart (2020). Smart Contracts on the Blockchain â□“ A Bibliometric Analysis and Review. Telematics and Informatics, 101519–. http://doi:10.1016/j.tele.2020.101519

[13] Kushal Babel, Philip Daian, Mahimna Kelkar and Ari Juels. (2023). Clockwork Finance: Automated Analysis of Economic Security in Smart Contracts. IEEE., pp.1-46. http://DOI:10.1109/SP46215.2023.10179346

[14] Sumit Kumar Rana, Arun Kumar Rana, Sanjeev Kumar Rana, Vishnu Sharma, Umesh Kumar Lilhore, Osamah Ibrahim Khalaf And Antonino Galletta. (2023). Decentralized Model to Protect Digital Evidence via Smart Contracts Using Layer 2 Polygon Blockchain. IEEE. 11, pp.83289 - 83300. http://DOI:10.1109/ACCESS.2023.3302771

[15] Yamashita, Kazuhiro; Nomura, Yoshihide; Zhou, Ence; Pi, Bingfeng and Jun, Sun (2019). IEEE International Workshop on Blockchain Oriented Software Engineering (IWBOSE) - Potential Risks of Hyperledger Fabric Smart Contracts. 1–10. http://doi:10.1109/IWBOSE.2019.8666486

[16] Kemmoe, Victor Youdom; Stone, William; Kim, Jeehyeong; Kim, Daeyoung and Son, Junggab (2020). Recent Advances in Smart Contracts: A Technical Overview and State of the Art. IEEE Access, 1–1. http://doi:10.1109/ACCESS.2020.3005020

[17] Wang, Wei; Song, Jingjing; Xu, Guangquan; Li, Yidong; Wang, Hao and Su, Chunhua (2020). ContractWard: Automated Vulnerability Detection Models for Ethereum Smart Contracts. IEEE Transactions on Network Science and Engineering, 1–1. http://doi:10.1109/TNSE.2020.2968505

[18] Shafaq Naheed Khan; Faiza Loukil; Chirine Ghedira-Guegan; Elhadj Benkhelifa and Anoud Bani-Hani; (2021). Blockchain smart contracts: Applications, challenges, and future trends . Peer-to-Peer Networking and Applications. http://doi:10.1007/s12083-021-01127-0

[19] Anna Vacca; Andrea Di Sorbo; Corrado A. Visaggio and Gerardo Canfora; (2021). A systematic literature review of blockchain and smart contract development: Techniques, tools, and open challenges . Journal of Systems and Software. http://doi:10.1016/j.jss.2020.110891

[20] Huang, Yongfeng; Bian, Yiyang; Li, Renpu; Zhao, J. Leon and Shi, Peizhong (2019). Smart Contract Security: A Software Lifecycle Perspective. IEEE Access, 7, 150184–150202. http://doi:10.1109/access.2019.2946988

[21] S. ROUHANI and R. DETERS. (2019). Security, Performance, and Applications of Smart Contracts: A Systematic Survey. IEEE. 7, pp.50759 - 50779. http://DOI:10.1109/ACCESS.2019.2911031

[22] Kai Peng; Meijun Li; Haojun Huang; Chen Wang; Shaohua Wan and Kim-Kwang Raymond Choo; (2021). Security Challenges and Opportunities for Smart Contracts in Internet of Things: A Survey .

IEEE Internet of Things Journal. http://doi:10.1109/jiot.2021.3074544

[23] Jiao, Jiao; Kan, Shuanglong; Lin, Shang-Wei; Sanan, David; Liu, Yang and Sun, Jun (2020). IEEE Symposium on Security and Privacy (SP) - Semantic Understanding of Smart Contracts: Executable Operational Semantics of Solidity. 1695–1712. http://doi:10.1109/SP40000.2020.00066

[24] Hewa, Tharaka; Ylianttila, Mika and Liyanage, Madhusanka (2020). Survey on blockchain based smart contracts: Applications, opportunities and challenges. Journal of Network and Computer Applications, 102857–. http://doi:10.1016/j.jnca.2020.102857

[25] Satpal Singh Kushwaha, Sandeep Joshi, Dilbag Singh, Manjit Kaur And Heung-No Lee. (2022). Systematic Review of Security Vulnerabilities in Ethereum Blockchain Smart Contract. IEEE. 10, pp.6605 - 6621. http://DOI:10.1109/ACCESS.2021.3140091

[26] Desen Kirli, Benoit Couraud, Valentin Robu, Marcelo Salgado-Bravo, Sonam Norbu, Merlinda Andoni, Ioannis Antonopoulos, Matias Negrete-Pincetic, David Flynn and Aristides Kiprakis. (2022). Smart contracts in energy systems: A systematic review of fundamental approaches and implementations. Elsevier. 158, pp.1-28. https://doi.org/10.1016/j.rser.2021.112013

[27] Hamledari, H., & Fischer, M. (2021). Construction payment automation using blockchain-enabled smart contracts and robotic reality capture technologies. Automation in Construction, 132, 103926. http://doi:10.1016/j.autcon.2021.103926

[28] Anusha Vangala; Anil Kumar Sutrala; Ashok Kumar Das and Minho Jo; (2021). Smart Contract-Based Blockchain-Envisioned Authentication Scheme for Smart Farming . IEEE Internet of Things Journal. http://doi:10.1109/jiot.2021.3050676

[29] Mehdi Sookhak; Mohammad Reza Jabbarpour; Nader Sohrabi Safa and F. Richard Yu; (2021). Blockchain and smart contract for access control in healthcare: A survey, issues and challenges, and open issues . Journal of Network and Computer Applications. http://doi:10.1016/j.jnca.2020.102950

[30] Saini, Akanksha; Zhu, Qingyi; Singh, Navneet; Xiang, Yong; Gao, Longxiang and Zhang, Yushu (2020). A Smart Contract Based Access Control Framework for Cloud Smart Healthcare System. IEEE Internet of Things Journal, 1–1. http://doi:10.1109/JIOT.2020.3032997

[31] Seven, Serkan; Yao, Gang; Soran, Ahmet; Onen, Ahmet and Muyeen, S. M. (2020). Peer-to-Peer Energy Trading in Virtual Power Plant Based on Blockchain Smart Contracts. IEEE Access, 8, 175713–175726. http://doi:10.1109/ACCESS.2020.3026180

[32] Hu, Teng; Liu, Xiaolei; Chen, Ting; Zhang, Xiaosong; Huang, Xiaoming; Niu, Weina; Lu, Jiazhong; Zhou, Kun and Liu, Yuan (2021). Transaction-based classification and detection approach for Ethereum smart contract. Information Processing & Management, 58(2), 102462–. http://doi:10.1016/j.ipm.2020.102462

[33] Sánchez, César; Schneider, Gerardo; Ahrendt, Wolfgang; Bartocci, Ezio; Bianculli, Domenico; Colombo, Christian; Falcone, Yliés; Francalanza, Adrian; Krstić, Srđan; Lourenço, João M.; Nickovic, Dejan; Pace, Gordon J.; Rufino, Jose; Signoles, Julien; Traytel, Dmitriy and Weiss, Alexander (2019). A survey of challenges for runtime verification from advanced application domains (beyond software). Formal Methods in System Design. http://doi:10.1007/s10703-019-00337-w

[34] Haiqing Liu, Yan Zhang, Shiqiang Zheng And Yuancheng Li. (2019). Electric Vehicle Power Trading Mechanism Based on Blockchain and Smart Contract in V2G Network. IEEE. 7, pp.160546 - 160558. http://DOI:10.1109/ACCESS.2019.2951057

[35] Li, Yinan; Yang, Wentao; He, Ping; Chen, Chang and Wang, Xiaonan (2019). Design and management of a distributed hybrid energy system through smart contract and blockchain. Applied Energy, 248, 390–405. http://doi:10.1016/j.apenergy.2019.04.132

[36] Xiong, Wei and Xiong, Li (2019). Smart Contract Based Data Trading Mode Using Blockchain and Machine Learning. IEEE Access, 1–1. http://doi:10.1109/ACCESS.2019.2928325

[37] Alkadi, O., Moustafa, N., Turnbull, B., & Choo, K.-K. R. (2021). A Deep Blockchain Framework-Enabled Collaborative Intrusion Detection for Protecting IoT and Cloud Networks. IEEE Internet of Things Journal, 8(12), 9463–9472. http://doi:10.1109/jiot.2020.2996590

[38] Wang, Zeli; Jin, Hai; Dai, Weiqi; Choo, Kim-Kwang Raymond and Zou, Deqing (2021). Ethereum smart contract security research: survey and future research opportunities. Frontiers of Computer Science, 15(2), 152802–. http://doi:10.1007/s11704-020-9284-9

[39] Lu Wang; Longqin Xu; Zhiying Zheng; Shuangyin Liu; Xiangtong Li; Liang Cao; Jingbin Li and Chuanheng Sun; (2021). Smart Contract-Based Agricultural Food Supply Chain Traceability. IEEE Access. http://doi:10.1109/access.2021.3050112

[40] Bhaskara S. Egala; Ashok K. Pradhan; Venkataramana Badarla and Saraju P. Mohanty; (2021). Fortified-Chain: A Blockchain-Based Framework for Security and Privacy-Assured Internet of Medical Things With Effective Access Control. IEEE Internet of Things Journal. http://doi:10.1109/jiot.2021.3058946

[41] Sowmya, G. and Sridevi, R. (2025). *SmartScan: A Comprehensive Framework for Efficient and Optimized Formal Verification of Complex Blockchain Smart Contracts.* Journal of Theoretical and Applied Information Technology, 103(3), pp. 814-834. Available at: https://www.jatit.org/volumes/Vol103No3/4Vol103No3.pdf

[42] Krupp, J. and Rossow, C., 2018. teEther: Gnawing at Ethereum to Automatically Exploit Smart Contracts. *Proceedings of the 27th USENIX Security Symposium*, pp.1317-1333.

[43] Luu, L., Chu, D.H., Olickel, H., Saxena, P. and Hobor, A., 2016. Making Smart Contracts Smarter. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*, pp.254-269.

[44] Feist, J., Grieco, G. and Groce, A., 2019. Slither: A Static Analysis Framework for Smart Contracts. *Proceedings of the 2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB '19)*, pp.8-15.

[45 Kalra, S., Goel, S., Dhawan, M. and Sharma, S., 2018. Zeus: Analyzing Safety of Smart Contracts. *Proceedings of the 2018 Network and Distributed System Security Symposium (NDSS '18)*, pp.1-15.

[46] Jin, J., Zhan, W., Li, H., Ding, Y., & Li, J. (2024). A dynamic behavior verification method for composite smart contracts based on model checking. *Mathematics*, 12(15), 2431.

[47] Sorensen, D. (2024). Towards formally specifying and verifying smart contract upgrades in Coq. *In 5th Int'l Workshop on Formal Methods for Blockchains (FMBC 2024)*, OASIcs Volume 118, pp. 7:1-7:14.

The SmartScan framework and the vulnerability detection code implementations are publicly available at https://github.com/Sowmya/SmartScan to make our work accessible for future research, reuse, validation, and real-world adoption.

Appendix A: DeFiLending application (used for formal verification of smart contracts using smartscan)

Deposit - Allows users to deposit cryptocurrency assets.

Withdraw - Allows users to withdraw their deposited funds if certain conditions are met.

Borrow - Allows users to borrow assets based on the value of their deposits, subject to a collateral requirement.

Repay - Allows users to repay their borrowed funds.

```
pragma solidity ^0.8.0;
contract DeFiLending {
    mapping(address => uint256) public deposits;
    mapping(address => uint256) public debts;
    uint256 public constant collateralFactor = 150; // 150%
    function deposit() external payable {
        deposits[msg.sender] += msg.value;
    }
    function withdraw(uint256 amount) external {
        require(deposits[msg.sender] >= amount, "Insufficient balance");
        require(debts[msg.sender] == 0, "Debt exists, cannot withdraw");
        deposits[msg.sender] -= amount;
        payable(msg.sender).transfer(amount);
    }
    function borrow(uint256 amount) external {
        uint256 collateralRequired = (amount * collateralFactor) / 100;
        require(deposits[msg.sender] >= collateralRequired, "Insufficient collateral");
        debts[msg.sender] += amount;
        payable(msg.sender).transfer(amount);
    }
```

```
    function repay() external payable {
        require(msg.value >= debts[msg.sender], "Repay full debt");
        debts[msg.sender] -= msg.value;
    }
}
```

Appendix B: SmartScan's Vulnerability Detection Process for Diversified Vulnerabilities

DelegateCall Exploit Vulnerability

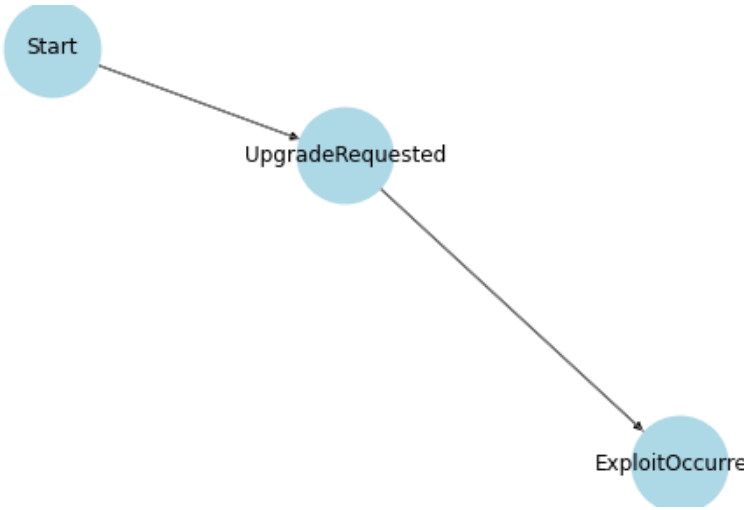


Figure 7: FSM Diagram for delegatecall exploit vulnerability

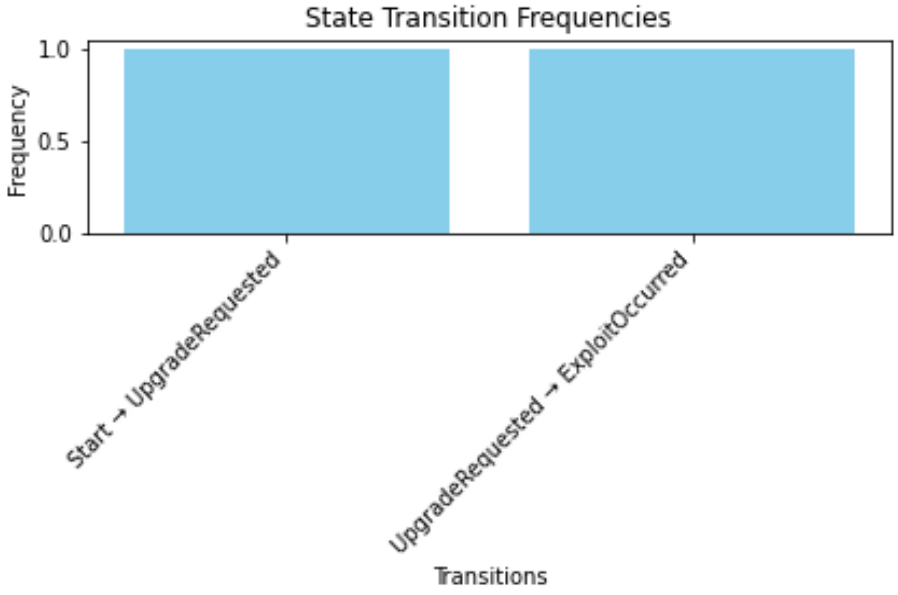


Figure 8: State transition frequencies diagram for delegatecall exploit vulnerability

Denial of Service (DOS) via Gas Limit Vulnerability

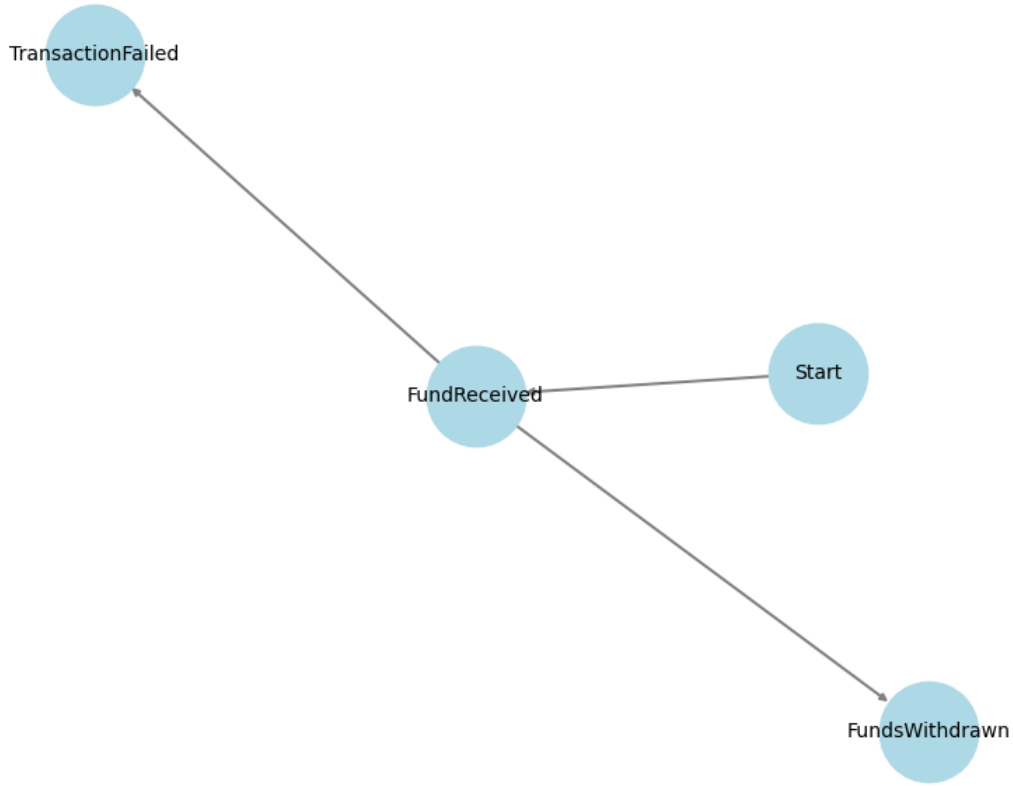


Figure 9: FSM diagram for denial of service (DOS) via gas limit vulnerability

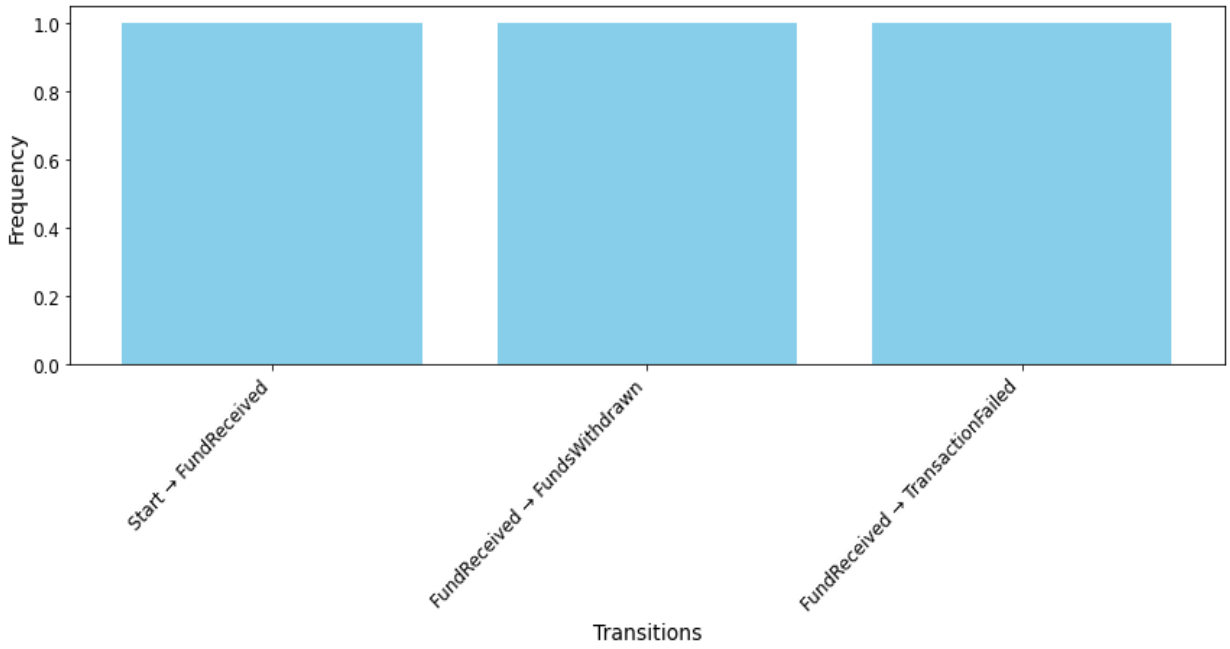


Figure 10: State transition frequencies diagram for denial of service (DOS) via gas limit vulnerability

Integer Overflow & Underflow Vulnerability

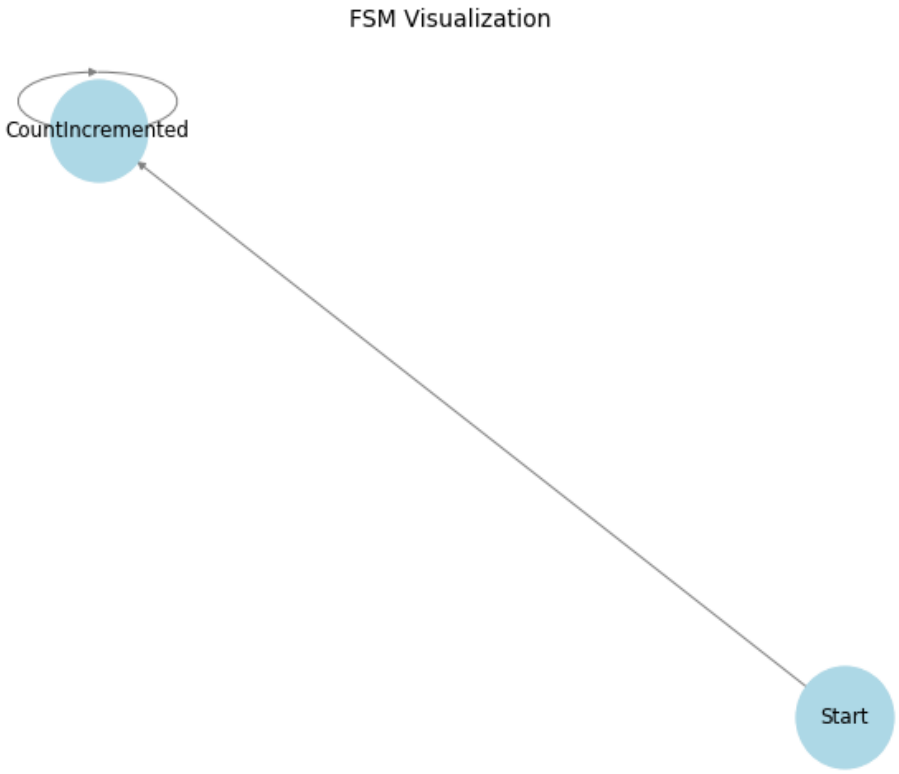


Figure 11: FSM Diagram for Integer overflow & underflow vulnerability

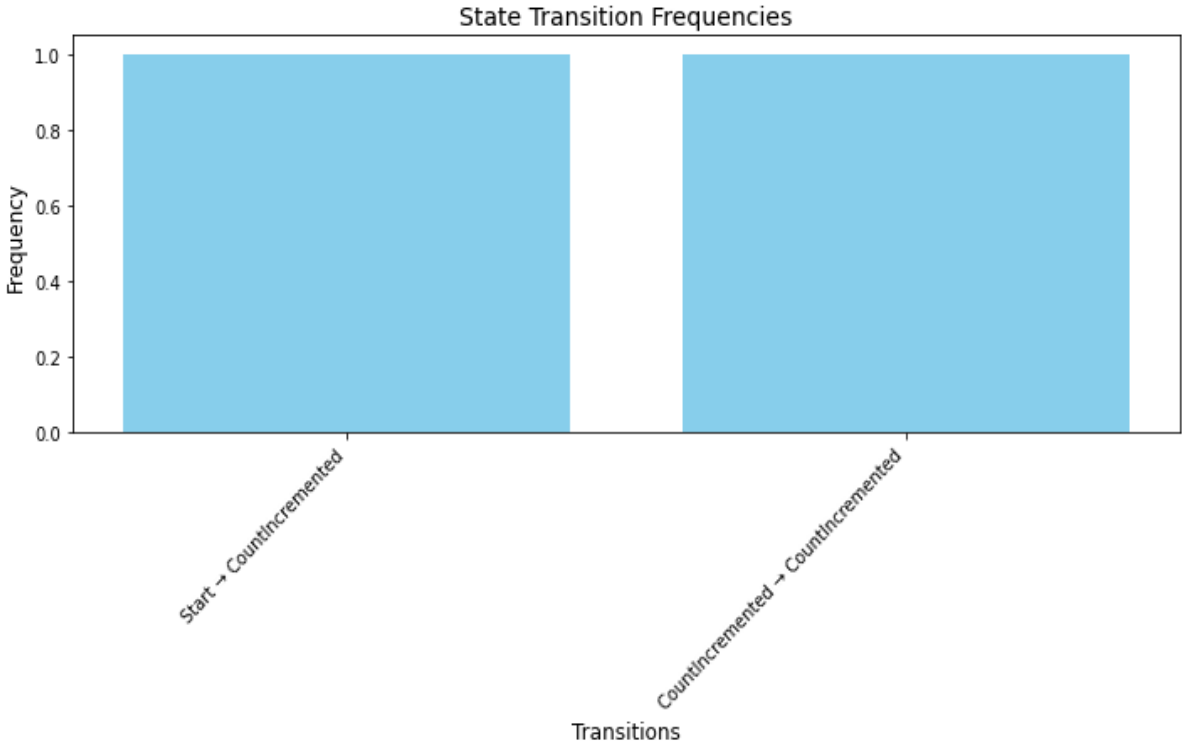


Figure 12: State transition frequencies diagram for integer overflow & underflow vulnerability

Reentrancy attack vulnerability

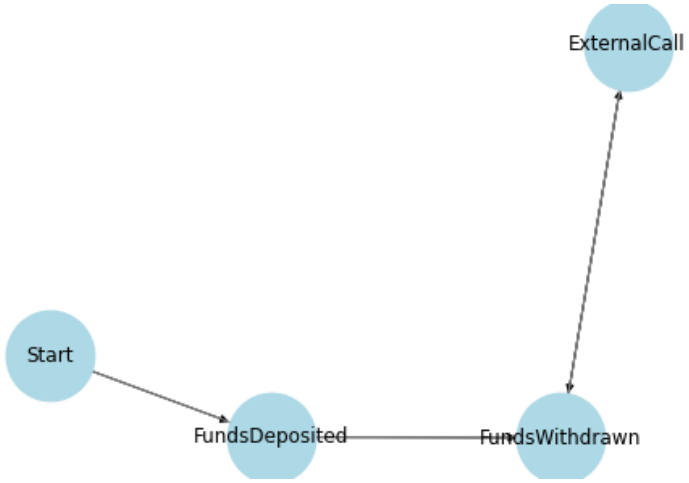


Figure 13: FSM diagram for re-entrancy attack vulnerability

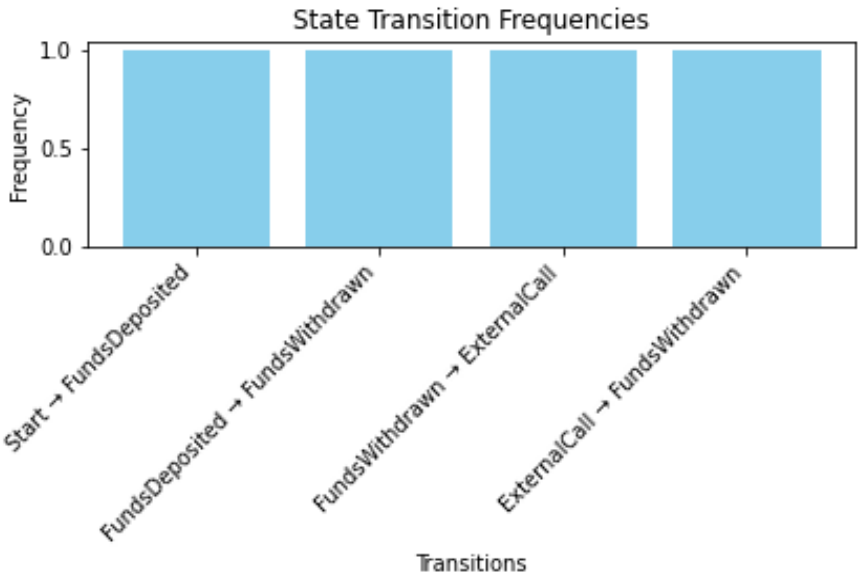


Figure 14: State transition frequencies diagram for reentrancy attack vulnerability

Tx.Origin authentication bypass vulnerability

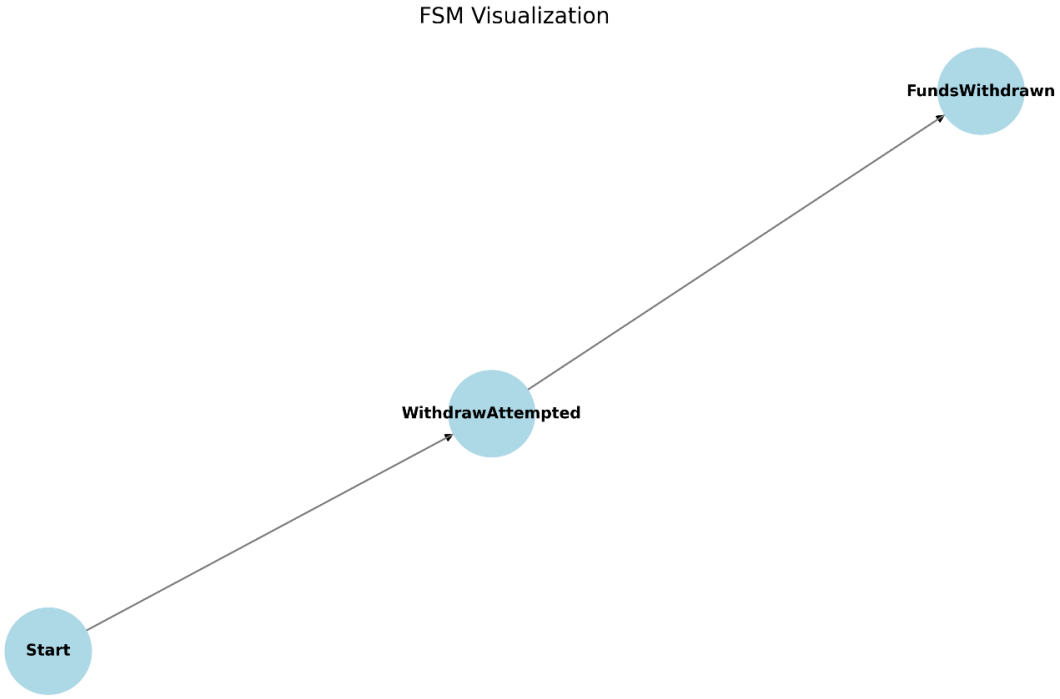


Figure 15: FSM diagram for tx. origin authentication bypass vulnerability

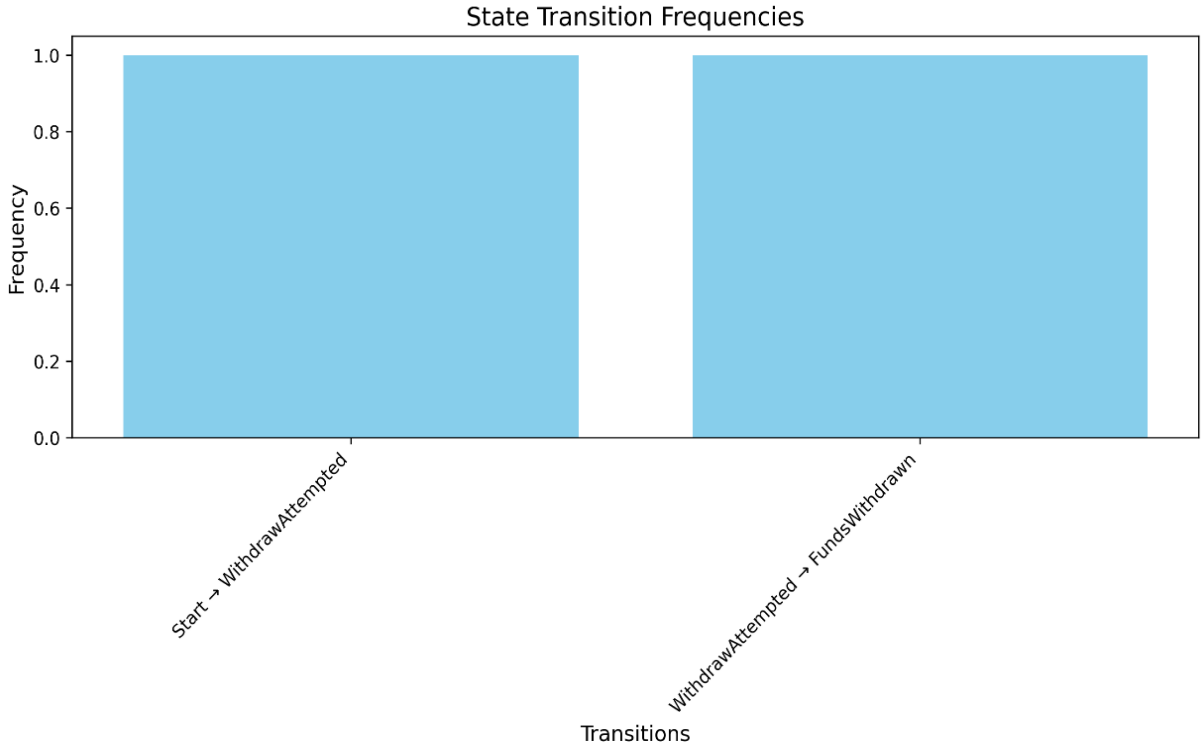


Figure 16: State transition frequencies diagram for tx.origin authentication bypass vulnerability

Uninitialized storage pointer vulnerability

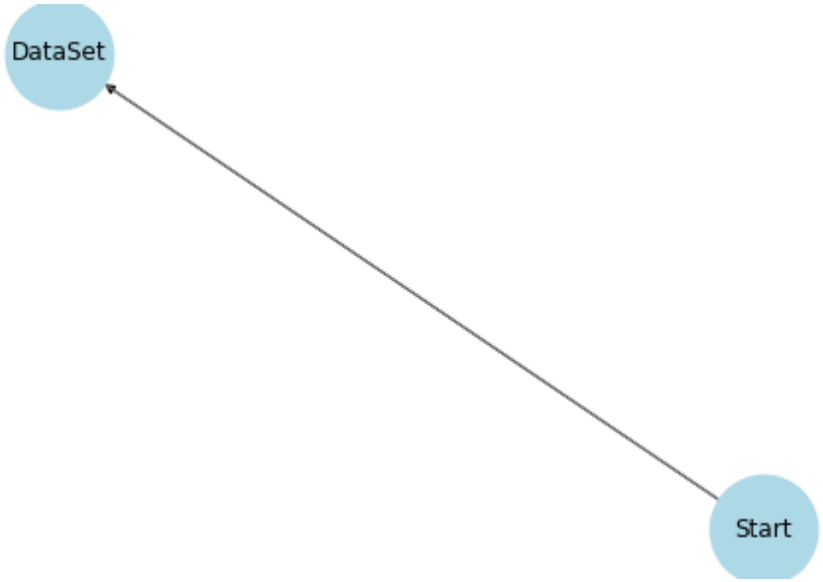


Figure 17: FSM Diagram for uninitialized storage pointer vulnerability

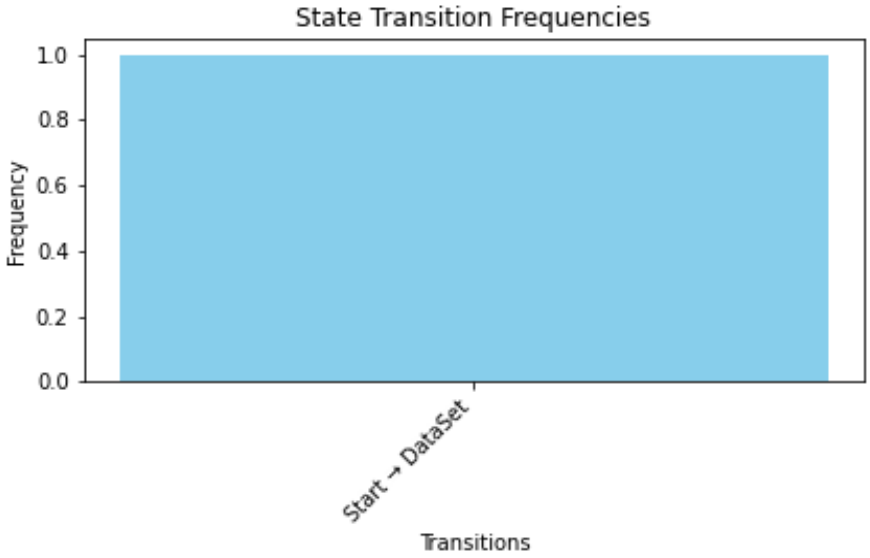


Figure 18: State transition frequencies diagram for uninitialized storage pointer vulnerability

Unprotected self-destruct vulnerability

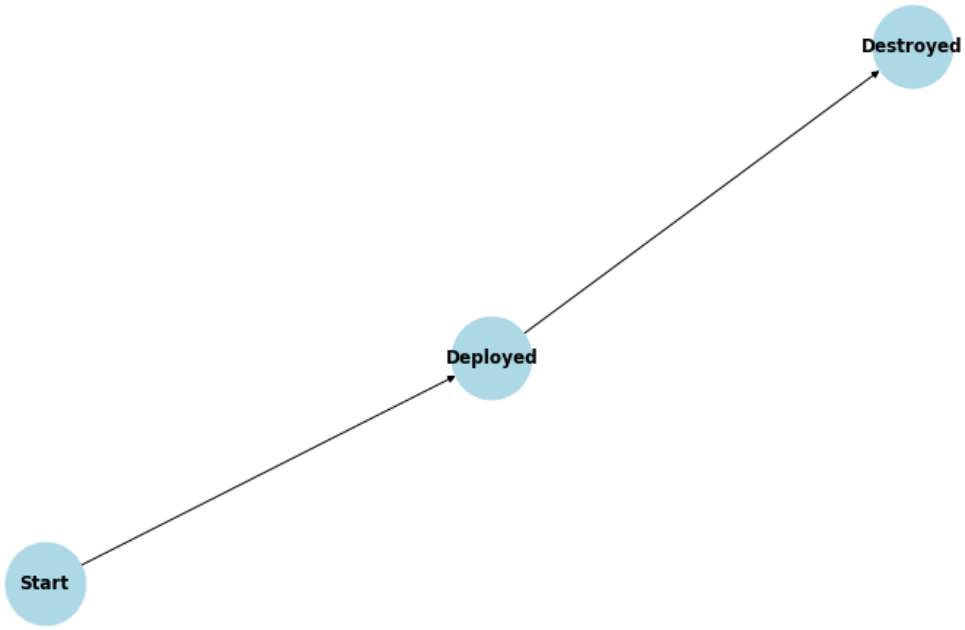


Figure 19: FSM Diagram for unprotected self-destruct vulnerability

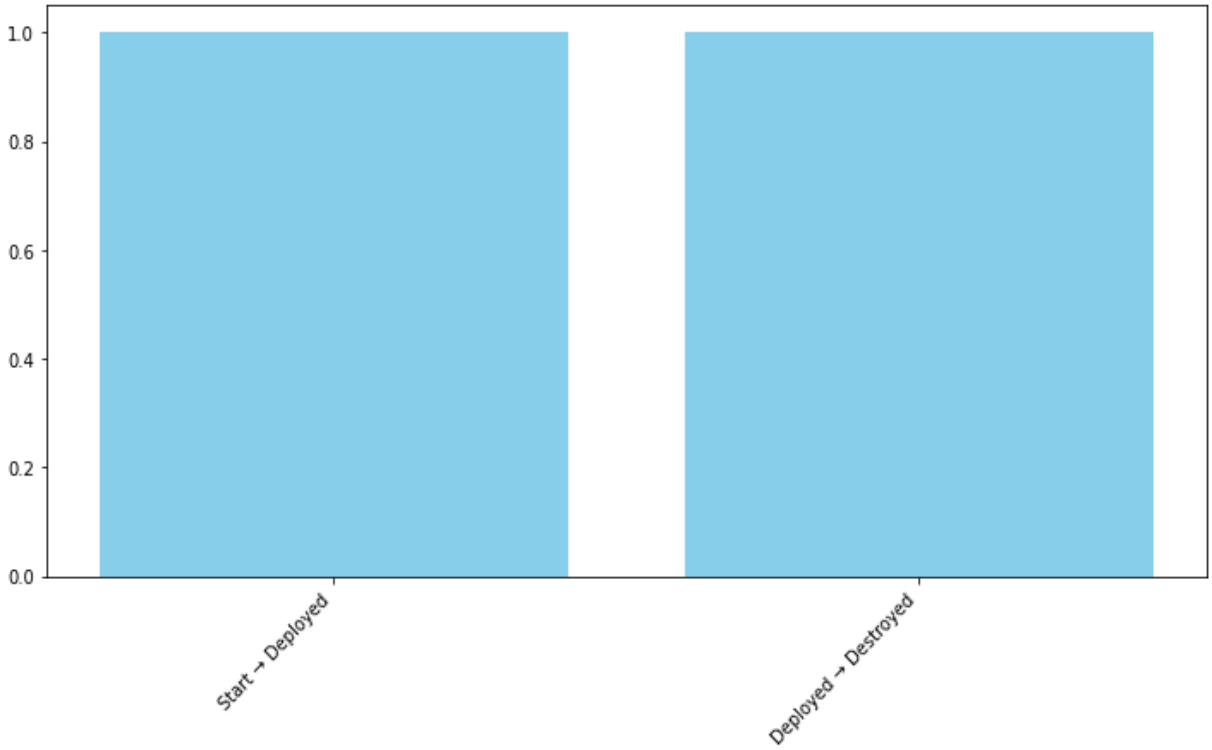


Figure 20: State transition frequencies diagram for unprotected self-destruct vulnerability

Short Address Attack Vulnerability

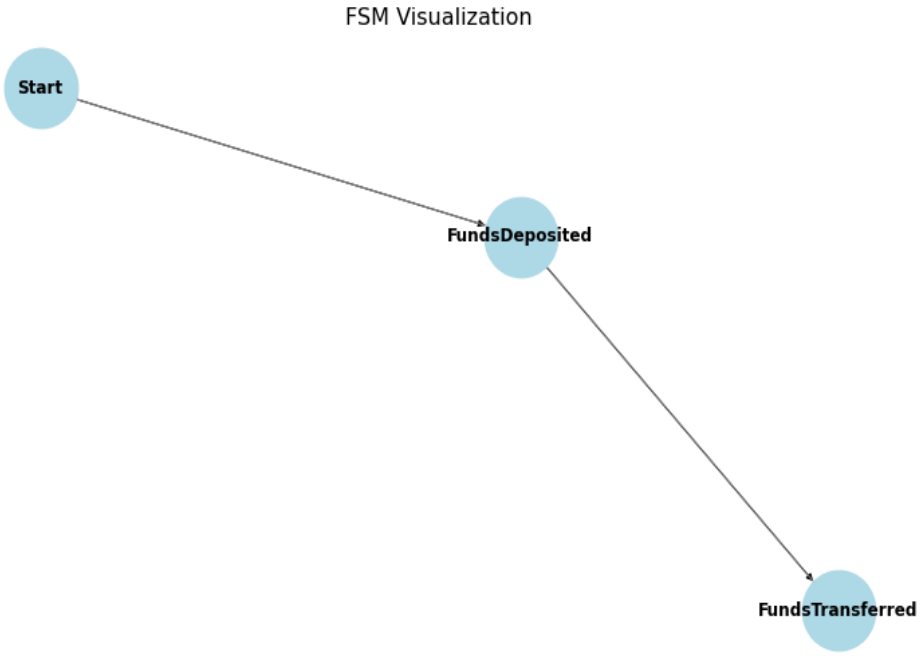


Figure 21: FSM diagram for short address attack vulnerability

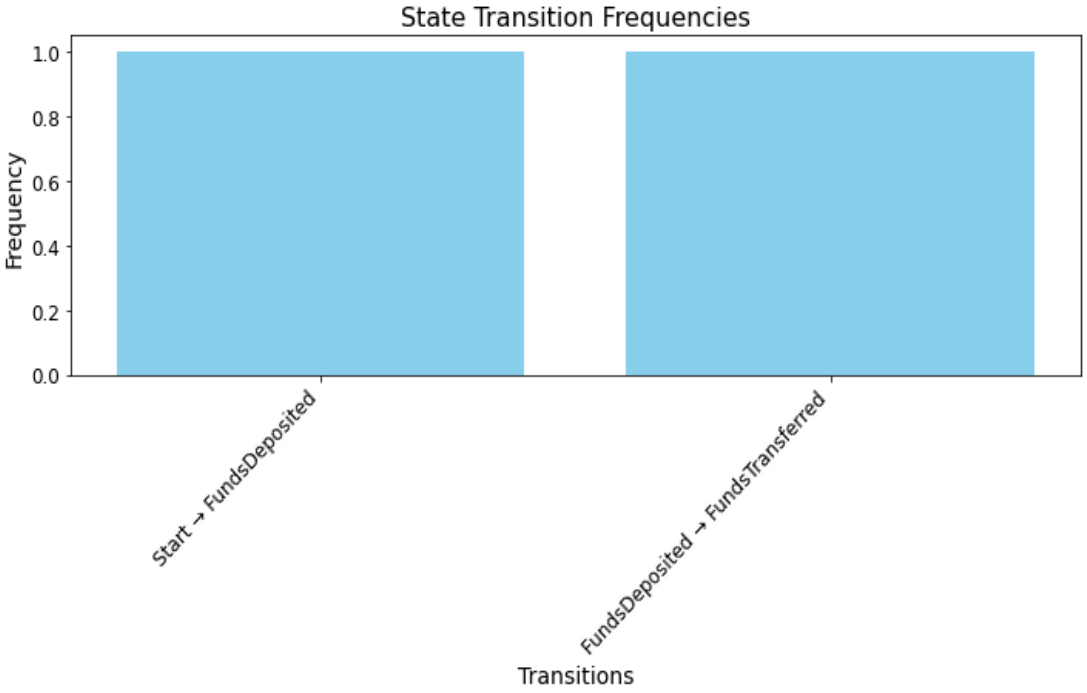


Figure 22: State transition frequencies diagram for short address attack vulnerability

Timestamp dependency vulnerability

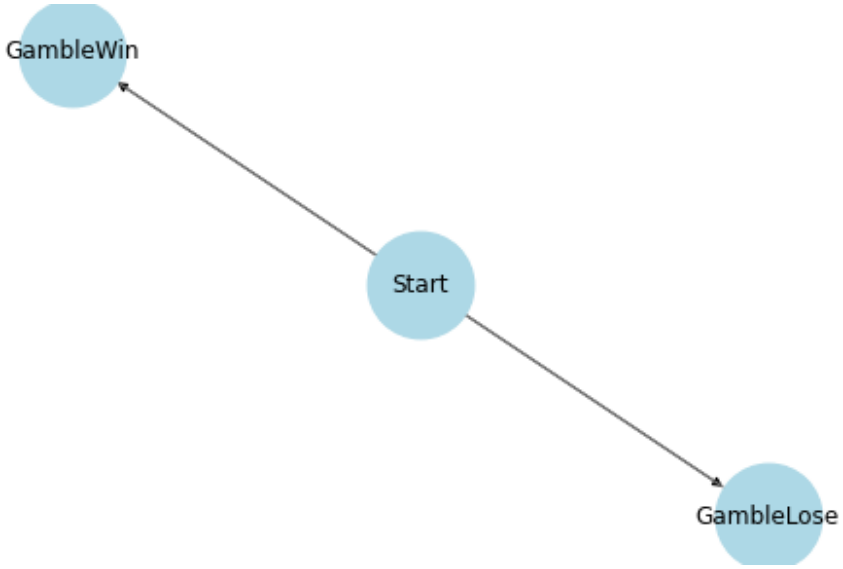


Figure 23: FSM diagram for timestamp dependency vulnerability

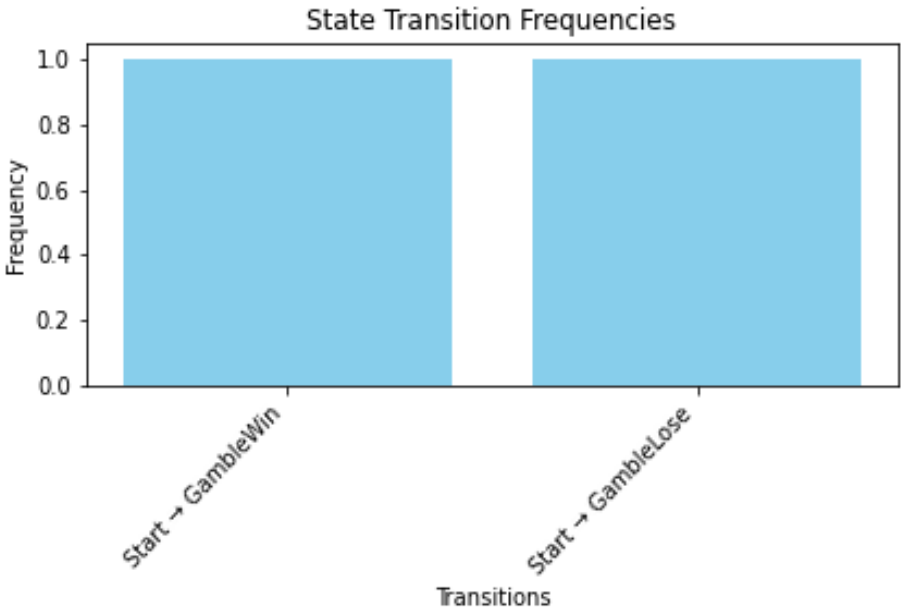


Figure 24: State transition frequencies diagram for timestamp dependency vulnerability

Block hash manipulation vulnerability

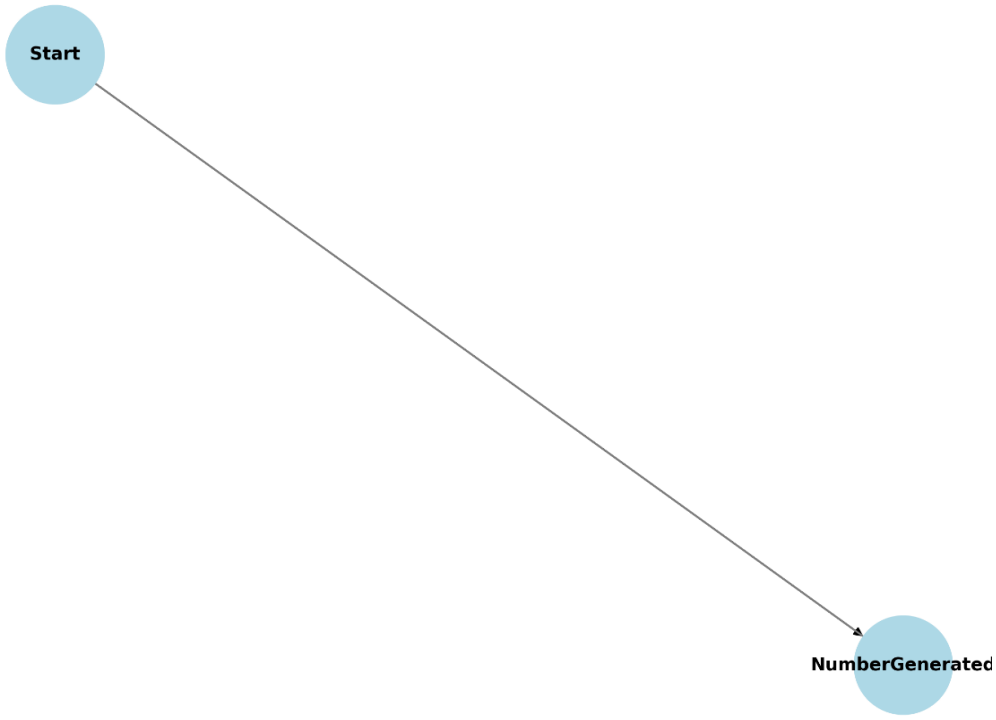


Figure 25: FSM diagram for block hash manipulation vulnerability

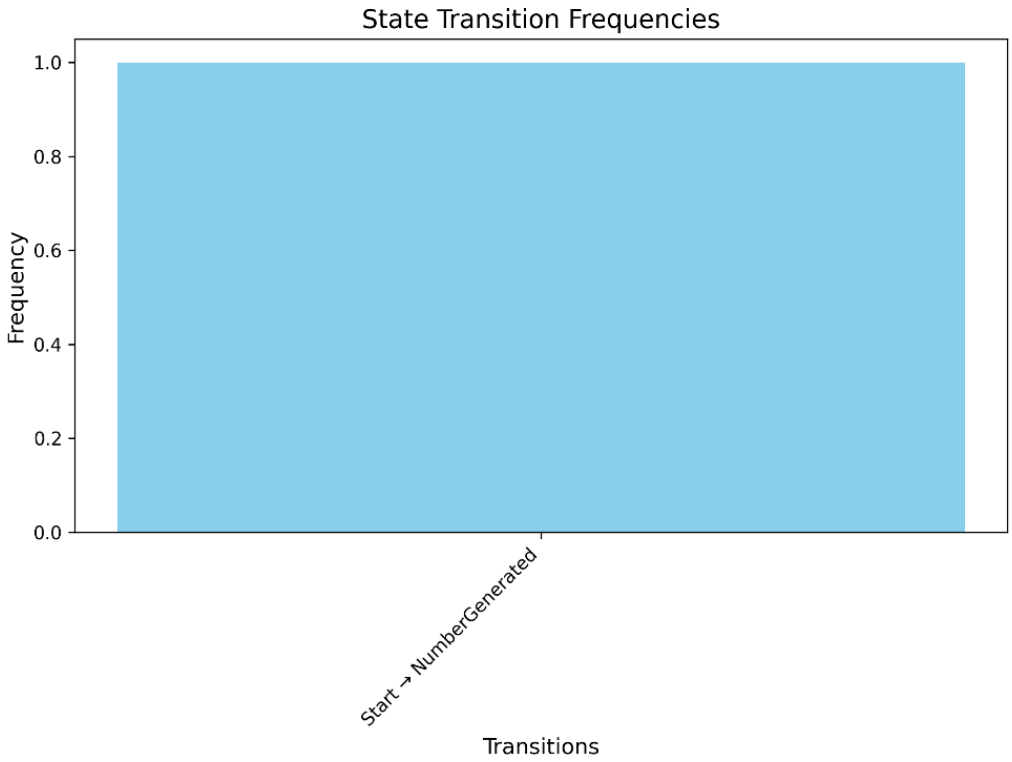


Figure 26: State transaction frequencies diagram for block hash manipulation vulnerability

Front-running attack vulnerability

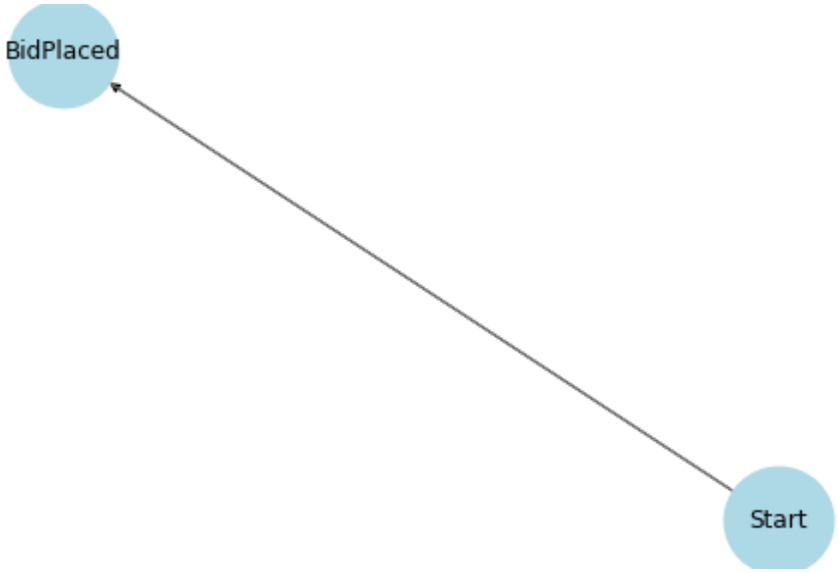


Figure 27: FSM diagram for front-running attack vulnerability

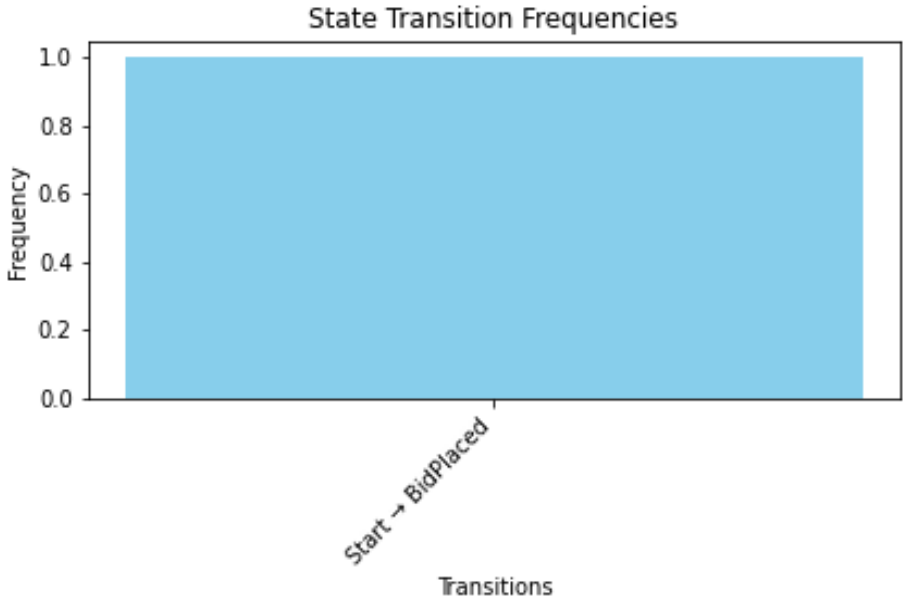


Figure 28: State transition frequencies diagram for front-running attack vulnerability

Weak randomness (predictable random numbers) vulnerability



Figure 29: FSM diagram for weak randomness (predictable random numbers) vulnerability

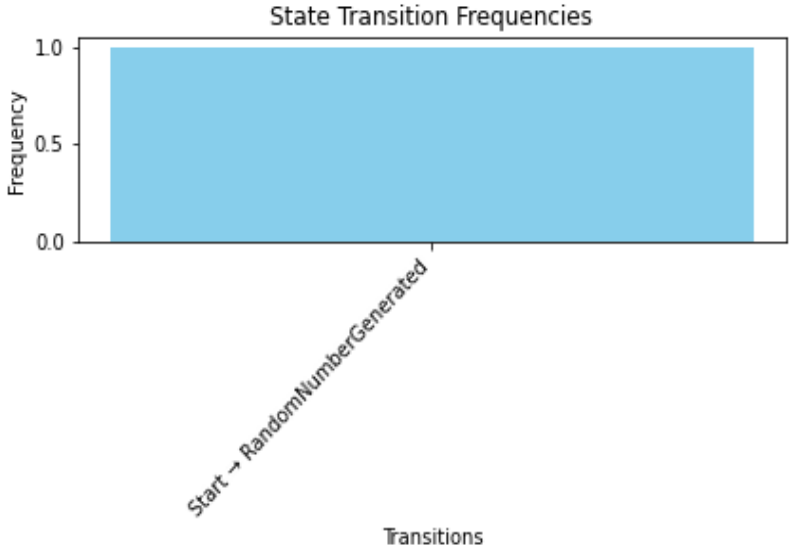


Figure 30: State transition frequencies diagram for weak randomness (predictable random numbers) vulnerability

Multivariate Ice-Covering Prediction on Transmission Lines Using CEEMDAN-SC-LSTM and CNN-GRU-Attention Models

Maojie Tian*, Huamin Sun, Weidong Liu, Shirui Sun, Chao Gao
State Grid Power Space Technology Co., Ltd, Beijing, 102211, China
E-mail: sg_tianmj@outlook.com
*Corresponding author

Keywords: LSTM, deep learning, high-altitude regions, power grid, transmission line, ice-covering

Received: September 8, 2025

In the power system, ice-covering on transmission lines in high-altitude areas can easily cause serious problems like increased line weight and wire breakage. To accurately predict the ice-covering and issue timely warnings to ensure the reliable operation, a method for predicting ice-covering in high-altitude areas based on Long Short-Term Memory (LSTM) and deep learning methods is proposed. In the feature extraction stage, a model for ice-covering feature extraction of transmission lines in cold regions based on CEEMDAN-SC-LSTM is proposed, combining spectral clustering, fully adaptive noise set empirical mode decomposition algorithm, and LSTM. This model can utilize the advantages of each algorithm to effectively decompose and cluster ice-covering data, extracting more representative features. A Multivariate Deep Regression (MDR) is built by integrating convolutional neural networks, gated recurrent units, and attention mechanisms, using the extracted features related to ice thickness changes as input. The experiment is based on a dataset collected from a specific main transmission line in a high-altitude mountainous area in western China from October 2024 to March 2025. The results showed that the prediction model had a coefficient of determination of 0.9887, an average absolute error of 0.0130, a root mean square error of 0.0142, and an average absolute percentage error of 1.2558%. Compared with baseline models such as support vector regression and random forest regression; the model can predict ice thickness more accurately. The early warning platform based on this method can effectively deliver ice-covering early warning, providing decision-making basis for power grid operation and maintenance. The warning platform based on this method can effectively deliver ice-covering warnings, providing decision-making basis for power grid operation and maintenance.

Povzetek: Študija predstavlja metodo globokega učenja za natančno napovedovanje in zgodnje opozarjanje na poledenitev visokonapetostnih daljnovodov, s čimer se izboljša zanesljivost obratovanja in vzdrževanja elektroenergetskega sistema.

1 Introduction

Ensuring the safe and stable operation of transmission lines in high-altitude regions is crucial for regional economic development and normal social functioning. However, the unique climate conditions in high-altitude regions make ice-covering on transmission lines a major threat to the operation [1]. The ice-covering phenomenon not only increases the weight of transmission lines, causing towers to deform or even collapse under excessive pressure, but also changes the electrical characteristics of the lines, leading to electrical faults such as flashover and short circuits. In severe cases, it can cause large-scale power outages and bring huge economic losses to society [2-3]. Driven by deep learning technology, it has shown great potential in time series prediction and feature extraction [4]. However, the time series data of ice-covering in high-altitude regions have characteristics such as nonlinearity, sudden changes, and uncertainty, as well as seasonal patterns, which make it difficult for traditional prediction methods to accurately capture complex patterns in the data. There are many factors that affect the ice-covering, such as climate, terrain, and line characteristics, which are intertwined and further increase the prediction difficulty [5-6].

To monitor images of overhead power lines covered in ice, Hao et al. recognized insulator and ice type images of overhead power lines based on deep weakly supervised learning and transfer learning, which reduced the interference of external factors. The precision, recall, and average accuracy reached 86.6%, 91.3%, and 90.1%, and the recognition speed reached 8ms/image [7]. The ice layer poses a threat to power systems, and the generated heat can prevent the growth of the ice layer. Therefore, Huang proposed a preventive scheduling model to alleviate frost and ice-covering phenomena by optimizing the power loss distribution on ice-covering transmission lines. The preventive scheduling model could avoid ice-covering during mild ice disasters, and effectively suppress the growth of ice-covering during

severe storms when complete deicing was not possible [8]. To deliver early warning of ice-covering and ensure safe operation, Zhang et al. introduced a mesoscale digital intelligent weather forecasting technology that considered the micro terrain to monitor and warn ice-covering on provincial power grid transmission lines. This method achieved significant results in achieving precise line inspection, enhancing the anti-ice-covering and disaster reduction capabilities [9]. To optimize the reliability of power supply in the distribution network, Yan et al. analyzed the current status of direct current deicing technology. A deicing system consisting of two sets of voltage source direct current deicing devices was proposed. The system had good effects on deicing and reactive power compensation, which could improve the power supply reliability [10]. Ice can reduce the electrical performance of insulators. To evaluate the ice-covering risk on insulators, Liang et al. built a method to automatically calculate the ice-covering parameters. The method could quantify the morphology and distribution of ice in the image, with a minimum processing time of 4.4s [11]. The summary of work is shown in Table 1.

Table 1: Summary of related work

Method	Dataset Type and Scale	Evaluation Metrics	Limitations	Reference
Deep Weakly Supervised Learning	Image data (quantity not specified)	Accuracy 86.6%, Recall 91.3%	Time-series characteristics not considered	Hao et al. [7]
Preventive Dispatch Model	Simulation data (scale not specified)	Inhibition effect on ice growth	Relies on power loss optimization	Huang [8]
Mesoscale Weather Forecast Technology	Provincial power grid monitoring data	Improved inspection accuracy and de-icing decision-making	Limited micro-terrain adaptability	Zhang et al. [9]
DC De-icing System	Experimental data	Improved power supply reliability	Ice prediction accuracy not quantified	Yan et al. [10]
Equal-Height Ridge Valley Analysis	Insulator image data	Single image processing time 4.4 seconds	Limited to morphological analysis, no prediction function	Liang et al. [11]
CEEMDAN-SC-LSTM+MDR	4320 time-series data points	R^2, MAE	High model complexity	Research Method

In summary, existing research in the field of predicting and monitoring ice-covering on transmission lines has achieved results, but there are still certain shortcomings. Firstly, insufficient consideration is given to the complex characteristics of ice-covering data, and the time series data is nonlinear, abrupt, and seasonal, making it difficult for traditional methods to accurately capture patterns. In addition, the consideration of influencing factors is not comprehensive, and multiple factors such as climate, terrain, and power line characteristics are intertwined. Existing research has not been fully integrated. In addition, feature extraction methods are not efficient and accurate enough to extract key features from multivariate data, and prediction models have limitations in handling long-term dependencies and automatically focusing on important features, resulting in poor prediction accuracy. To address these challenges, an innovative method for predicting ice-covering in high-altitude regions based on Long Short-Term Memory (LSTM) and deep learning techniques is proposed. In the feature extraction stage, considering the complexity and diversity of ice-covering data, a Complete Ensemble Empirical Mode Decomposition with Adaptive Noise - Spectral clustering - LSTM (CEEMDAN-SC-LSTM) ice-covering feature extraction model is built by combining SC, CEEMDAN algorithms, and LSTM. The model fully leverages the strengths of various algorithms. It effectively decomposes and clusters ice-covering data, thereby extracting more representative features. In terms of predictive model construction, combining CNN, GRU, and attention mechanism models, the extracted features related to changes in ice-covering thickness are used as inputs to construct a Multivariate Deep Regression (MDR) for predicting ice-covering based on deep learning and multivariate regression. Finally, based on this prediction method, a power grid line ice-covering warning platform suitable for high-altitude regions is designed based on data and models. The research aims to use this method to comprehensively and real-time monitor the ice-covering on transmission lines, and issue timely warnings when the risk of ice-covering first appears, thereby providing accurate and timely decision support for power grid operation and maintenance personnel. This study proposes the following core hypotheses: H1: The CEEMDAN algorithm outperforms standard EMD in decomposing multi-scale features of ice-covered data from high-altitude regions; H2: The MDR model integrating CNN-GRU-attention mechanism demonstrates significantly higher prediction accuracy than traditional models such as SVR and ANN.

2 Methods and materials

2.1 Construction of ice-covering feature extraction model integrating multiple algorithms

The traditional methods for predicting ice-covering on power grid lines are mainly based on meteorological parameters and simple physical models, which are difficult to accurately capture the dynamic process and inherent laws of ice-covering formation. The prediction accuracy and reliability are limited, and cannot satisfy the high requirements for safe and stable operation [12-14]. This study proposes a method for predicting ice-covering in high-altitude regions based on LSTM and deep learning techniques. However, the time series data of transmission line ice-covering presents nonlinearity, sudden changes, and uncertainty, and there are seasonal patterns, which increases the difficulty of analysis. In addition, there are many factors that affect ice-covering, including climate, terrain, and electrical characteristics. These factors are intertwined and jointly determine the severity and frequency of ice-covering. In addition, different types of ice-covering have different characteristics, such as rime, rime, etc. Accurately extracting their features requires comprehensively considering multiple factors, which further increases the complexity and challenge of feature extraction. To optimize the accuracy and efficiency, an ice-covering feature extraction model integrating multiple algorithms is developed by combining the SC algorithm, CEEMDAN algorithm, and LSTM network.

The research first collects various signals transmitted by transmission line sensors, and conducts detailed preprocessing to ensure data quality. Based on diverse real-time data resources in high-altitude regions, fixed data samples are carefully selected, and ice-covering feature extraction and accurate classification work are carried out. The collected transmission line data in high-altitude regions is normalized. The normalization is displayed in equation (1).

$$x_{new} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

In equation (1), x represents the data sample. Next, the CEEMDAN decomposes the preprocessed transmission line data signals in high-altitude regions into some Intrinsic Mode Functions (IMFs). Compared with Variational Mode Decomposition (VMD) and Empirical Wavelet Transform (EWT), CEEMDAN demonstrates unique advantages in processing nonlinear and non-stationary ice-covering time series data. By adaptively adding noise and performing multiple decompositions, CEEMDAN effectively resolves the mode mixing issue inherent in traditional Empirical Mode Decomposition (EMD), more accurately extracting multi-scale features from ice-covering data. In contrast, VMD and EWT exhibit greater parameter sensitivity when decomposing complex signals, and their performance becomes less stable and accurate when handling ice-covering data with abrupt transition characteristics. After analyzing the frequency, energy, and other characteristic parameters of these IMFs, the components most closely related to ice-covering characteristics are selected to achieve ice-covering recognition. When operating this algorithm, Gaussian white noise $n^i(t)$ is first added to signal $e(t)$ to generate experimental signal $e^i(t)$, as presented in equation (2).

$$e^i(t) = e(t) + n^i(t) \tag{2}$$

In the signal processing process, the EMD polarity decomposition method is uniformly applied to all obtained experimental signals. Based on this operation, the first-order mode decomposition signal I_1^i corresponding to each experimental signal can be obtained. Subsequently, the mean of these first-order mode decomposition signals is taken as the first-order mode component I_1 of the entire sequence, as shown in equation (3).

$$I_1 = \sum_{i=1}^{I} I_1^i \tag{3}$$

Finally, the predetermined steps are repeated to accurately calculate the modal components of each order until they cannot be further decomposed. The output results are input into the spectral clustering algorithm. This algorithm is used to construct a similarity matrix, and used as a basis to segment the data, thereby dividing different ice-covering state groups. When the spectral clustering algorithm starts running, it will deeply analyze the data and carefully construct a similarity matrix $\mathbf{W}_{ij}$ based on the similarity between each data point, as shown in equation (4).

$$\mathbf{W}_{ij} = \exp\left(-\frac{d^2\left(v_i, v_j\right)}{2\sigma^2}\right) \tag{4}$$

In equation (4), d signifies the distance between two sample points. v represents the data sample point. σ represents the scale parameter. $\mathbf{W}_{ij}$ is normalized to a Laplacian matrix, as shown in equation (5).

$$\begin{cases} L_{\text{nor}} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \\ L_{\text{nor}} = D^{-1} L = I - D^{-1} W \end{cases} \tag{5}$$

In equation (5), I represents the identity matrix. D represents a diagonal matrix. Next, the feature decomposition of the Laplacian matrix is performed to calculate the eigenvalues and eigenvectors, and the eigenvectors related to the minimum non-zero eigenvalue are selected to form a new matrix. Finally, each row of the new matrix is taken as a point in a high-dimensional space. After clustering these points, the data is classified based on spectral properties.

Finally, the study takes the feature results extracted by the CEEMDAN algorithm as input for LSTM, and uses LSTM to classify the ice-covering features of transmission lines, accurately discriminating their ice-covering conditions. LSTM combines the memory storage function of traditional models with new LSTM functions [15-16]. The LSTM model optimizes the

long-term dependency phenomenon of the RNN model through input gates, forget gates, etc. The "gate" is designed to control the memory state, storage time, and distance data, as presented in Figure 1.

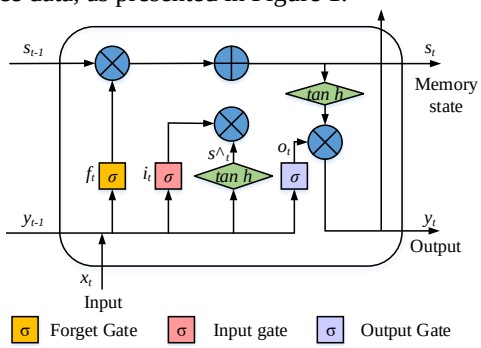


Figure 1: The LSTM structure

In Figure 1, the input gate i_t, output gate o_t, and forget gate f_t are presented in equation (6).

$$\begin{cases} i_t = \sigma\left(W_i \square\left[y_{t-1}, x_t\right] + b_i\right) \\ o_t = s\left(W_o \square\left[y_{t-1}, x_t\right] + b_o\right) \\ f_t = \sigma\left(W_f \square\left[y_{t-1}, x_t\right] + b_f\right) \end{cases} \tag{6}$$

In equation (6), W_i, W_o, and W_f signify the weight matrix of the corresponding "gate" at time t. b_i, b_o, and b_f signify the offset of the corresponding "gate" at t. x_t, y_t, and s_t respectively represent the input and output values at time t, as well as the memory state of the neuron. By training the LSTM network, the time series problem of ice-covering data in high-altitude regions can be effectively solved, accurately classifying ice-covering characteristics. Thus, the steps of the ice-covering feature extraction based on CEEMDAN-SC-LSTM are shown in Figure 2.

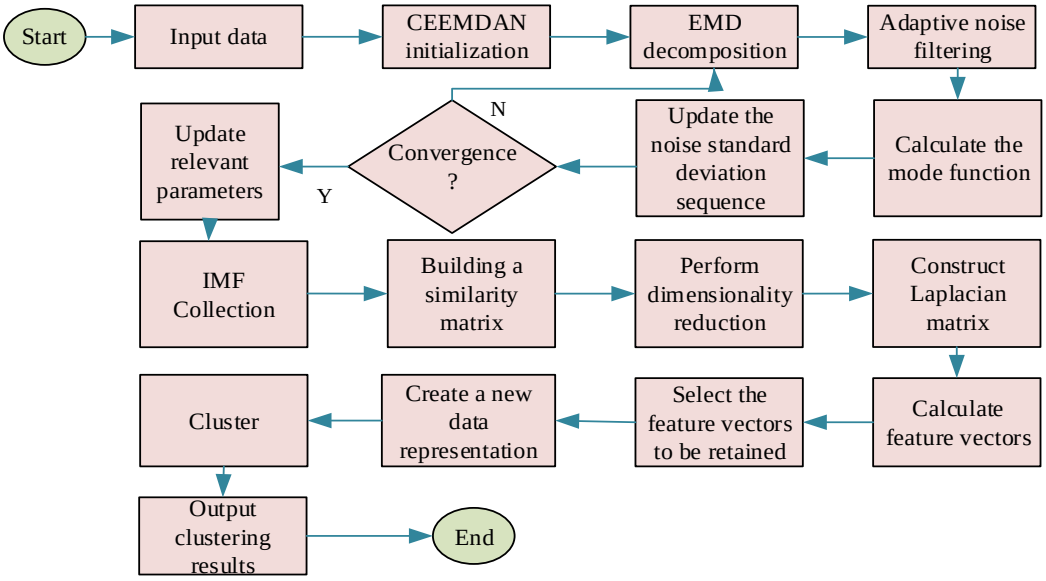


Figure 2: Flow chart of ice-covering feature extraction method for transmission lines based on CEEMDAN-SC-LSTM

In Figure 2, after inputting relevant data, the system first performs CEEMDAN initialization operation, followed by EMD decomposition. Subsequently, adaptive noise filtering is applied to the decomposed data to update the noise standard deviation sequence, and the steps of calculating the modulus function and constructing the Laplacian matrix are sequentially completed. If convergence is not achieved, the parameters are updated, IMF is collected, and a similarity matrix is constructed. If convergence occurs, a new data representation is created and clustered. and the results are output. The complete architecture specification of the CEEMDAN-SC-LSTM model is shown in Table 2.

Table 2: CEEMDAN-SC-LSTM model architecture specifications

Model Components	Specific parameters	Activation function
CEEMDAN	Number of trials: 50, noise width: 0.2	No (signal decomposition algorithm, no activation function concept)
sc	Similarity matrix calculation method: Gaussian kernel function, scale parameter: adaptive adjustment according to data	No (cluster algorithm, no activation function concept)

LSTM	Number of layers: 2, hidden layer sizes: 20,10, dropout rate: 0.2	Forgetting gate, input gate, and output gate: Sigmoid function; generating new candidate memory vectors and final output working memory: Tanh function

This study demonstrates multiple advantages of introducing the SC algorithm after applying the CEEMDAN algorithm to decompose transmission line data signals in high-altitude regions and obtain a series of IMFs. Firstly, while the decomposed IMFs from CEEMDAN contain abundant information, their varying correlation with ice coverage characteristics and inherent noise interference require specialized processing. The SC algorithm constructs a similarity matrix based on data point similarity, and then analyzes the frequency and energy characteristics of IMF to accurately distinguish ice related components and noise, forming different ice-covering state groups. Secondly, when applying LSTM for classification, the spectral clustering process enhances data clarity by creating well-defined cluster boundaries. This provides LSTM with more discriminative input features that can accurately learn different ice cover patterns, ultimately improving classification accuracy and efficiency.

2.2 Construction of ice-covering prediction model for transmission lines based on deep learning

After completing the feature extraction and classification, to accurately predict the ice-covering thickness, CNN, GRU, and attention mechanism models are combined. The extracted features related to changes in ice-covering thickness are used as inputs to the model, and an MDR model is constructed for predicting ice-covering. The CNN structure and GRU unit structure are shown in Figure 3.

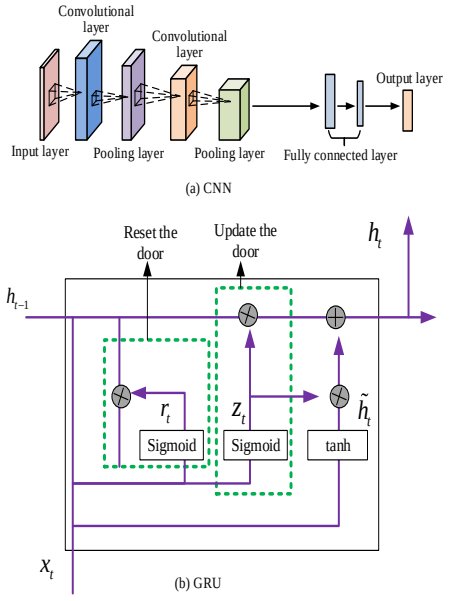


Figure 3: Schematic diagram of CNN and GRU

In Figure 3, the MDR first introduces the CNN to extract local spatial features from the input features. In the complex ice thickness prediction on transmission lines, traditional methods have certain limitations. CNN can effectively solve this challenge by converting meteorological data into image format, fully leveraging its own advantages, deeply extracting weather features, and significantly improving prediction accuracy [17-18]. Next, the GRU model is introduced to accurately input rich features extracted by CNN. With the unique mechanism of GRU, long-term dependencies hidden in time series data are deeply captured [19-20]. The description for GRU is presented in equation (7).

$$\begin{cases} z_t = \sigma\left(\mathbf{W}_z \cdot \left[\mathbf{h}_{t-1}, x_t\right]\right) \\ r_t = \sigma\left(\mathbf{W}_r \cdot \left[\mathbf{h}_{t-1}, x_t\right]\right) \\ \mathbf{h}_t = \tanh\left(\mathbf{W} \cdot \left[r_t \odot \mathbf{h}_{t-1}, x_t\right]\right) \\ \mathbf{h}_t = \left(1 - z_t\right) \odot \eta_{t-1} + z_t \odot \mathbf{h}_t \end{cases} \tag{7}$$

In equation (7), z_t signifies the update gate. $\mathbf{h}_t$ signifies the new unit vector. r_t denotes the reset gate. $\mathbf{h}_t$ signifies the hidden state vector. Each $\mathbf{W}$ represents its corresponding weight matrix. Finally, to enable the MDR to automatically emphasis the most important features for predicting ice-covering thickness and further improve model performance, attention mechanisms are specifically introduced and attention layers are carefully designed. This layer takes the output processed by GRU as input, calculates attention weights accurately through a specific algorithm, and then performs weighted average processing on the output of GRU based on these weights. Based on the above steps, Figure 4 presents the implementation process of the MDR.

In Figure 4, during the implementation of the MDR, the network structure is first established, and the network loss function and evaluation function lay the foundation for model training. Next, the ice-covering dataset of transmission lines is collected and decomposed using the CEEMDAN method to extract more detailed features. Subsequently, a transmission line feature dataset is constructed and spectral clustering is performed to form training and testing sets. The later trains the network. After continuously adjusting the network parameters, the MDR can better learn the patterns in the data. After training, the performance is assessed using a test set to prove its predictive accuracy. The entire process, from data preprocessing to model training and evaluation, has formed a complete implementation plan for the ice-covering prediction model.

Finally, based on the real-time monitoring data collected by the mature power grid ice-covering sensor network, the study realizes centralized management, real-time data

monitoring, and remote transmission through the Internet of Things cloud platform, ensuring the integrity and timeliness of the original data. In the data processing stage, the CEEMDAN-SC-LSTM is used to complete key feature extraction. During the model construction process, multiple factors like environmental temperature, humidity, and wind speed are comprehensively considered for dynamic correction, and the MDR predicts the ice-covering thickness. Finally, an intelligent early warning platform for power grid ice-covering suitable for high-altitude regions is designed. Figure 5 presents the overall architecture of the platform.

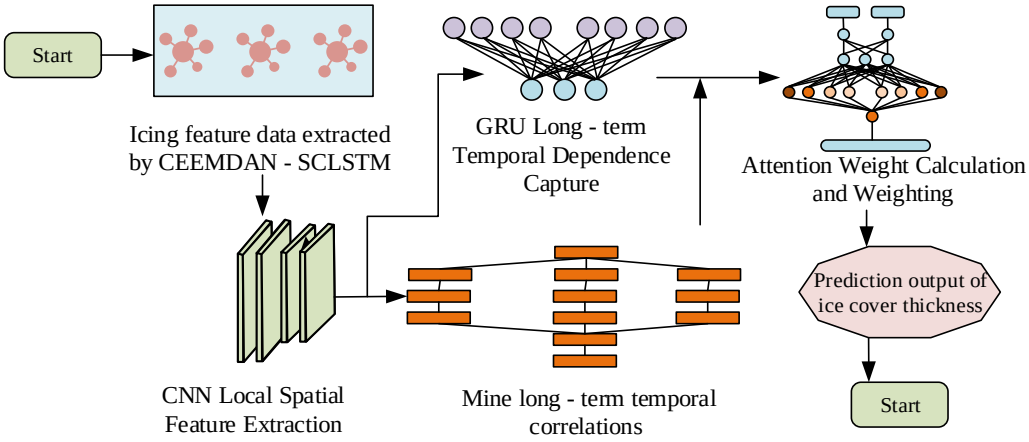


Figure 4: Implementation flowchart of MDR

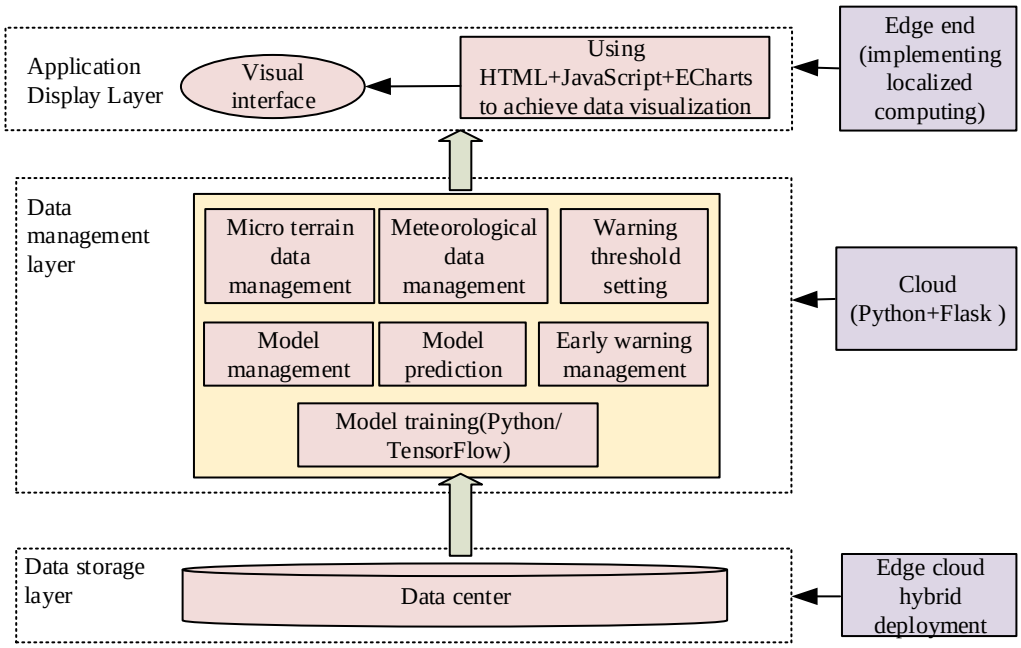


Figure 5: Schematic diagram of the ice-covering warning platform for power grid lines in high-altitude regions

In Figure 5, the platform is divided into four levels. The bottom layer is the data storage layer, composed of data centers, responsible for storing various types of data. Above it is the data management layer, which includes multiple modules such as micro terrain data management, meteorological data management, model management, model prediction, warning threshold setting, and early warning management. It also includes model training functions, which process and manage data. The upper layer is the application display layer, which visually displays the processed information to users through a visual interface. The collaborative work between various layers, from data storage to management and processing, and finally visualization presentation, jointly realizes the early warning function of power grid line ice-covering in high-altitude areas. The hybrid method for ice-covering prediction on transmission lines in high-altitude regions, integrating CEEMDAN-SC-LSTM and MDR models, is presented in pseudocode in Table 3.

Table 3: Pseudocode of ice coverage prediction method for transmission lines in high-altitude regions

Pseudo code for predicting icing on transmission lines in high-altitude regions by integrating CEEMDAN-SC-LSTM and MDR models
# CEEMDAN - SC - LSTM Feature Extraction Pseudocode
def CEEMDAN_SC_LSTM_feature_extraction(data):
# Data preprocessing
normalized_data = normalize(data)
# CEEMDAN decomposition
imfs = CEEMDAN(normalized_data)
# Select IMFs related to icing characteristics
selected_imfs = select_imfs(imfs)
# Spectral clustering
similarity_matrix = build_similarity_matrix(selected_imfs)
laplacian_matrix = build_laplacian_matrix(similarity_matrix)
eigenvalues, eigenvectors = eigen_decomposition(laplacian_matrix)
selected_eigenvectors = select_eigenvectors(eigenvalues, eigenvectors)
clusters = spectral_clustering(selected_eigenvectors)
# LSTM classification
lstm_model = build_lstm_model()
lstm_model.train(selected_imfs, clusters)
features = lstm_model.extract_features(selected_imfs)
return features
# MDR Prediction Model Pseudocode
def MDR_prediction_model(features):
# CNN feature extraction
cnn_output = cnn_feature_extraction(features)
# GRU capturing long-term dependencies
gru_output = gru_long_term_dependency(cnn_output)
# Attention mechanism
attention_weights = calculate_attention_weights(gru_output)
weighted_gru_output = apply_attention(gru_output, attention_weights)
# Predict ice thickness
predicted_thickness = predict_ice_thickness(weighted_gru_output)
return predicted_thickness

3 Results

3.1 Model performance testing

To verify the proposed ice-covering prediction method based on LSTM and deep learning methods, the study first needs to test the CEEMDAN-SC-LSTM and MDR. To optimize the accuracy and efficiency of ice-covering feature extraction for transmission lines, targeted testing is conducted on typical datasets. The dataset originates from a specific main transmission line in a high-altitude mountainous area in western China. The region has an average elevation of over 2,000m, with terrain dominated by mountains and plateaus, featuring significant elevation changes, crisscrossing mountain ranges, and deep valleys. The complex topography leads to significant local variations in meteorological conditions, creating diverse microclimate environments. In terms of meteorological conditions, the region experiences consistently low temperatures throughout the year, with particularly severe winters. The annual average temperature is between -5°C and 5°C, and the lowest temperature in winter can drop below -30°C. The humidity fluctuates greatly, ranging from 40% to 90%. Wind speeds also show distinct seasonal and regional variations, averaging 3-8 m/s annually, with speeds exceeding 15 m/s in special terrain areas like mountain passes and canyons. Precipitation is mainly concentrated in summer, although there is also significant snowfall in winter, with annual precipitation ranging from 400 to 800mm. These combined meteorological conditions result in frequent ice-covering on transmission lines in the region, providing rich data samples for research. The data collection spanned from October 2024 to March 2025, with hourly sampling frequency, yielding a total of 4,320 data points. The dataset is divided in an 8:1:1 ratio, with 3,456 training samples, 432 validation samples, and 432 test samples. This partition ensures sufficient training data for model learning while providing ample

validation and test samples to evaluate the model's generalization ability and predictive performance. The validation set is used to fine-tune hyperparameters and prevent overfitting, while the test set serves as the final evaluation benchmark for the model's performance on unseen data, ensuring its accuracy and reliability.
The operating environment and parameter settings for this experiment are presented in Table 4.

Table 4: Experimental operating environment

Experimental environment		Setting
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	3,000
Hyperparameter	CNN layers	2
	Number of CNN convolution kernels	32 and 64
	Number of GRU layers	2
	Number of GRU units	20 and 10
	Attention unit quantity	50
	LSTM	Number of layers: 2, hidden layer sizes: 20,10, dropout rate: 0.2

In this study, a grid search method was employed to determine hyperparameters for CNN, GRU, and attention mechanism models. The process involved three steps: Firstly, defining the hyperparameter search range; Secondly, constructing a parameter grid; Finally, selecting the hyperparameter configuration with the lowest loss and highest accuracy based on validation set evaluation results. The hyperparameter tuning results are presented in Table 5.

Table 5: Hyperparameter tuning experiment results

Hyperparameter	CNN layers	CNN convolution kernels	GRU layers	GRU units	Attention unit quantity	Learning rate	Batch size	Accuracy	Loss value
Combination 1	1	1,632	1	105	30	le-5	4	0.856	0.213
Combination 2	1	1,632	1	105	30	le-5	8	0.872	0.198
Combination 3	1	1,632	1	105	30	le-5	16	0.865	0.205
Combination 4	1	1,632	1	2,010	40	le-4	4	0.881	0.189
Combination 5	1	1,632	1	2,010	40	le-4	8	0.895	0.176
Combination 6	1	1,632	1	2,010	40	le-4	16	0.889	0.182
Combination 7	2	3,264	2	105	50	le-5	4	0.902	0.168
Combination 8	2	3,264	2	105	50	le-5	8	0.925	0.145
Combination 9	2	3,264	2	105	50	le-5	16	0.918	0.152
Combination 10	2	3,264	2	2,010	50	le-4	4	0.936	0.138
Combination 11	2	3,264	2	2,010	50	le-4	8	0.952	0.121
Combination 12	2	3,264	2	2,010	50	le-4	16	0.947	0.128

As shown in Table 5, after comprehensively evaluating the model performance across different hyperparameter configurations, the final configuration (2-layer CNN with 32 and 64 convolution kernels; 2-layer GRU with 20 and 10 units; 50 attention units; learning rate of 1e-4; batch size of 8) achieved the lowest loss and highest accuracy. This configuration delivered optimal performance on the validation set, providing more accurate predictions for ice-covering on transmission lines in high-cold regions.

After preprocessing the collected data, the MDR is trained on the self-built dataset, as presented in Figure 6. In Figure 6 (a), the loss of the MDR decreased with the iteration, from 0.1800 to 0.1084 in the training set and from 0.1362 to 0.0915 in the validation set, improving generalization ability. From Figure 6 (b), the MDR model had an accuracy of over 90% on both sets. The final accuracy was 96.93% and 98.21%. The MDR performs well in training and can be used for predicting ice-covering. To address the overfitting risks in the MDR model, the study implemented multiple mitigation strategies. Firstly, data normalization was applied during preprocessing to reduce fluctuations and improve convergence, effectively preventing overfitting on specific features. Secondly, regularization techniques were employed during training to constrain parameter magnitudes and avoid excessive weight growth. Hyperparameter tuning, including learning rate adjustments and batch size optimization, was conducted to ensure stable training processes, while cross-validation was utilized to evaluate model generalization capabilities. By partitioning datasets into training and validation sets, the study comprehensively assessed model performance across different datasets, promptly identifying overfitting risks. Additionally, an early termination mechanism was implemented: When the validation loss stabilizes after a specified number of iterations, the training stops, effectively preventing overfitting. The SC is chosen as the comparative method in the experiment. Modal decomposition and feature extraction are performed on the same data, as displayed in Figure 7.

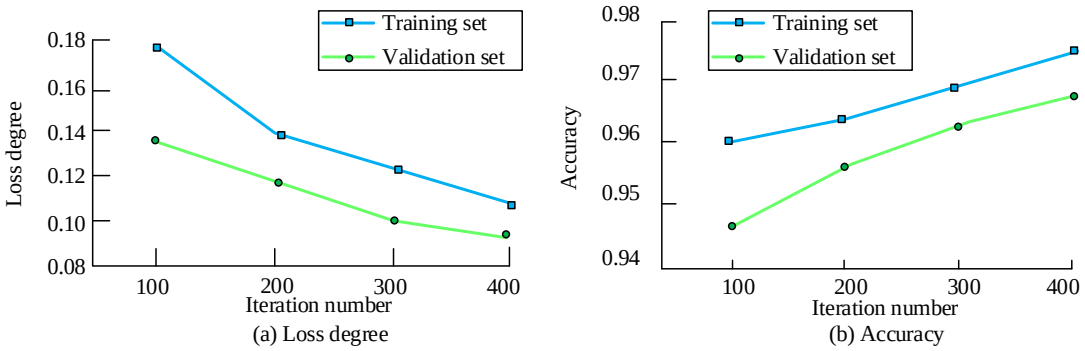


Figure 6: Training results of MDR model

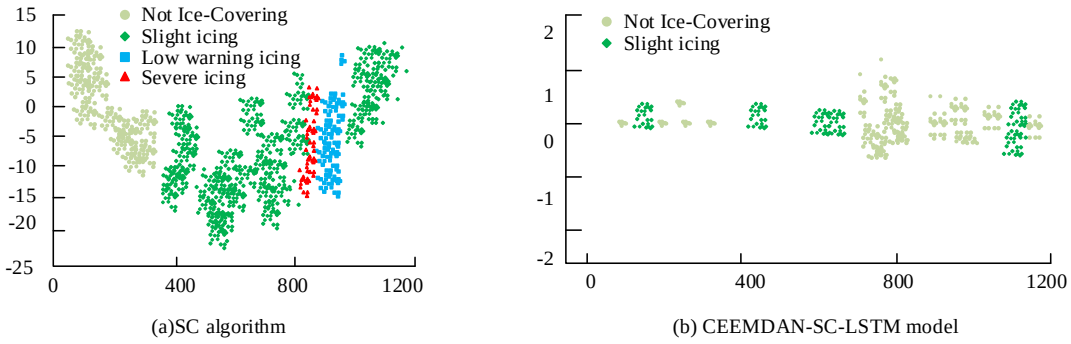


Figure 7: Comparison of feature extraction results of CEEMDAN-SC-LSTM model

Figure 7 (a) shows the raw data classification using the SC algorithm, and Figure 7 (b) displays the classification results of the CEEMDAN-SC-LSTM. From Figure 7 (a), when relying solely on spectral clustering for the extraction and classification of ice-covering features, the distribution of data points was relatively mixed. The data points of non-ice-covering, slight ice-covering, and low warning ice-covering states were intertwined, making it difficult to form clear and distinguishable clustering boundaries, resulting in difficulty in accurately identifying these degrees of ice-covering. However, for severe ice-covering, spectral clustering lacks effective mining of multi-scale features when processing complex time-series ice-covering data, resulting in large and inaccurate warning time judgments and inability to timely and accurately identify severe ice-covering risks. From Figure 7 (b), the CEEMDAN algorithm first performs adaptive decomposition on the ice-covering data, mines multi-scale features, and combines the clustering advantages of SC with the learning ability of LSTM for time-series data. The clustering of data points in different ice-covering states is clear, the recognition accuracy is significantly improved, and the extracted features are richer and more comprehensive, accurately extracting and classify ice-covering features. This demonstrates the superiority of the model in ice-covering feature processing. The test results of DBSCAN, hierarchical clustering, SC algorithm, and the CEEMDAN-SC-LSTM model for the silhouette

coefficient, Davies-Boulding index, Calinski-Harabasz index, and Xie-Beni index are shown in Table 6.

Table 6: Quantitative performance evaluation results of the CEEMDAN-SC-LSTM model

Model	SC	CEEMDAN-SC-LSTM	DBSCAN	Hierarchical clustering
Silhouette coefficient	0.32	0.72	0.45	0.52
Davies-Bouldin Index	2.31	0.87	1.89	1.67
Calinski-Harabasz	45.78	120.65	78.32	92.15
Xie-Beni	0.48	0.21	0.35	0.3

As shown in Table 6, the CEEMDAN-SC-LSTM model outperformed all comparison algorithms across all metrics. Specifically, compared with the SC algorithm, the silhouette coefficient increased from 0.32 to 0.72, indicating significantly improved cohesion and separation in clustering results. The Davies-Boulding index decreased from 2.31 to 0.87, demonstrating more compact and well-separated clusters. The Calinski-Harabasz index rose from 45.78 to 120.65, further validating the enhanced clustering quality. These quantitative metrics demonstrate that the SC algorithm achieves remarkable clustering improvements when combined with CEEMDAN and LSTM. The ablation results of the CEEMDAN-SC-LSTM model are presented in Table 7.

Table 7: Ablation results of the CEEMDAN-SC-LSTM model

Model	Coefficient of determination (R^2)	MAE	RMSE	MAPE	Classification accuracy
CEEMDAN-SC-LSTM	0.9887	0.0130	0.0142	1.2558%	97.1%
SC-LSTM	0.9423	0.0321	0.0356	3.1245%	92.5%
CEEMDAN-LSTM	0.9567	0.0289	0.0312	2.8765%	93.8%

As shown in Table 7, the CEEMDAN-SC-LSTM model outperformed all other models across all metrics. When CEEMDAN and LSTM components were removed, the classification accuracy dropped from 97.1% to 92.5% and 93.8%, respectively. The results demonstrate that the optimal data separability is achieved through their synergistic interaction.

Then, the CEEMDAN-SC-BP, CEEMDAN-SCCNN, CEEMDAN-SCSVM, and CEEMDAN-SC-LSTM are used to classify the raw data (divided into no ice-covering, slight ice-covering, and low warning ice-covering areas), as shown in Figure 8.

Figure 8 displayed the ice-covering classification results of the CEEMDAN-SC-BP, CEEMDAN-SCNN, CEEMDAN-SC-SVM, and CEEMDAN-SC-LSTM. On the self-built dataset, the CEEMDAN-SC-BP model had the worst classification performance. Compared with the basic CEEMDAN-SC-SVM model, the CEEMDAN-SCNN and CEEMDAN-SC-SVM models partially improved their classification performance, with 11 misclassifications. The CEEMDAN-SC-LSTM had the best classification performance, with only 6 misclassifications and a classification accuracy of around 97.1%. Under the same experimental conditions, the MDR monitors the ice-covering situation from January to March 2025, as displayed in Table 8.

From Table 8, the MDR prediction model had good predictive performance. The accuracy was all above 96.5%, with the highest reaching 98.1%, demonstrating a high consistency between the predicted results and the actual situation. The R^2 was all close to 0.99, indicating that the model had strong explanatory power for ice-covering thickness. The maximum Mean Absolute Error (MAE) was 0.7mm and the minimum was only 0.2mm, demonstrating that the average error was small. The maximum MAPE was 4.00% and the minimum was 1.82%, reflecting a relatively small prediction error and high prediction accuracy. This further validates the effectiveness and reliability for predicting ice-covering. Finally, on the test set of the self-built dataset, the MDR is compared with the comparison models Support Vector Regression (SVR), Random Forest Regression (RFR), Gradient Boosting Regression Tree (GBRT), and Artificial Neural Network (ANN), as presented in Figure 9.

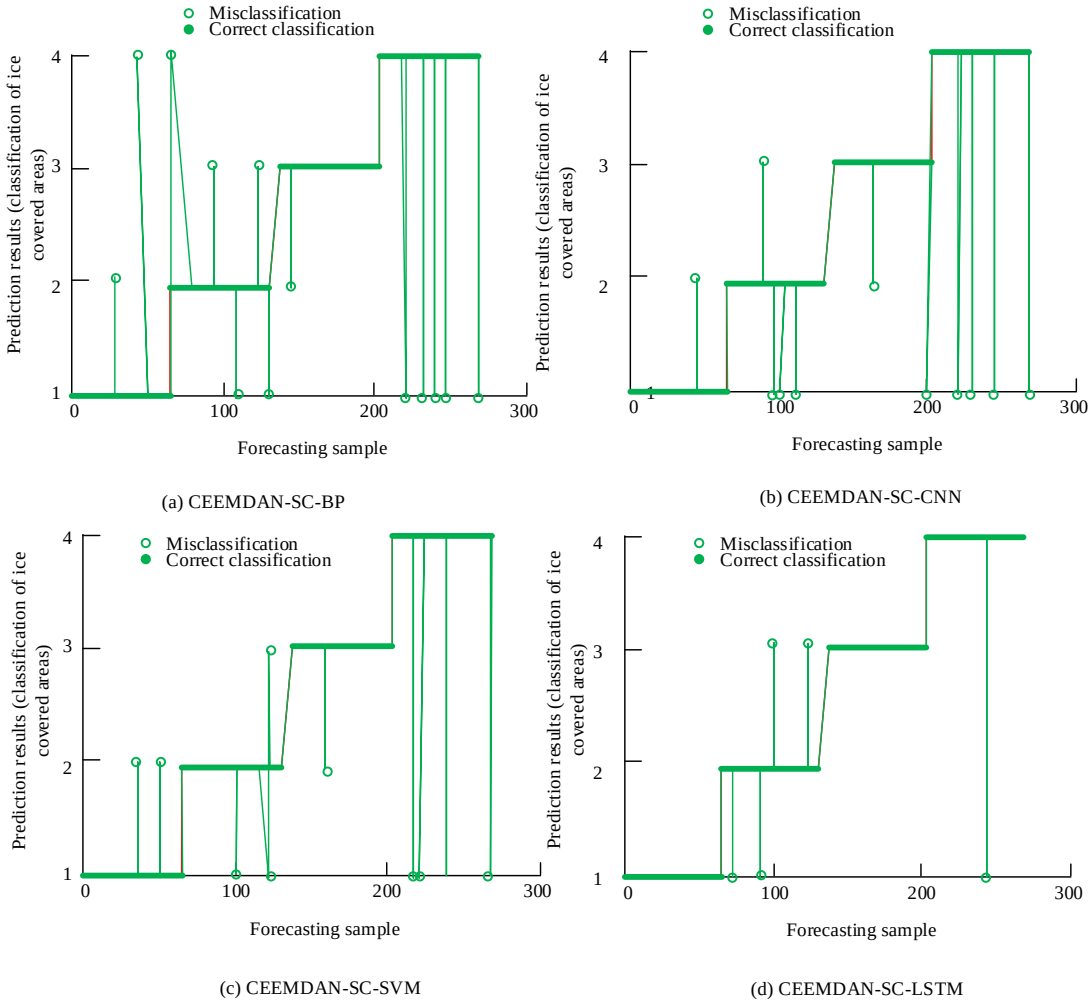


Figure 8: Comparison of classification results of ice-covering areas using different classification models

Table 8: Monitoring results of ice-covering from January to March 2025

Monitoring time	Actual ice thickness (mm)	Predicted ice thickness (mm)	MAE (mm)	MAPE (%)	Accuracy rate (%)	R^2	Variance	95% confidence interval
2025-01-10	10	9.6	0.4	4.00	96.5	0.9852	0.002	[9.55,9.65]
2025-01-20	15	14.5	0.5	3.33	97.2	0.9878	0.003	[14.45,14.55]
2025-02-05	20	19.3	0.7	3.50	97.8	0.9891	0.004	[19.25,19.35]
2025-02-15	22	21.6	0.4	1.82	98.1	0.9903	0.002	[21.55,21.65]
2025-03-05	8	7.8	0.2	2.50	96.9	0.9865	0.001	[7.75,7.85]

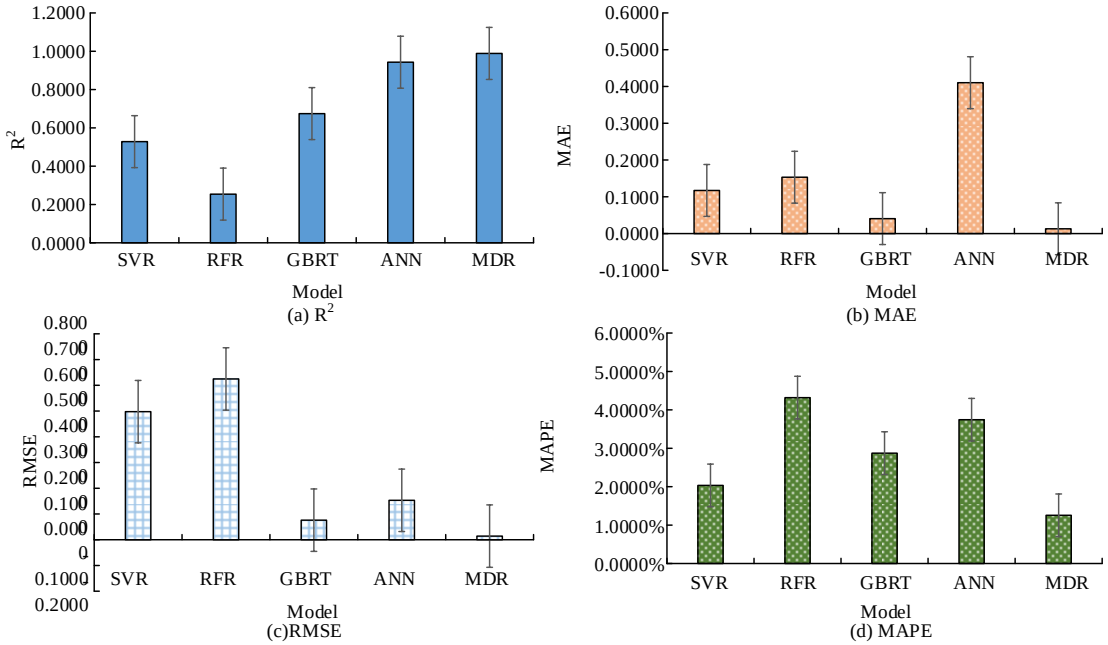


Figure 9: Comparison of MDR prediction models

Figure 9 shows the performance of MDR, SVR, RFR, GBRT, and ANN models in ice-covering prediction. From Figure 9 (a), the R^2 of the MDR model reached 0.9887, close to 1, indicating its strong explanatory power for ice-covering thickness, far higher than that of SVR (0.5281), RFR (0.2541), GBRT (0.6744), and ANN (0.9429). From Figure 9 (b), the MAE of the MDR model was 0.0130, which was the smallest among all models, indicating that the average error was the smallest. From Figure 9 (c), the MDR model was 0.0142, which also performed the best, reflecting its high prediction accuracy. From Figure 9 (d), in the MAPE index, the MDR model was 1.2558%, outperforming other comparison models. Overall, the MDR model performs well in all indicators and can more accurately and stably predict the ice-covering thickness of transmission lines in high-altitude regions compared to other models. The test results of the MDR classification warning model and deep baseline models (LSTM-only, GRU-only, CNN-LSTM, and Transformer) in terms of accuracy, recall, F1 score, and overall accuracy derived from the confusion matrix are shown in Table 9.

Table 9: Performance comparison between MDR model and deep baseline model

Model	Precision (%)	Recall (%)	F1 (%)	Overall accuracy (%)
MDR	97.8	97.2	97.5	97.3
LSTM	91.5	90.8	91.1	90.9
GRU	92.3	91.6	91.9	91.7
CNN-LSTM	95.6	95.1	95.3	95.2
Transformer	96.4	95.9	96.1	96.0

As shown in Table 9, the MDR model demonstrated superior performance in ice-covering classification tasks, achieving 97.8% accuracy, 97.2% recall, 97.5% F1 score, and 97.3% overall accuracy, significantly outperforming all deep learning baselines. The LSTM-only and GRU-only models performed the worst due to their lack of spatial feature extraction capabilities and key feature focusing mechanisms, with overall accuracy below 92%, particularly showing higher errors in borderline classifications such as "slight icing" versus "low warning icing." Although the CNN-LSTM model incorporates spatial features through CNN, its classification metrics remain over 2% lower than MDR, indicating that the attention mechanism prevents precise focus on critical icing features. While the Transformer model captures long-term temporal correlations, it underperforms in identifying local features of icing data (such as abrupt icing level transitions caused by micro-meteorological changes), resulting in 1.4% lower than that of MDR. This validates the architecture rationality of the MDR model, and its high performance ensures the accuracy of ice warning.

The generalization ability test results of the MDR model in specific regions are shown in Table 10.

Table 10: The generalization ability test results of MDR model in specific regions

Area	Season	R^2	MAE	MAPE
Eastern Plains	Winter	0.9234	0.0321	2.8765%
Southern hilly area	Winter	0.9012	0.0356	3.1245%
Northern mountainous area (non-original study area)	Winter	0.8876	0.0389	3.4567%

As shown in Table 10, the proposed model demonstrated adaptability across different regions and seasons, although its performance showed some decline. In the eastern plains and southern hilly areas, the model maintained an R^2 value above 0.9 with relatively low MAE and MAPE, indicating good predictive performance in these regions. However, in the northern mountainous area (non-original study region), the R^2 value dropped to 0.8876 while MAE and MAPE increased. This may be attributed to significant differences in terrain and climate conditions between the northern mountainous area and the original study region, which affect the model's prediction accuracy. Future work may involve collecting more regional and seasonal data to optimize and improve the model, thereby enhancing its generalization ability. The results of the MDR model ablation experiment are shown in Table 11.

Table 11: Results of MDR model ablation experiments

Model	R^2	MAE	RMSE	MAPE
MDR	0.9887	0.0130	0.0142	1.2558%
Remove CNN	0.9654	0.0215	0.0236	2.1548%
Remove GRU	0.9589	0.0248	0.0272	2.4876%
Remove attention mechanism	0.9732	0.0187	0.0203	1.8765%

As shown in Table 11, CNN, GRU, and the attention mechanism all play crucial roles in the MDR model. After removing the CNN, the R^2 was reduced to 0.9654 while increasing MAE, RMSE, and MAPE, demonstrating that the local spatial features extracted by CNN significantly contribute to ice thickness prediction. Eliminating the GRU further degraded performance, with R^2 dropping to 0.9589, indicating that the GRU's ability to capture long-term temporal dependencies substantially impacts prediction accuracy. Removing the attention mechanism can also lead to performance degradation, indicating that this component enables the model to automatically focus on key features, thereby improving prediction accuracy.

The ice-covering prediction model developed in this study, which integrates LSTM and deep learning techniques, demonstrates high accuracy through three key factors. Firstly, the dataset originates from transmission line corridors in a high-altitude mountainous region, where complex terrain and diverse meteorological conditions ensure rich diversity and representativeness, comprehensively capturing various ice-covering scenarios. Secondly, the CEEMDAN-SC-LSTM model effectively leverages the strengths of each component: CEEMDAN algorithm decomposes multi-scale features in ice data, SC algorithm performs precise clustering, while LSTM network processes temporal patterns to identify time-varying characteristics. Thirdly, the MDR prediction model combines CNN, GRU, and attention mechanism to optimize critical information utilization, significantly enhancing predictive performance.

3.2 Analysis of ice-covering warning for power grid lines in high-altitude regions

After verifying the effectiveness of the proposed CEEMDAN-SC-LSTM ice-covering feature extraction method and MDR prediction model, the study applies the proposed high-altitude area power grid ice-covering warning platform to a certain high-altitude mountainous area power grid line. The power grid lines in this high-altitude mountainous area often face serious ice-covering problems in winter, which poses a great threat to the power grid. The terrain in this area is complex, the climate is variable, the micro meteorological characteristics are obvious, the transmission lines are widely distributed, and some areas have inconvenient transportation, making manual inspection difficult. To timely and accurately grasp the ice-covering and deliver early warnings, this platform is used to monitor and predict line ice-covering, as displayed in Table 12.

Table 12: Monitoring results of ice-covering on power grid lines

Monitoring time	Actual ice thickness (mm)	Predicted ice thickness (mm)	Absolute error (mm)	Relative error (%)	Level of warning
2025-01-05 10:00	8	7.5	0.5	6.25	Light ice-covering
2025-01-10 14:00	12	11.2	0.8	6.67	Low warning ice-covering
2025-01-15 09:00	18	17.3	0.7	3.89	Low warning ice-covering

2025-01-20 11:00	25	24.1	0.9	3.6	Severe ice-covering
2025-01-25 13:00	5	5.3	0.3	6.0	Uncovered ice (close to threshold)

From Table 12, the predicted ice thickness by this method was close to the actual ice thickness, with a maximum absolute error of 0.9mm and a maximum relative error of 6.67%, indicating a relatively small overall error. The judged warning level was also consistent with the actual situation, and the warning accuracy was high. This indicates that the ice-covering prediction method based on LSTM and deep learning methods can effectively predict and deliver the ice-covering situation of transmission lines in high-altitude areas, providing timely decision-making basis, and helping to take measures to ensure the safe and stable operation.

4 Discussion

The CEEMDAN-SC-LSTM feature extraction model and MDR prediction model developed in this study demonstrated exceptional performance (R^2=0.9887) on the high-altitude mountainous dataset (October 2024-March 2025), significantly outperforming baseline models such as SVR (R^2=0.5281) and ANN (R^2=0.9429). This superiority stems from three key advantages: 1) multi-scale feature decomposition: The CEEMDAN algorithm adaptively extracts multi-modal features from ice-covered data, resolving the modal aliasing issue of traditional EMD (e.g., Reference [9] using only meteorological data), while SC clustering enhances feature separability. 2) Temporal modeling capability: The GRU-attention mechanism synergistically captures long-term meteorological patterns (e.g., Sudden temperature drop leads to ice formation), surpassing static image analysis methods in References [7] and [11]. 3) Regional adaptability: The model maintains high precision in complex micro-terrain areas (average altitude>2,000m), although its generalization in homogeneous climate zones (e.g., plains) requires further validation.

Notably, the extremely low MAE (0.0130mm) may be limited by the seasonal bias of the dataset (dominated by winter data). Therefore, it is necessary to integrate annual data to evaluate the robustness of the model during the spring thawing period.

5 Conclusion

Ice-covering on transmission lines in high-altitude regions poses a serious threat to power grid. This study aims to accurately predict ice-covering conditions. Therefore, a prediction method for ice-covering was proposed based on LSTM and deep learning methods. In the feature extraction stage, a CEEMDAN-SC-LSTM-based ice-covering feature extraction model was proposed by combining SC, CEEMDAN, and LSTM. A MDR prediction model was constructed by combining CNN, GRU, and attention mechanism models, taking the extracted features related to changes in ice-covering thickness as inputs. Finally, a power grid line ice-covering warning platform suitable for high-altitude regions was designed based on data and models to deliver real-time monitoring and warning. The classification accuracy was about 97.1%, the R^2 of the MDR model was 0.9887, and the MAE was 0.0130, demonstrating excellent performance in various performance indicators. The warning platform was applied to power grid lines in high-altitude mountainous areas, with a maximum absolute error of 0.9mm and a maximum relative error of 6.67% between predicted and actual ice-covering thickness, which could effectively provide warnings. However, there are issues with limited data samples and high model complexity in this study. Future research can collect more data, improve the model structure, make better predictions, and provide better support for the safe and stable running.

References

[1] Snaiki R, Jamali A, Rahem A. A metaheuristic-optimization-based neural network for icing prediction on transmission lines. Cold Regions Science and Technology, 2024, 224(1): 104249-104261. https://doi.org/10.1016/j.coldregions.2024.104249

[2] Wu C, You G, Chen Y. Power system resilience assessment considering the impact of ice disaster. 2024 3rd International Conference on Power Systems and Electrical Technology (PSET). IEEE, 2024,1(0): 482-487. https://doi.org/10.1109/PSET62496.2024.10808788

[3] Liu Y, Li Q, Liu J. Research on an icing-thickness calculation model for power transmission lines based on KPCA and GWO. Electric Power Systems Research, 2025, 248(1): 111877-111889. https://doi.org/10.1016/j.epsr.2025.111877

[4] Li B, Bai J, Wang W. Anti-icing and temperature-rising behavior of Superhydrophobic nanoporous Aluminum conductor by anodic oxidation. Journal of Physics: Conference Series. IOP Publishing, 2024, 2921(1): 012013-012024. https://doi.org/10.23919/PCMP.2023.000322

[5] Meixia L Y U, Xinyang Y A N, Jinmei H E, et al. Wire icing thickness prediction model of transmission line in the northeastern slope region of Qinghai-Xizang Plateau based on random forest algorithm. Journal of Arid Meteorology, 2025, 43(3): 488. DOI: 10.11755/j.issn.1006-7639-2025-03-0488

[6] Rajendrants S, Yogarathnarrus A, Zhao T. Improving grid resilience and reliability through dynamic line rating during ice precipitating conditions. 2025 IEEE PES Grid Edge Technologies Conference & Exposition (Grid Edge). IEEE, 2025,1(1): 1-5. https://doi.org/10.1109/GridEdge61154.2025.10887

467

[7] Hao Y, Liang W, Yang L. Methods of image recognition of overhead power line insulators and ice types based on deep weakly - supervised and transfer learning. IET generation, transmission & distribution, 2022, 16(11): 2140-2153. https://doi.org/10.1049/gtd2.12428

[8] Huang W, Hu B, Shahidehpour M. Preventive scheduling for reducing the impact of glaze icing on transmission lines. IEEE Transactions on Power Systems, 2021, 37(2): 1297-1310. https://doi.org/10.1109/TPWRS.2021.3099978

[9] Zhang Y, Zhang F, Qiu Y. An early warning method of transmission line icing based on power grid meteorological monitoring data. Journal of Physics: Conference Series. IOP Publishing, 2023, 2532(1): 012008-012014. https://doi.org/10.3934/mbe.2023334

[10] Yan X, Yuebin Z, Guobang B. Research on MMC de-icing system capable of power grid interconnection. 2021 Annual Meeting of CSEE Study Committee of HVDC and Power Electronics (HVDC 2021). IET, 2021, 2021(1): 29-38. https://doi.org/10.1049/icp.2021.2534

[11] Liang W, Hao Y, Huang L. Automatic calculation of graphical shed parameters for icing overhead power line insulators using contour peak-valley analysis. IEEE Transactions on Power Delivery, 2023, 39(2): 811-822. https://doi.org/10.1109/TPWRD.2023.3336002

[12] Ma X, Liu F, Liu Y. Design of low power consumption icing monitoring system for transmission lines. 2024 6th International Conference on Energy, Power and Grid (ICEPG). IEEE, 2024,1(10): 1066-1070. https://doi.org/10.1109/ICEPG63230.2024.10775502

[13] Hou H, Wan Y, Wang Z. Explainable Physical Deep-Learning Model for Overhead Transmission Line Icing-Thickness Prediction. IEEE Transactions on Industry Applications, 2025, 1(1):1-11. https://doi.org/10.1109/TIA.2025.3587191

[14] Qiu Z, Zhu X, Liao Cl. Detection of transmission line insulator defects based on an improved lightweight YOLOv4 model. Applied Sciences, 2022, 12(3): 1207-1220. https://doi.org/10.3390/app12031207

[15] Wang L, Chen Z, Zhang W. The causes and forecasting of icing events on power transmission lines in southern China: A review and perspective. Atmosphere, 2023, 14(12): 1815-1826. https://doi.org/10.3390/atmos14121815

[16] Dong, B., Jiang, X., & Yin, F. (2022). Development and prospect of monitoring and prevention methods of icing disaster in China power grid. IET Generation, Transmission & Distribution, 16(22), 4480-4493. https://doi.org/10.1049/gtd2.12614

[17] He X, Yang J, Zeng H. Monitoring Method and Application of Icing Thickness and Icing Type of Distribution Network Overhead Line Based on A Hydrophobic Marker and Image Defogging. 2024 IEEE 2nd International Conference on Power Science and Technology (ICPST). IEEE, 2024,1(1): 731-736. https://doi.org/10.1109/ICPST61417.2024.10602157

[18] Hou H, Wang Y, Bai X. Modelling icing growth on overhead transmission lines: Current advances and future directions. Energy Conversion and Economics, 2024, 5(6): 343-357. https://doi.org/10.1049/enc2.12131

[19] Zhou Y, Yang F, Xu J. A Non-Stop Ice-Melting Method for Icing Lines in Distribution Network Based on a Flexible Grounding Device. Energies, 2025, 18(8): 1886-1898. https://doi.org/10.3390/en18081886

[20] Zhang G. Transmission line trip faults under extreme snow and ice conditions: a case study. Energy Informatics, 2025, 8(1): 7-18. https://doi.org/10.1186/s42162-024-00463-8

Application of Machine Learning Algorithms for Anomaly Detection in Cybersecurity Threat Mitigation

Kim Son Lim[1], Shih Yin Ooi[1,2*], Yee Jian Chew[1,2], and Md Shohel Sayeed[1,3]
[1]Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, Melaka 75450, Malaysia
[2]Centre for Advanced Analytics (CAA), COE for Artificial Intelligence, Multimedia University, Jalan Ayer Keroh Lama, Melaka, 75450, Malaysia
[3]Centre for Intelligent Cloud Computing (CICC), COE for Advanced Cloud, Multimedia University, Jalan Ayer Keroh Lama, Melaka, 75450, Malaysia
E-mail: 1181402836@student.mmu.edu.my, syooi@mmu.edu.my, chewyeejian@mmu.edu.my, shohel.sayeed@mmu.edu.my
[*]Corresponding author

Overview paper

Keywords: artificial intelligence, cybersecurity, threat detection, machine learning, anomaly detection, adversarial attacks, ransomware, incident response, data security, predictive modelling

Received: July 5, 2025

The integration of Artificial Intelligence (AI) into cybersecurity has transformed the landscape of threat detection, analysis, and mitigation. As cyber-attacks become increasingly sophisticated and evasive, traditional rule-based defences are no longer sufficient to identify zero-day exploits and advanced persistent threats. AI-driven approaches, leveraging machine learning and deep learning, enable proactive anomaly detection, behavioural modelling, and predictive analytics that enhance both the accuracy and agility of cyber defence mechanisms.

This paper provides a comprehensive examination of AI applications in cybersecurity, spanning anomaly detection, automated incident response, and adaptive defence frameworks. It also emphasizes the emerging role of AI in vulnerability management, where predictive modelling, natural language processing, and automated remediation are used to identify, prioritize, and mitigate vulnerabilities before they can be exploited. A real-world case study of Panasonic's VERZEUSE™ platform is presented to illustrate the industrial implementation of AI-enhanced cybersecurity. The platform exemplifies how AI-based predictive analytics, threat intelligence integration, and continuous monitoring can strengthen risk management and compliance in complex IT and IoT ecosystems.

The findings demonstrate that AI substantially improves detection accuracy, response speed, and proactive defence capabilities. However, challenges related to data quality, model robustness, interpretability, and ethical deployment must be addressed to ensure trustworthy adoption. The study concludes that the future of cybersecurity depends on harmonizing human expertise with adaptive AI systems to achieve resilient, self-learning defence frameworks.

Povzetek: Prispevek obravnava uporabo umetne inteligence v kibernetski varnosti, kjer metode strojnega in globokega učenja bistveno izboljšujejo zaznavanje groženj, odzivanje na incidente in proaktivno zaščito, hkrati pa opozarja na izzive zaupanja, razložljivosti in etične rabe.

1 Introduction

1.1 Overview of the role of artificial intelligence in cybersecurity

Cybersecurity has become a strategic and operational priority for organizations globally due to the proliferation of interconnected systems, cloud services, and IoT devices. As the digital landscape expands, so too does the attack surface available to malicious actors. Modern threats range from conventional malware to sophisticated, adaptive techniques such as advanced persistent threats (APTs), zero-day exploits, and ransomware campaigns [1], [2]. These attacks can result in severe consequences, including data breaches, financial loss, operational disruption, and reputational damage.

Traditional cybersecurity approaches, primarily rule-based systems and signature-based detection, lack adaptive and predictive capabilities when confronting rapidly evolving or previously unknown attack vectors. Static defense mechanisms are inadequate in the face of adversaries who leverage automation, obfuscation, and AI-enhanced attack techniques [3], [4].

To meet these challenges, cybersecurity strategies increasingly incorporate Artificial Intelligence (AI) and Machine Learning (ML) to bolster defence capabilities. AI enables real-time threat detection, behavioural analysis, and predictive modelling, providing security systems with the ability to identify anomalies and respond autonomously [1], [5]. Unlike conventional tools, AI-driven systems can analyse vast amounts of historical and real-time data, detect emerging patterns indicative of malicious activity, and continuously improve their performance over time.

This section presents an overview of the role of AI in cybersecurity, focusing on its application in threat intelligence, intrusion detection, behavioural analytics, and adaptive response mechanisms.

1.2 Importance of detecting and mitigating cyber threats

The failure to detect and mitigate cyber threats in a timely manner can lead to severe consequences for both public and private sector organizations. Cyber incidents such as data breaches may result in the unauthorized disclosure of sensitive information, privacy violations, operational disruptions, financial losses, and reputational damage [6]. In critical infrastructure environments, undetected attacks can compromise essential services, disrupt business continuity, and pose risks to public safety [7].

Recent studies indicate that the global average cost of a data breach has reached millions of dollars per incident, further emphasizing the financial implications of inadequate cyber defences [8]. Consequently, rapid and proactive threat detection has become imperative. Organizations must adopt mechanisms capable of identifying and mitigating cyber threats in their early stages to minimize impact and ensure resilience.

Failure to implement timely countermeasures can erode stakeholder trust, undermine competitive advantage, and in severe cases, jeopardize the survival of the affected entity [9]. Advanced cybersecurity strategies, including AI-enhanced detection systems, are critical in maintaining situational awareness and achieving operational continuity in an increasingly hostile digital environment.

1.3 Objectives

This study aims to examine the diverse applications of AI in the field of cybersecurity, with the goal of enhancing threat detection, response, and mitigation capabilities. By exploring a range of AI-driven methodologies, such as machine learning, behavioural analytics, and anomaly detection. This research seeks to illustrate how AI technologies contribute to strengthening organizational cyber defence mechanisms [9], [10].

1.4 The key area of focus in ai-drive cybersecurity

This study highlights several critical domains in which AI contributes to enhancing cybersecurity capabilities:

1. **Anomaly and behaviour detection**: AI, particularly through ML algorithms, is leveraged to monitor and identify abnormal patterns in network traffic, user behaviour, and system activities. These approaches enable early detection of potentially malicious actions that traditional rule-based systems may overlook [11], [12].
2. **Automated defence and response**: AI supports the automation of incident response, intrusion prevention, and vulnerability management. By enabling faster decision-making and dynamic adaptation to threats, AI improves the speed and effectiveness of cybersecurity operations [13].
3. **Challenges and risks**: Despite its benefits, the deployment of AI in cybersecurity faces several challenges, including the need for high-quality labelled data, the risk of adversarial attacks targeting AI models, and ongoing concerns surrounding ethics, privacy, and regulatory compliance [14].
4. **Emerging techniques**: Advanced AI paradigms such as reinforcement learning, federated learning, and integration with blockchain technologies are gaining traction as future-forward strategies to increase cyber resilience and secure distributed environments [15], [16].

By exploring these dimensions, this research provides insights into how AI technologies can be strategically applied to help organizations address evolving cyber threats and more effectively safeguard digital assets.

2 Cybersecurity threat landscape

The cybersecurity threat landscape is in a state of continuous evolution, presenting organizations across various industries with an expanding spectrum of sophisticated cyber-attacks that pose significant operational and financial risks [2,17]. Prominent categories of cyber threats include:

a. **Phishing:** A social engineering attack vector aimed at deceiving individuals into divulging sensitive data or inadvertently executing malicious software, often without the user's knowledge [18].
b. **Malware:** A broad classification of malicious software—encompassing viruses, worms, Trojans, and spyware—designed to infiltrate, disrupt, or damage computer systems without authorization [19].
c. **Ransomware:** A subclass of malware that encrypts critical data assets and extorts payment from victims in exchange for decryption keys, causing substantial operational disruption and financial loss [20].
d. **Distributed Denial-of-Service (DDoS) Attacks:** Coordinated attempts to incapacitate network services by overwhelming target systems with a flood of illegitimate traffic originating from multiple distributed sources [21].
e. **Insider threats:** Malicious or negligent activities perpetrated by authorized users or employees

that result in unauthorized data access, exfiltration, or system compromise [22].

f. **Advanced Persistent Threats (APTs):** Highly sophisticated, often state-sponsored cyber campaigns characterized by prolonged and targeted intrusion efforts designed to covertly extract sensitive information from high-value organizations [23].

The sophistication and frequency of these cyber threats have escalated markedly, severely impacting critical sectors such as finance, healthcare, government, and infrastructure. The increasing digital dependency of organizations amplifies the potential consequences of these threats, jeopardizing financial stability, operational continuity, and corporate reputation [24].

2.1 Emergence of advanced threats and associated challenges

The rising complexity of cyber-attacks is fundamentally reshaping cybersecurity paradigms. Polymorphic malware, capable of dynamically altering its code to evade signature-based detection, exemplifies a formidable challenge to traditional defence mechanisms [25], [26]. Compounding this issue, zero-day vulnerabilities—previously undisclosed security flaws lacking available patches—enable attackers to compromise systems prior to the deployment of defensive countermeasures [27].

Moreover, adversaries are increasingly exploiting automated tools and AI to enhance the precision, scale, and efficacy of cyber-attacks. AI-driven methodologies facilitate the orchestration of sophisticated phishing campaigns, development of adaptive stealth malware, and coordination of complex multi-vector assaults [28], [29]. This technological escalation underscores an urgent imperative for cybersecurity frameworks to adopt adaptive, intelligence-driven defences capable of countering evolving threats.

2.2 Limitations of traditional cybersecurity solutions

Conventional cybersecurity approaches predominantly relying on static rules and signature-based detection face critical limitations in addressing the dynamic and rapidly evolving threat landscape [30]. These reactive methods frequently fail to detect novel or polymorphic attacks that continuously evolve to bypass fixed detection criteria [31]. Consequently, traditional solutions are insufficient for proactive threat identification and mitigation.

AI has emerged as a pivotal enabler, imparting cybersecurity systems with adaptive, predictive, and comprehensive capabilities [32]. AI-driven platforms enhance anomaly detection, early threat identification, and automated response mechanisms, thereby substantially improving organizational security postures [33]. Given the fluidity of cyber threats, the integration of AI is essential to maintain resilient defences and safeguard sensitive information assets [34].

3 Artificial intelligence in threat detection

The increasing sophistication of cyber threats necessitates the deployment of AI to augment threat detection efficacy. AI-powered systems can analyse vast volumes of heterogeneous security data in real-time, discerning novel attack patterns and anomalies beyond the scope of traditional methods [35], [36]. This capability markedly improves detection accuracy and latency, empowering organizations to adopt a proactive cybersecurity stance [37].

3.1 Machine learning for anomaly detection

A cornerstone of AI-enabled threat detection is the application of machine learning algorithms to identify anomalous behaviours indicative of security incidents. Machine learning techniques in cybersecurity are conventionally classified into three categories [38], [39]:

- **Supervised learning:** Models are trained on labelled datasets containing both benign activities and known cyber-attacks, enabling classification of future observations based on learned discriminative features.
- **Unsupervised learning:** Algorithms analyse unlabelled data to detect deviations from normative behavioural baselines, facilitating identification of previously unseen or unknown threats.
- **Semi-supervised learning:** A hybrid approach that leverages both labelled and unlabelled data to enhance detection performance, particularly valuable when labelled data is limited [40].

By leveraging these methodologies, AI-driven cybersecurity systems can detect anomalous activities spanning networks, user behaviours, and host systems, thereby identifying threats potentially missed by signature-based defences [41].

3.2 Behavioural analysis and pattern recognition

Building on anomaly detection, AI systems monitor user and system behaviours—such as login patterns, data access, and network utilization—to establish behavioural baselines. Using clustering and classification techniques, deviations from these baselines are identified as potential indicators of compromise, enabling timely threat detection and response.

3.3 Natural Language Processing (NLP) for phishing detection

AI techniques powered by Natural Language Processing (NLP) have demonstrated efficacy in detecting phishing attempts by analysing linguistic cues within emails and electronic communications. NLP models can identify deceptive language patterns and suspicious contextual signals, facilitating rapid mitigation of phishing

campaigns which often serve as entry points for broader cyber-attacks.

4 Real-time threat detection with deep learning

Deep learning—an advanced subset of machine learning utilizing neural networks—has enhanced AI-based threat detection by enabling the identification of polymorphic malware and complex attack vectors through multi-layered feature extraction from network and system telemetry. Organizations employing deep learning approaches benefit from reduced detection latency and improved threat identification accuracy, thus mitigating potential damages [42].

4.1 AI in threat mitigation

Effective cybersecurity extends beyond detection to timely mitigation. AI enhances incident response by automating threat containment and remediation actions.

4.2 Automated incident response and threat remediation

AI outputs that maps to actionable runbooks and remediation steps not just alerts are integrated with existing playbook as part of the SIEM SOAR functions. AI-enabled security orchestration, automation, and response (SOAR) platforms rapidly detect incidents and autonomously execute predefined remediation workflows, such as quarantining infected devices, blocking malicious IP addresses, and isolating suspicious files, thereby reducing dwell time and limiting attack impact.

4.3 AI-enhanced intrusion prevention systems (IPS)

Integrating AI with Intrusion Prevention Systems (IPS) enables dynamic threat recognition and blocking capabilities beyond static signature rules. Machine learning models continuously analyse network and system data to identify and respond to emerging threats, adapting security postures in real-time.

4.4 Role of AI in vulnerability management

AI assists in proactive vulnerability management by scanning organizational IT assets to identify exploitable weaknesses. Through risk scoring based on exploitability and impact, AI prioritizes remediation efforts, enabling cybersecurity teams to focus resources on the most critical vulnerabilities.

- Automated discovery and classification where AI-driven scanner leverage ML and NLP to analyse system configurations, code repositories, and threat feeds to automatically detect new vulnerabilities even before CVEs are published. Pattern recognition for detecting zero-day or configuration-based vulnerabilities that traditional signature-based tools might miss. AI-algorithms cluster similar vulnerabilities, reducing duplicate finding and streamlining analyst review.
- Risk prioritization and predictive scoring where all model combine assets criticality, exploit availability, threat intelligence, and business context to assign dynamic risk score. Predictive models assess which vulnerabilities are most likely to be exploited in the near future, enabling teams to patch what matters the most. The approach goes beyond CVSS scoring by factoring in real-world exploitation trends.
- Intelligence patch and remediate planning where AI can recommend or automate patch deployment sequences by evaluating dependencies, system uptime windows and potential service impacts. ML analyses past remediation patterns to predict patch success probability and minimize operational disruptions. Integration with SOAR platform allows AI to tigger patch workflow to predict patch workflow automatically for low-risk systems.
- Continuous monitoring and adaptive defence continuously ingest logs and telemetry to verify if patched systems remain secure or if exploit attempts persist. Behavioural analytics can flag regression vulnerabilities or partial remediations that conventional checks may overlook. Predictive Analytics for Proactive Cyber defence. Adaptive learning ensures detection models evolve alongside the threat landscape
- Threat intelligence correlation correlates external threat feeds, dark web data, and CVE disclosures to identify vulnerabilities actively discussed or weaponized by attackers. This correlation supports proactive vulnerability mitigation, turning raw data into actionable intelligence.

AI-driven predictive analytics synthesize historical and current threat intelligence to forecast likely attack vectors, empowering organizations to pre-emptively strengthen defences and reduce exposure.

5 Challenges of AI in cybersecurity

Despite its transformative potential, AI adoption in cybersecurity faces notable challenges:

a. Data quality and availability
The effectiveness of AI models hinges on access to extensive, high-quality datasets representative of evolving threats. The dynamic nature of cyber threats complicates data collection, annotation, and timely updating, potentially degrading model performance. Data quality and label scarcity models needs good, representative labelled events, and noisy logs lad to false positives and analyst fatigue.

b. Adversarial attacks on AI models
AI systems themselves are vulnerable to adversarial manipulation, where attackers craft inputs designed to evade detection or cause

misclassification, undermining trust in AI-based defences.

c. Ethical and privacy concerns
Deployment of AI-driven monitoring and decision-making systems raises significant privacy and ethical considerations, particularly regarding data collection, user consent, and potential biases.

d. Interpretability and trustworthiness
The complexity of AI, particularly deep learning models, often results in opaque decision processes that hinder interpretability and user trust. Explainable AI (XAI) methods are crucial to enhance transparency, accountability, and stakeholder confidence. Integration with existing playbooks, handling telemetry, models, and retraining governed by privacy regulatory regimes and product safety.

6 Emerging trends in Ai-driven cybersecurity

As AI continues to evolve, several innovative trends are shaping the future of cybersecurity, offering new methods to defend against increasingly sophisticated threats.

a. Reinforcement learning for adaptive defence
Reinforcement learning (RL) has emerged as a promising approach for developing autonomous cybersecurity systems capable of adapting to dynamic and evolving threats. By interacting with simulated or real network environments, RL agents learn optimal defence policies through trial-and-error, continuously refining their strategies based on feedback or rewards. In cybersecurity contexts, RL can be applied to intrusion detection, malware containment, and network traffic control, where decisions, such as isolating suspicious nodes or adjusting firewall rules must be made dynamically under uncertainty. For example, RL-based intrusion detection systems can automatically adjust detection thresholds or response actions in real time, minimizing false positives while improving reaction speed to new attack patterns. This adaptability enables proactive and context-aware defence, aligning cybersecurity operations with evolving attack behaviours.

b. Federated learning for distributed cybersecurity
Federated learning (FL) addresses a critical challenge in cybersecurity, i.e., how to collaboratively improve threat detection models without compromising sensitive or proprietary data. In a federated setup, multiple organizations or devices train local AI models on their own data and share only model updates, not the raw data itself. This decentralized learning paradigm enhances privacy and compliance with data protection regulations while enabling cross-organizational intelligence sharing. In practice, FL supports distributed intrusion detection systems and IoT security frameworks where data from endpoints or edge devices can collectively improve anomaly detection models. By aggregating global intelligence without centralizing data, federated learning strengthens the overall robustness of cyber defences against emerging and distributed attack vectors.

c. Integration of AI with blockchain
Combining AI with blockchain technology enhances the security, transparency, and integrity of AI models and data, mitigating risks of tampering and ensuring trustworthiness.

d. AI-driven deception technologies
AI-powered honeypots and deception environments emulate realistic systems to lure and analyse attacker behaviours, providing actionable intelligence for improving defensive strategies.

e. Ethical and regulatory implications regulatory requirements
Compliance with data protection regulations such as the EU's GDPR and the US CCPA imposes strict constraints on AI-based cybersecurity solutions regarding personal data handling and user privacy.

f. Ethical considerations
Organizations must balance AI deployment with respect for privacy, consent, and civil liberties, ensuring transparent communication and adherence to ethical standards.

g. Development of guidelines and standards
Standards bodies like IEEE and NIST are actively developing ethical frameworks and best practices to guide responsible AI use in cybersecurity, emphasizing transparency, fairness, and accountability.

6.1 Case study: panasonic's Ai-driven cybersecurity ecosystem

Example of AI with Security adopted by the Panasonic. AI transforms vulnerability management from a reactive process to a proactive, predictive, and continuous function. Panasonic's use of AI platforms like VERZEUSE™ exemplifies how enterprises can apply advanced analytics to strengthen cyber resilience, reduce exposure, and maintain compliance in complex IT/OT ecosystems.

- It uses AI to simulate attack paths and predict the most probable exploitation vectors.
- It automates reporting and risk assessment, enabling faster patch development and deployment cycles.
- It integrates with Vehicle SOC operations, ensuring continuous monitoring and adoptive response.

VERZEUSE – Vehicle Risk and Vulnerability Analysis System a proprietary AI-driven platform that automatically identifies, evaluates, and prioritises

potential vulnerabilities in connected vehicles. The key components of VERZEUSE

- TARA (Threat Analysis & Risk Assessment) design where the tool leverages an internal threat intelligence database (Panasonic's collected example of threats, mitigation, previous incidents) to map threats, vulnerability and control appropriate of the ECU type. Generates an SO/SAE-21434-compliant output: threat scenarios, risk assessments, required countermeasures virtualization extension for implementation in modern cockpit system with virtualization or hypervisor containers.
- Runtime integrity checker to ensure monitoring functions themselves aren't tampered with during operation. Includes a "heartbeat" mechanism with digital signatures so that logs and status reports are validated and external verification (Vehicle SOC or equivalent) can detect if tampering has occurred. If the integrity module detects tampering or abnormal shutdown of security components, it triggers alert/notification outside the vehicle. This has been "adopted as an in-vehicle product" in some cases and highly evaluated by OEMs.
- Threat evaluation and security Test Assistance toolkit for testing, SIRT (Security Incident Response Team) to handle vulnerabilities post shipment, using attack path analysis. After vehicles are shipped, known vulnerabilities (from internal or external sources) are fed into the system. It uses data like: SBOM (Software Bill of Materials) for each ECU, connection topology (how ECUs are networked or communicate) to understand how vulnerabilities might be exploited or escalate across ECUs. Panasonic Newsroom Global+1. It also uses Panasonic's Threat Intelligence data to understand current exploit patterns. The algorithm estimates possible attack routes & impact across the whole vehicle (not just isolated ECUs), calculates risk scores, and helps prioritize which vulnerabilities to remediate first. Ensures alignment with ISO/SAE 21434 for vulnerability management and risk assessment. Shown to reduce time for analysis significantly.

Through the platform, Panasonic has transitioned from periodic vulnerability scanning to predictive, model-based vulnerability management, enhancing both speed of remediation and accuracy or risk prioritization, particularly viral for IoT and automotive systems where downtime directly impacts product safety.

From another example, Panasonic has taken several concrete steps involving SOC, SIEM, and SOAR for automation orchestration especially in its VERZEUSE™ programs and vehicle factory operations. The Key experiment in SOC services includes:

a. **Vehicle SOC + McAfee partnership**
Panasonic and McAfee built a Vehicle Security Operation Center to handle monitoring of connected vehicles. The data architecture involves Automotive Intrusion Detection Systems onboard vehicles sending telemetry / log / alert data to a backend SOC + SIEM system for analysis and visualization. This indicates Panasonic is operationalizing SIEM for post-shipment / runtime vehicle security monitoring.

b. **VERZEUSE™ for SIEM (Operation / Post-shipment)**
As part of VERZEUSE™, they have a module for SIEM. It "automatically detects and analyses cyber-attacks on shipped vehicles" to "accelerate and streamline security monitoring. It supports threat intelligence from Panasonic Group's broader CI sources.

c. **VERZEUSE™ for SIRT (Vulnerability Management after shipment)**
For vulnerabilities discovered post-shipment, Panasonic uses VERZEUSE for SIRT to analyse risk using SBOM (software bill-of-materials), ECU connectivity, and their own threat intelligence, to prioritize remediations.

d. **Factory / Manufacturing SOC Operations**
Panasonic has had SOCs operating for factories (since ~2016) to protect factory networks and systems. They analyse communications in factory environments for anomalies & malware intrusions.

e. **Proof of Concepts (PoCs) on log analytics & anomaly detection**
Panasonic's Annual Reports mention PoCs to "analyse huge data logs and to detect abnormalities. These to be preliminary experiments to build capability in monitoring, detection and response.

7 Conclusion

AI has become indispensable in fortifying cybersecurity defences against increasingly complex cyber threats. Leveraging machine learning, deep learning, and related AI techniques, organizations achieve improved detection accuracy, faster response times, and adaptive protection.

Looking forward, AI's role in autonomous cybersecurity systems and cross-domain intelligence integration is poised to expand, contingent on addressing challenges such as adversarial robustness, model explainability, and ethical deployment. A multidisciplinary approach involving technologists, policymakers, and industry stakeholders is imperative to harness AI's potential while safeguarding privacy and civil rights.

Disclosure statement

We declare there are no conflicts of interest regarding the research, authorship, or publication of this article.

References

[1] S. A. Shaikh, N. A. Shaikh, and M. N. Shaikh, "A review of artificial intelligence techniques in cybersecurity," *2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, Sukkur, Pakistan, 2020, pp. 1–5, doi: 10.1109/iCoMET48670.2020.9073885.

[2] M. Conti, A. Dehghantanha, K. Franke, and S. Watson, "Internet of Things security and forensics: Challenges and opportunities," *Future Generation Computer Systems*, vol. 78, pp. 544–546, Jan. 2018, doi: 10.1016/j.future.2017.07.060.

[3] M. Vinayakumar, K. P. Soman, and P. Poornachandran, "Evaluating deep learning approaches to intrusion detection," *Procedia Computer Science*, vol. 132, pp. 485–490, 2018, doi: 10.1016/j.procs.2018.05.198.

[4] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," *2010 IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 2010, pp. 305–316, doi: 10.1109/SP.2010.25.

[5] IBM Security, "Cost of a Data Breach Report 2023," IBM Corporation, 2023. [Online]. Available: https://www.ibm.com/reports/data-breach

[6] M. Conti, A. Dehghantanha, K. Franke, and S. Watson, "Internet of Things security and forensics: Challenges and opportunities," *Future Generation Computer Systems*, vol. 78, pp. 544–546, Jan. 2018, doi: 10.1016/j.future.2017.07.060.

[7] H. A. Mohammed and I. M. Husien, "A Deep Transfer Learning Framework for Robust IoT Attack Detection," *IJCAI*, vol. 48, no. 12, Sep. 2024, doi: 10.31449/inf.v48i12.5955.

[8] M. Alazab, S. Venkatraman, P. Watters, and M. Alazab, "Zero-day malware detection based on supervised learning algorithms of API call signatures," in *Proceedings of the Ninth Australasian Data Mining Conference - Volume 121*, in AusDM '11. AUS: Australian Computer Society, Inc., 2011, pp. 171–182.

[9] L. Idouglid, S. Tkatek, and K. Elfayq, "Multilayer Perceptron-Based Defense Mechanisms for Securing Industrial IoT in Industry 4.0 Environments," *IJCAI*, vol. 49, no. 33, Aug. 2025, doi: 10.31449/inf.v49i33.6944.

[10] R. M. Alguliyev, Y. E. Imamverdiyev, and L. A. Sukhostat, "Cyber-attacks detection in critical information infrastructures using artificial intelligence approaches: A review," *IEEE Access*, vol. 9, pp. 14014–14043, 2021, doi: 10.1109/ACCESS.2021.3052133.

[11] M. Vinayakumar, K. P. Soman, and P. Poornachandran, "Evaluating deep learning approaches to intrusion detection," *Procedia Computer Science*, vol. 132, pp. 485–490, 2018, doi: 10.1016/j.procs.2018.05.198.

[12] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," *2010 IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 2010, pp. 305–316, doi: 10.1109/SP.2010.25.

[13] N. Idika and B. Bhargava, "Extending attack graph-based security metrics and aggregating their application," *IEEE Trans. Dependable and Secure Computing*, vol. 9, no. 1, pp. 75–85, Jan.-Feb. 2012, doi: 10.1109/TDSC.2011.44.

[14] R. M. Alguliyev, Y. E. Imamverdiyev, and L. A. Sukhostat, "Cyber-attacks detection in critical information infrastructures using artificial intelligence approaches: A review," *IEEE Access*, vol. 9, pp. 14014–14043, 2021, doi: 10.1109/ACCESS.2021.3052133.

[15] H. Huang, Z. Wang, and X. Huang, "A blockchain-based privacy-preserving incentive mechanism in crowdsensing applications," *IEEE Access*, vol. 7, pp. 62996–63006, 2019, doi: 10.1109/ACCESS.2019.2916556.

[16] Y. Liu, R. Wang, and Y. Chen, "Federated learning for cyber security: Concepts, applications, and future directions," *IEEE Trans. Ind. Informatics*, vol. 18, no. 5, pp. 2926–2939, May 2022, doi: 10.1109/TII.2021.3085284.

[17] M. Conti, A. Dehghantanha, K. Franke, and S. Watson, "Internet of Things security and forensics: Challenges and opportunities," *Future Generation Computer Systems*, vol. 78, pp. 544–546, Jan. 2018.

[18] R. Kumar and P. Kumar, "Phishing attack detection using machine learning techniques," in *Proc. Int. Conf. Computational Intelligence and Data Science*, 2019, pp. 157–160.

[19] S. Axelsson, "The base-rate fallacy and its implications for the difficulty of intrusion detection," in *Proc. ACM Conf. Computer and Communications Security*, 1999, pp. 1–7.

[20] K. M. Rogers, "Ransomware: Evolution, mitigation and prevention," *Journal of Information Security*, vol. 7, no. 1, pp. 11–22, Mar. 2016.

[21] K. Thomas et al., "Design and evaluation of a real-time network traffic anomaly detection system," *IEEE Trans. Dependable Secure Comput.*, vol. 14, no. 5, pp. 492–505, Sep.–Oct. 2017.

[22] D. L. White, "Insider threat detection: Techniques and countermeasures," *Journal of Cybersecurity and Privacy*, vol. 2, no. 3, pp. 98–112, 2018.

[23] A. A. Azab, M. Alazab, and S. Alazab, "Advanced persistent threat: Detection, prevention and mitigation," *IEEE Access*, vol. 8, pp. 169583–169602, 2020.

[24] S. Z. Alawneh, A. J. Alenezi, and I. A. Alenezi, "Cybersecurity risks in critical infrastructure: A survey," *Computers & Security*, vol. 93, article 101751, Apr. 2020.

[25] H. J. Wang, M. I. Wang, and H. Yu, "Polymorphic malware detection using machine learning," in *Proc. IEEE Int. Conf. Big Data*, 2018, pp. 3087–3095.

[26] T. Holz, M. Steiner, F. Dahl, E. Biersack, and F. C. Freiling, "Measurements and mitigation of peer-to-peer-based botnets: A case study on Storm worm," in

Proc. 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats, 2008.
[27] Z. Zhao, J. Fan, and Q. Chen, "Zero-day vulnerability detection based on machine learning," *IEEE Access*, vol. 7, pp. 18439–18450, 2019.
[28] Y. Yang et al., "Artificial intelligence-enabled phishing attacks: Threats and countermeasures," *IEEE Network*, vol. 34, no. 2, pp. 123–129, Mar.–Apr. 2020.
[29] A. S. Elmaghraby and M. M. Losavio, "Cybersecurity challenges in Smart Cities: Safety, security and privacy," *Journal of Advanced Research*, vol. 5, no. 4, pp. 491–497, 2014.
[30] M. Roesch, "Snort - lightweight intrusion detection for networks," in *Proc. 13th USENIX Conf. System Administration*, 1999, pp. 229–238.
[31] G. Creech and J. Hu, "A semantic approach to host-based intrusion detection systems using contiguous and discontiguous system call patterns," *IEEE Trans. Comput.*, vol. 63, no. 4, pp. 807–819, Apr. 2014.
[32] N. H. Motlagh, M. Bagaa, and T. Taleb, "UAV-based IoT platform: A crowd surveillance use case," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 29–35, Apr. 2018.
[33] K. A. Yokoyama, S. Yamaguchi, and Y. Hori, "Anomaly detection using deep learning for cyber-physical system security," in *Proc. IEEE Int. Conf. on Cyber-Physical Systems*, 2020, pp. 45–54.
[34] M. S. Islam, T. Kwak, M. H. Islam, and J. H. Park, "Cybersecurity and artificial intelligence: Research advances and applications," *IEEE Access*, vol. 7, pp. 12864–12877, 2019.
[35] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Commun. Surveys Tuts.*, vol. 10, no. 4, pp. 56–76, Fourth Quarter 2008.
[36] J. Zhang, X. Xiao, and M. Li, "A survey on machine learning techniques for phishing detection," *IEEE Access*, vol. 6, pp. 29661–29674, 2018.
[37] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[38] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
[39] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[40] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*. Cambridge, MA, USA: MIT Press, 2006.
[41] D. E. Denning, "An intrusion-detection model," *IEEE Trans. Software Eng.*, vol. SE-13, no. 2, pp. 222–232, Feb. 1987.
[42] A. S. Tanenbaum and D. J. Wetherall, *Computer Networks*, 5th ed. Upper Saddle River, NJ, USA: Pearson, 2011.

RMGAT: Bipartite Interaction Graph-Based Relation Fusion Multi-Head Graph Attention Recommendation Model

Junju Sun*, Longlong Zhang
School of Information and Communication Engineering, Xinyang Vocational and Technical College, Xinyang 464000, China
E-mail: gengxiaoyuan@byau.edu.cn, zhanglonglong@xyvtc.edu.cn
*Corresponding author

Keywords: graph representation learning, bipartite interaction graph, recommendation model, simulation experiment, multi-head graph attention, relation extraction

Received: April 17, 2025

Recommendation systems are crucial for accurately matching user needs in an environment with massive information. Traditional recommendation models have many deficiencies when dealing with user-item relationships and it is difficult to fully explore the information contained in user behaviors. This paper proposes a relation fusion multi-head graph attention recommendation model based on bipartite interaction graphs (RMGAT), aiming to innovate the performance of recommendation systems. The RMGAT model is based on the theory of user behavior, clearly dividing the interaction between users and items into explicit and implicit relationships, and constructing a bipartite interaction graph. By modeling these two types of relationships respectively, the explicit preferences with high credibility and the widely existing implicit latent interests can be effectively distinguished, overcoming the drawback of the traditional model in the semantic fuzzy processing of user behaviors. Meanwhile, the relational fusion multi-head graph attention mechanism is innovatively designed. With the help of the multi-head structure, the multiple semantic features of the explicit and implicit relationships are captured in parallel. The attention coefficient is used to dynamically adjust the weights to achieve the efficient complementary fusion of explicit and implicit information, significantly enhancing the model's expression ability for complex relationships, and thereby improving the accuracy and diversity of the recommendation results. After experimental verification on multiple typical datasets, the RMGAT model performs outstandingly in the recommendation scenarios of different fields and has achieved significant improvements in multiple key indicators compared with mainstream models. Future research will focus on developing lightweight multi-head mechanisms, introducing dynamic relationship modeling, and constructing cross-modal heterogeneous fusion frameworks, etc., in order to further optimize model performance and expand its application potential in more fields.

Povzetek: Članek predstavi priporočilni model RMGAT, ki gradi na bipartitnem grafu uporabnik–izdelek in ločeno modelira eksplicitne in implicitne odnose.

1 Introduction

The popularization and development of the Internet not only facilitates people to quickly obtain a large amount of information, but also results in information overload [1]. How to obtain valuable information from massive data quickly and efficiently has become a challenge in the scientific research field. Recommendation system is a new research direction in the field of machine learning. As a new technology to alleviate information overload [2-4], it can find information of interest to users according to user preferences from massive data information and recommend it to users, which has been successfully applied to various Internet platforms. On the other hand, the Internet platform has a huge user group and item information, as well as the historical interaction information between users and items, which provides strong data support for the research and development of the recommendation system. With the development of research on recommendation system, the performance of recommendation system is getting better and better. Users can quickly obtain items of interest according to their preferences, and the platform can also actively recommend items of interest to users through the recommendation system, so as to better enhance user experience and user viscosity. Relying on the use of high-performance servers for large-scale training, graph representation learning is a popular research field of machine learning in recent years. Advanced machine learning algorithms [5-7] are integrated to represent nodes in the graph as low-dimensional feature vectors using machine learning methods, and at the same time, node vectors represent structural information in the graph as much as possible. The recommendation system can make full use of a variety of interactive behavior

data [8], mining relationship information from graph structure data, and learning the feature vector representation of users, items and related entities in graph structure data. The combination of graph representation learning and recommendation system can effectively enhance the learning ability of recommendation system, and then improve the accuracy of recommendation and user satisfaction. The recommendation system based on graph representation learning has high research value and is also the main trend in the field of recommendation system. In view of the limitations of the traditional attention mechanism in relation modeling, this study proposes a relation fusion multi-head graph attention mechanism. The different semantic features of the explicit and implicit relationships are captured in parallel by using the multi-head structure, and the weights of the two types of relationships are dynamically adjusted through the attention coefficient to achieve the complementary fusion of explicit and implicit information. This mechanism can effectively explore the diversity of users' interests, enhance the model's ability to express complex relationships, and improve the accuracy and diversity of recommendation results.

2 Related works

The recommendation model based on machine learning can process the recommendation with complex operations, find out the hidden patterns in the data, extract the features from the data set composed of users and their voting conditions, obtain the algorithm model, and then use the model to predict the recommendation [9]. However, the recommendation algorithm based on machine learning can only model the low-order feature representation in the user's historical interaction data, and lacks the modeling ability for the higher-order feature representation. Most of the data in a recommendation system can be regarded as graph-structured data in nature.

The current SOTA models in the field of recommendation systems (such as SGL, DGCF, and UltraGCN) have significantly enhanced the accuracy and robustness of recommendations under the GNN framework by improving representation learning and interaction modeling. SGL constructs multi-view node representations through "data augmentation + contrastive learning" to supplement the deficiencies of traditional supervised signals. Specifically, it includes: using node discarding, edge discarding and random walk to generate different views of the same node to alleviate the dominance of highly advanced nodes and noise interference. Maximize the similarity of different views on the same node, minimize the similarity of views on different nodes, and enhance the discrimination ability of node representations. However, data augmentation strategies need to be customized for different scenarios and overly rely on hyperparameter tuning. Self-supervised tasks increase computing costs, and scalability on large-scale data needs to be further verified. DGCF models the latent motives of user-item interaction through intention decoupling and generates intent-aware interaction graphs. Specifically, it includes: decomposing the user/item embedding into multiple blocks, with each block corresponding to a potential intention. Through iterative optimization of the intent-aware graph, high-impact interaction relationships are highlighted. Introduce distance correlation loss, enforce the independence of different intentions, and enhance interpretability. However, the computational complexity is high, the number of intentions needs to be preset, and the flexibility is insufficient. The ability to model the intentions of cold-start users/items is relatively weak. UltraGCN improves efficiency and accuracy by simplifying the message passing mechanism and multi-task learning. Specifically, it includes: directly optimizing the convergence conditions of GCN to avoid excessive smoothing caused by multi-layer stacking. Combine the principal loss of the user-item graph and the constrained loss of the item-item graph to capture the collaborative signal. Alleviate data sparsity and improve training stability through weighted negative samples. However, the modeling ability for high-order relations is limited and lacks dynamic adaptability. Relying on predefined graph structures makes it difficult to handle dynamic interactive data.

For example, the user's interaction with items, from the perspective of graph structure, once the user clicks or purchases the item, it is associated with these items. But the vast majority of deep learning tasks are conducted on Euclidean structured data. Based on the given interaction graph structure and initial node, capture the complex interaction relationship between users and item nodes [10, 11]. Graph representation learning uses user interaction graph topology and node attribute information to map each node to a low-dimensional vector representation, so as to improve the recommendation effect and effectively alleviate the problems of data sparsity and cold start [12].

The recommendation system based on the knowledge graph is to use the entity, attribute, relationship and other information in the knowledge graph to recommend resources or services that users may be interested in [13]. In the recommendation system integrated into the knowledge graph, the graph neural network follows the idea of neighborhood aggregation, aggregates the feature information of the neighbor entity in the knowledge graph, and then updates the feature representation of the central entity based on it, so as to realize the mining of multiple types and multiple relationships in the knowledge graph. The association information of knowledge attributes and the interaction and cooperation information between users and items are integrated into the feature representation process of users and items to effectively represent the potential semantics of users and items in vector space [14]. The first order interaction information between users and items is used to build the initial entity set of users and items, and the first order interaction information is explicitly encoded into the initial entity set through the co-propagation layer, so as to enhance the representation of users and items. By using the knowledge graph structure, the

higher-order neighbors of the entities are extended, and the extended entity set is constructed to make the message propagate along the link. The dynamic embedding method of knowledge perception is used to improve the recommendation effect by generating different attention coefficients to focus on different head and tail entities.

For the four recommendation models related to this study, namely "GraphRec, KGAT, LightGCN, and PersonalRank", a brief comparison is shown in Table 1 [15-19].

The initialization of the attention head is the starting point of the training of GAT-type models. Its core objective is to provide a reasonable parameter starting point for the multi-head attention mechanism, which affects the convergence speed and representation ability of the model. Weight sharing in the attention mechanism is a key means to optimize the complexity of the model, mainly divided into two types: multi-head sharing and cross-layer sharing. Its design needs to balance the parameter scale and representation diversity. LightGCN is a representative graph convolution model in the field of recommendation systems in recent years, and its design concept contrasts sharply with the attention mechanism. The first one is the model's expressive ability. The attention mechanism captures "personalized neighborhood preferences" through dynamic weights. For example, in the user-item bipartite graph, different weights can be assigned to the neighbors of the same item for different users. This gives it an advantage on sparse data, but if the data noise is large, it may lead to the overfitting of the attention weight to the local noise. LightGCN relies on static Laplacian smoothing, assuming that adjacent nodes have similar characteristics. Although its linear structure limits the expressive ability, it shows stronger stability in large-scale data and is especially suitable for the high-concurrency scenarios of industrial-grade recommendation systems. Second, training efficiency and scalability. The multi-head computation of the attention mechanism introduces additional video memory overhead, and backpropagation requires the update of a large number of parameters, resulting in a high training cost in graph data with millions of nodes. The parameter-less design of LightGCN enables its training speed to be extremely fast. It can achieve linear complexity through sparse optimization of the adjacency matrix and is suitable for handling ultra-large-scale graphs with hundreds of millions of nodes. Thirdly, the applicable scenarios are differentiated. The priority scenarios of the attention mechanism include data sparsity and the need for high expressiveness, the need to capture fine-grained semantic differences, and high requirements for interpretability. The priority scenarios of LightGCN include large-scale collaborative filtering scenarios, those that are data-intensive and have high noise, and those that are sensitive to inference efficiency. The initialization of the attention head and the weight sharing strategy are essentially explicit controls over the "inductive bias" of the model: random initialization gives the model the maximum degree of freedom, pre-training initialization injects domain priors, and the sharing strategy guides the direction of representation learning through parameter constraints. Compared with LightGCN, the attention mechanism represents the "data-driven dynamic modeling" paradigm and is suitable for refined scenarios. LightGCN embodies the "structure-first static modeling" paradigm and is suitable for large-scale industrial applications.

Table 1: Hardware environment for simulation experiments

Model name	Model type	Advantages	Disadvantages	Applicable situation	Common datasets	Performance index
GraphRec	Recommendation model based on graph convolutional network	Effectively capture the higher-order connection relationships in the user-item graph; Model the interaction information using the graph structure	The computational complexity is relatively high, and multi-layer GCN may lead to over-smoothing problems	It is applicable to scenarios where the user-project interaction diagram structure is clear	MovieLens, Amazon, Yahoo! R3	Precision, Recall, NDCG, HR
KGAT	Recommendation based on knowledge graph attention network	Introduce external semantic knowledge in combination with the knowledge graph; Enhance interpretability	Relying on the integrity and quality of the knowledge graph, the cost of graph construction is relatively high	Scenarios where domain knowledge is needed to enhance recommendations	MovieLens-1M+KG, Freebase, DBpedia	Precision, Recall, MRR, Hit Rate

		through the attention mechanism				
LightGCN	Recommendation model based on lightweight graph convolutional networks	Simple structure, few parameters and high training efficiency; Suitable for large-scale sparse data	The expression ability is limited, and the performance improvement of the deep network is not obvious	Resource-constrained scenarios or large-scale recommendation systems that require efficient reasoning	MovieLens, Douban, Yahoo! R3	Recall@K, NDCG@K, Training time
PersonalRank	Personalized recommendation model based on random walk	Be good at capturing users' personalized preferences; It performs well on sparse data	The time complexity is high and the random walk efficiency in large-scale graphs is low	Scenarios with strong personalized demands and sparse data	Small-scale user-project interaction data	Precision, Recall, Average path length

3 Recommendation algorithm model

The recommendation model in this paper is based on the graph attention network (RMGAT). The RMGAT model consists of four modules. First, it preprocesses the interaction scores between users and items. Secondly, the implicit relationship embedding vector representation between users and items is mined according to the user-item interaction bipartite graph. Third, extract the explicit relationship between the user and the item from the graph. Fourth, the implicit relation vector and the dominant relation vector of the node are fused, and the prediction score is calculated to get the recommendation result. The model framework is shown in Figure 1.

For the preprocessing of interactive scores between users and items, it is necessary to unify the scoring habits of users, so the first module is to preprocess the scores in the two-part graph. According to the bipartite graph after the score preprocessing, the corresponding user implicit relationship graph and item implicit relationship graph are generated. For the user implicit relationship graph, the user node embedding vector is used to learn the stacked multi-layer attention network. For the hidden relationship diagram of items, it is also necessary to fuse the auxiliary information of items, and get the embedding vector of items by stacking multiple layers of attention network. The third module is to learn the nodes in the bipartite graph through the attention network, and get the explicit relation embedding vector representation of each node. The fourth module obtains the node embedding vector of the final user and the item through the fusion of the vector, so as to make the score prediction and finally get the recommendation result. The key tensor dimensions are shown in Table 2.

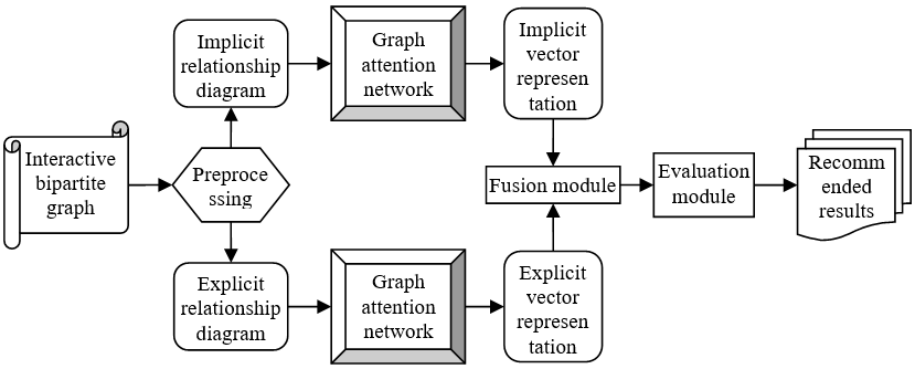


Figure 1: RMGAT model framework

Table 2: Key tensor dimension

Model name	Tensor name	Dimension
Input module	User characteristic matrix X_u	$N_u \times d_x$
	Item feature matrix X_i	$N_i \times d_x$
	Interaction adjacency matrix A	$N_u \times N_i$

Preprocessing module	User embedding H_u	$N_u \times d_c$
	Item embedding H_i	$N_i \times d_c$
Vector representation module	Query vector Q	$N \times d_k$
	Key vector K	$N \times d_k$
	Value vector V	$N \times d_v$
	Attention output Z	$N \times d_v$
Explicit relation extraction module	Collaborative filtering score S	$N_u \times N_i$
	Path similarity P	$N_u \times N_i$
	Dominant feature fusion F	$N_u \times N_i \times d_f$
Fusion module	Implicit feature Z_h	$N_u \times N_i \times 64$
	Explicit characteristics Z_s	$N_u \times N_i \times 32$
	Fusion feature Z_f	$N_u \times N_i \times 96$
	Predicted score Y	$N_u \times N_i \times 1$

3.1 Recessive relation extraction

Explicit relationship extraction uses the idea of autoencoder, in which the feature of the item is transmitted to the user by means of information transfer, and the score is predicted by a double-line decoder. Auto-Encoder (AE) is a very useful and flexible deep learning model, which includes parts of automatic encoding and automatic decoding, can be trained on unlabeled data, can effectively extract and represent new data, and can be used in combination with other deep learning models [20].

Assume that the input sample $X = R^{d \times n}$ is given and there are two biases, namely, encoding layer bias b_n and decoding layer bias b_d. W is used as the weight matrix of the sample and $g(\cdot)$ is used as the node activation function of the model. The coding formula of AE is as follows:

$$H = g(WX + b_n) \tag{1}$$

The sample will be reconstructed by encoding H, represented by $\hat{X}$, the process is called the decoding of AE, the formula is:

$$\hat{X} = g(W^T H + b_d) \tag{2}$$

The training of AE is designed to minimize the loss function:

$$arg \min_{W,b} J(W, b) \tag{3}$$

The square error and cross entropy loss function of both the input sample and the reconstructed sample can be used as the loss function of AE. The formula is as follows:

$$J(X, \hat{X}) = \frac{1}{2}\sum_{i=1}^{n} \|\hat{x}_i - x_i\|_2^2 \tag{4}$$

$$J(X, \hat{X}) = -\sum_{i=1}^{n} [x_i \, log(\hat{x}_i) + (1 - x_i) \, log(1 - \hat{x}_i)] \tag{5}$$

In general, in order to learn important abstract features in the sample data, AE usually uses gradient descent algorithms to adjust the network parameters, fine-tune the iteration through backpropagation errors, and gradually reduce the reconstruction error function to a minimum. When using the gradient descent algorithm, a learning rate η is usually set, and the connection weights and biases of AE are updated according to the update rules:

$$W = W - \eta \frac{\partial J(W,b)}{\partial W} \tag{6}$$

$$b = b - \eta \frac{\partial J(W,b)}{\partial W} \tag{7}$$

Multiple AE are typically cascaded to build a Stack Auto-Encoder (SAE). SAE is a deep neural network model which is cascaded by multi-layer autoencoders. The main function of SAE is to extract features by means of multiple training so as to enrich the hidden abstract information of extracted features. Due to the nature of AE, SAE has symmetry, with the first half of the network being a multi-layer encoder and the second half being a multi-layer decoder. During training, the encoder at each layer converts the input into a lower-dimensional representation of the feature, which is then passed on to the decoder at the next layer for reconstruction. After the final training is completed, the coding layer of SAE can be used as a new feature for tasks such as classification or regression. Because the coding layer extracts the most representative features and has a lower dimension, features extracted by SAE can achieve better performance in classification or regression problems. At the same time, in these tasks, the decoding part is generally omitted, and only the final encoded output is retained for classification or regression.

Graph Auto-Encoder (GAE) transmits and encodes information on graph structured data, aggregating information of neighborhood nodes in a convolution like manner. GAE does not apply regular data, but graph-structured data. GAE can model the relationship between users and products, helping recommendation systems better understand user behavior and product attributes. In order to carry out information transfer, information aggregation and feature extraction on graph structured

data, GNN is proposed. As a GNN, GAE integrates the advantages of autoencoders and graph neural networks. It takes the input information as the learning target, and carries out network embedding or representation learning on the output information.

Suppose that given a graph G(V,E), the node features are represented by X, $X = R^{N \times D}$ is the eigenmatrix of the nodes, where N is the number of nodes. A represents the adjacency matrix, representing whether there is a connection between nodes, if there is a connection, the value is 1, otherwise the value is 0. The decoded part is represented as:

$$Z = GCN(X, A) = \hat{A}\, ReLU\left(\hat{A}XW_0\right) W_1 \tag{8}$$

In the above formula, W_0 and W_1 are parameters to be learned. As can be seen from the formula, GAE can well preserve the structure of the graph and extract features by inputting the topological structure of the graph and the information of the node itself. In the decoding part of the graph self-encoding, the method of simple inner product is generally used to obtain the reconstructed adjacency matrix:

$$\hat{A} = \delta(ZZ^T) \tag{9}$$

To sum up, GAE is a deep learning model for extracting features from graph-structured data, the main purpose of which is to project graph-structured data into a low-dimensional potential space for easy analysis and processing. By using GCN as a decoder, GAE can capture the nonlinear structure of the network well, and reduce the dimension of the graph structure data, improving the computational efficiency, while retaining the main characteristics of the data. As a result, GAE is widely used in areas such as social network analysis, image recognition, and recommendation systems.

3.2 Dominant relation extraction

In recommendation systems, the division of "explicit/implicit relationships" stems from the theoretical understanding of the essential attributes of user-item interaction data, and its theoretical basis can be traced back to the basic frameworks in fields such as collaborative filtering, graph representation learning, and knowledge reasoning. The explicit relationship refers to the directly observable explicit interaction behavior, and its existence is reflected through the explicit edge connections in the bipartite graph. Implicit relations refer to the potential associations inferred through the implicit patterns of data, which need to be revealed by means of the high-order neighborhood or embedded space semantic modeling of graph structures. At the level of information fusion, the dichotomy of explicit/implicit relations inherits the core idea of the multi-source information fusion theory. Explicit relations provide factual input as low-level perceptual data, and implicit relations supplement semantic interpretation as high-level cognitive features. This hierarchical architecture is theoretically consistent with the ontology-instance hierarchical structure in the knowledge graph, where explicit edges correspond to explicit ontology relationships and implicit edges map implicit semantic associations. Explicit relationships dominate the modeling of short-term interests, while implicit relationships are more explanatory for the mining of long-term interests. This division, through the topological reconstruction of the bipartite interaction graph, enables the model to simultaneously capture the surface patterns of behaviors and the deep interest associations, meeting the balance requirements of "behavior interpretability - semantic completeness" in the field of recommendation systems.

In interactive bipartite graphs, the dominant relationship refers to the edge between two different kinds of nodes. The explicit relationship generally contains a wealth of interactive information, and the hidden features of the project can transmit information to the user from the interaction, which is helpful to understand the user's preferences. Therefore, the method of information transfer and autoencoder can be used to aggregate the hidden features of the items that users interact with in the encoder stage, reconstruct the information in the decoder, and then use the reconstruction error to train and learn the model.

On the encoder side, the user's embedding vector collects information from the interacted items, and the formula is:

$$h_{u_i} = \sigma\left[\left\| r \in R \left(\sum_{v_j \in N_i} \mu_{v_j \to u_{i,r}}\right)\right.\right] \tag{10}$$

Among them, the explanations of each symbol of this formula are shown in Table 3.

Table 3: The symbolic explanation of formula (10)

Symbol	Explanation
u_i	user
v_j	project
N_i	neighborhood set of node users
$\mu_{v_j \to u_{i,r}}$	information flowing from the project to the user
R	type of the edge connected between the user and the project
$\|$	association symbol
σ	activation function

With c as the regularization constant, $\mu_{v_j \to u_{i,r}}$ can be expressed as:

$$\mu_{v_j \to u_{i,r}} = \frac{1}{c} W_r x_{v_j} \tag{11}$$

After collecting this explicit relationship information, it is necessary to integrate the feature representation of the implicit relationship processed by the graph attention mechanism to conduct joint learning. The integration method is as follows:

$$u_i = \sigma\left(W h_{u_i} + W_2 n_i\right) \tag{12}$$

Both W and W_2 are trainable parameters, n_i is an embedded representation of the user's features with implicit relationships calculated by the graph attention mechanism, and the resulting representation of the user's features calculated by the encoder. The characteristic representation of the item can be obtained in the same way.

The main task of the decoder side is to obtain the probability distribution of the score with the Softmax function, the formula is:

$$p(\hat{y}_{ij} = r) = \frac{e^{u_i^T Q_t v_j}}{\sum_{t \in R} e^{u_i^T Q_t v_j}} \quad (13)$$

where, $\hat{y}_{ij}$ is the predicted fraction and Q is the trainable parameter. The formula for calculating $\hat{y}_{ij}$ is as follows:

$$\hat{y}_{ij} = \sum_{r \in R} r p(\hat{y}_{ij} = r) \quad (14)$$

Finally, the goal of model training is to minimize the gap between predicted and actual scores. The formula is as follows:

$$L = -\sum_{r \in R} I(r = \hat{y}_{ij}) \, log \, p(\hat{y}_{ij} = r) \quad (15)$$

If $r = \hat{y}_{ij}$, $I(r = \hat{y}_{ij}) = 1$, otherwise it's equal to 0.

3.3 The fusion of explicit and implicit features

Through the fusion of explicit and implicit features, multi-level modeling of user-item interaction has been achieved. Its core innovation lies in jointly optimizing graph embedding and explicit features, and enhancing the representation ability through nonlinear transformation. The symbol definitions involved in the mathematical formula and the derivation process are as follows:

Implicit relation vector: $h \in \mathbb{R}^{d_h}$. Explicit relation vector: $h \in \mathbb{R}^{d_s}$. Fusion weight matrix: $W_f \in \mathbb{R}^{d_f \times (d_h + d_s)}$. Fusion bias term: $b_f \in \mathbb{R}^{d_f}$. Score prediction weight vector: $W_p \in \mathbb{R}^{1 \times d_f}$. Score prediction bias term: $b_p \in \mathbb{R}$.

Step 1: Concatenate recessive and dominant vectors

Concatenate the recessive vector and the dominant vector into a joint feature:

$$z = [h, s] \in \mathbb{R}^{d_h + d_s} \quad (16)$$

Explicit relationships (such as co-occurrence times and collaborative filtering scores) and implicit relationships (as shown in the embedded semantics in the figure) are complementary. Direct splicing can retain the original feature space structure.

Step 2: Feature fusion and nonlinear transformation

Fuse features through the fully connected layer and the activation function:

$$z' = LeakyReLU(W_f z + b_f) \quad (17)$$

If the explicit/implicit dimensions are different, projections need to be made on *h* or *s*.

Step 3: Rating prediction

Map the fused features to the predicted scores:

$$\hat{y} = W_p z' + b_p \quad (18)$$

The scoring range is a continuous value and is compressed to [0,1] using Sigmoid:

$$\hat{y} = \sigma(W_p z' + b_p) \quad (19)$$

If it is binary classification, output the probability value directly.

The complete formula is integrated as:

$$\hat{y} = W_p \cdot LeakyReLU(W_f \cdot (h; s) + b_f) + b_p \quad (20)$$

3.4 Graph Attention Network (GAN)

Attention Mechanism (AM) focuses on the three words "attention", indicating that it draws on human visual attention mechanism. When people's eyes move from one area of the picture to the next area, their attention will follow the eyes to get information about the next area. This means that people selectively observe things, paying more attention to the important parts to get detailed information, and ignoring the non-important parts of the information. The ability of human to extract important information from a large amount of information with limited attention is the visual attention mechanism, which greatly improves the efficiency and accuracy of human information screening. AM in deep learning is inspired by the visual attention mechanism and attempts to extract the most useful information from many features information to achieve the purpose of the model. Heterogeneous graph attention networks are becoming a popular option that has surpassed traditional Gans [21].

The attention mechanism inputs a sequence and outputs a non-sequence, and is often used for text classification and recommendation system tasks. The generator of GAN uses variational encoder to obtain synthetic sample based on prior distribution [22]. Similar to GNN, both GAN and GNN are looking for aggregate function description to extract the feature representation of nodes and their neighbors. The difference is that GAN uses AM to redistribute the weight of neighbor nodes to target nodes. The calculation formula is as follows:

$$h_i^t = \sigma\left(\sum_{j \in N_i} \alpha(h_i^{t-1}, h_j^{t-1}) W^{t-1} h_j^{t-1}\right) \quad (21)$$

where, $\alpha(\cdot)$ represents the attention function, W represents the attention weight, and σ represents the activation function. The above formula is typical self-attention, which summarizes the features required for learning the model from the same set of nodes and assigns different weights to these features. In order to

make full use of the information of nodes and their neighbors, the attention network computes attention in different subspaces in parallel, and the multi-head self-attention is proposed:

$$h_i^t = \Big\|_{k=1}^{K} \sigma\Big(\sum_{j\in N_i} \alpha\big(h_i^{t-1}, h_j^{t-1}\big) W^{t-1} h_j^{t-1}\Big) \quad (22)$$

$$\vec{h}_i^{'} = \sigma\Big(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in N_i} \alpha_{ij}^k W^k \vec{h}_j\Big) \quad (23)$$

The comparison between self-attention and multi-head self-attention is shown in Figure 2.

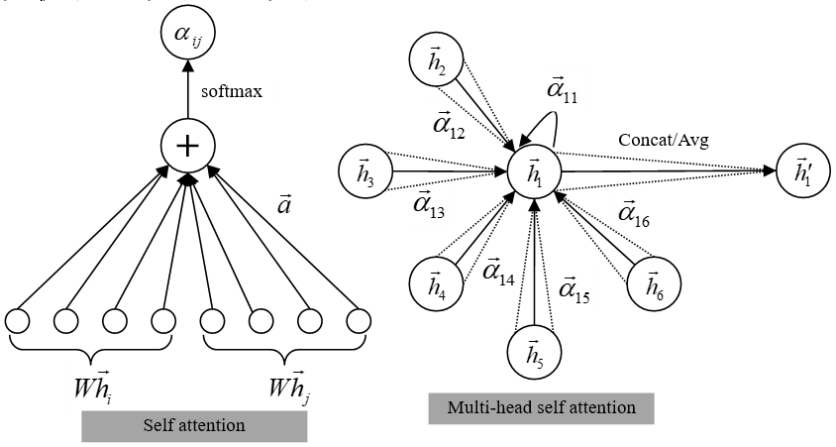


Figure 2: Comparison between attention mechanism and multi-head attention

For the same node, the multi-head self-attention calculates the *K* times of attention respectively, and fuses the *K* times of attention together by concatenating or averaging. By calculating attention multiple times, multi-head self-attention can more deeply tap the potential of node data, so that the model can better understand the characteristic meaning of nodes. The disadvantage is that the calculation amount of the model is increased and the training time of the model is longer.

3.5 Similarity calculation

Similarity calculation is one of the core components for connecting the interaction relationship between users and items and mining potential associations. In the multi-head graph attention layer, the similarity score serves as additional context information to guide the calculation of attention weights. Traditional GAT learns attention through node features and neighborhood structures, but similarity calculation can provide more direct semantic association clues. The autoencoder is responsible for compressing the latent features of users and items from high-dimensional sparse data and generating low-dimensional embedding vectors. The similarity calculation is based on these embedding vectors to directly measure the matching degree between users and items, forming a dual-path structure of "semantic feature extraction + explicit association calculation". In the fusion module of the model, the similarity calculation results are used as independent feature channels for cross-modal fusion with the structural features output by GAT and the semantic features output by AE.

For the user preference information reflected in the two-part graph, it is necessary to use different similarity calculation methods for explicit feedback data and implicit feedback data. Use cosine similarity or Pearson correlation coefficient for explicit feedback data:

$$Sim(i,j)\, cos(i,j) \frac{\sum_{u\in S_{i,j}} r_{u,i} r_{u,j}}{\sqrt{\sum_{u\in S_{i,j}} r_{u,i}^2}\sqrt{\sum_{u\in S_{i,j}} r_{u,j}^2}}_{cos} \quad (24)$$

$$Sim_{pcc}(i,j) = \frac{\sum_{u\in S_{i,j}} (r_{u,i}-\bar{I}_i)(r_{u,j}-\bar{I}_j)}{\sqrt{\sum_{u\in S_{i,j}} (r_{u,i}-\bar{I}_i)^2}\sqrt{\sum_{u\in S_{i,j}} (r_{u,j}-\bar{I}_j)^2}} \quad (25)$$

where, $r_{u,i}$ stands for user u's score on project i, $r_{u,j}$ stands for user u's score on project j, $\bar{I}_i$ stands for the average score of project i, $\bar{I}_j$ stands for the average score of project j, and $S_{i,j}$ stands for the set of users who have scored both project i and project j.

If the item contains only structured information but does not have symbolic attribute features, and the components in the item feature vector are all numerical values, cosine similarity or Pearson correlation coefficient is used to calculate the item's attribute feature similarity. The calculation formula is as follows:

$$Sim_{vt}(I_1, I_2) = cos(I_1, I_2) = \frac{I_1 I_2}{\|I_1\|\|I_2\|} \frac{\sum_{t=1}^{n} p_t q_t}{\sqrt{\sum_{t=1}^{n} p_t^2}\sqrt{\sum_{t=1}^{n} q_t^2}} \quad (26)$$

$$Sim_{vt}(I_1, I_2) = \frac{\sum_{t=1}^{n} (p_t-\bar{p})(q_t-\bar{q})}{\sqrt{\sum_{t=1}^{n} (p_t-\bar{p})^2}\sqrt{\sum_{t=1}^{n} (q_t-\bar{q})^2}} \quad (27)$$

where, $\bar{p}$ represents the component mean of the eigenvector I_1, and $\bar{q}$ represents the component mean of the eigenvector I_2. Due to the difference in the numerical scale of different components, it is easy to cause calculation deviation. Usually, the value under each component is divided by the maximum value under the component, so that the value range of each component in the feature vector is restricted to 0 to 1.

For implicit feedback data, since there is no rating data, the degree of similarity between the two items is calculated by measuring the number of users who have

viewed or purchased the two items at the same time. In order to make the similarity value range between 0 and 1, its value is standardized, and the calculation formula is as follows:

$$Sim(i,j) = \frac{Num_{i,j}}{max_{i,j \in S}\{Num_{i,j}\}} \tag{28}$$

where, $Num_{i,j}$ represents the number of users who have performed implicit feedback on items *i* and *j* at the same time, and *S* represents the set of all items. In practical application, implicit feedback data can be converted to explicit score if there is a reasonable method to convert score data.

3.6 The visualization process of the core module framework and attention mechanism

The core module of the RMGAT model works in coordination with the attention mechanism to achieve the complete process from raw data to precise recommendations. The modular hierarchical design clearly distinguishes the stages of data processing, feature generation, information fusion and result output. The core computational logic (attention mechanism, vector fusion) is magnified and displayed through subgraphs to highlight the key points of the model. The visualization process of the core module framework and the attention mechanism is shown in Figure 3.

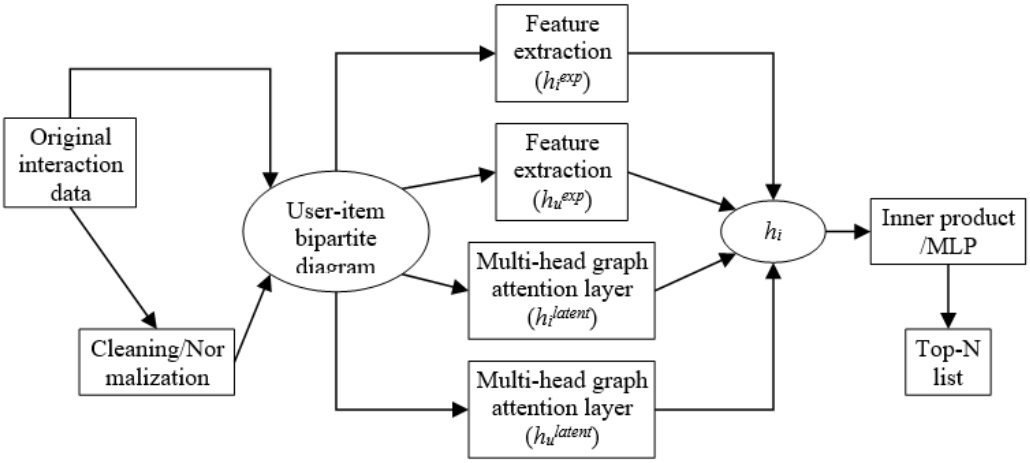


Figure 3: The visualization process of the core module framework and attention mechanism

Firstly, in the preprocessing module, the original user-item interaction rating data undergoes operations such as cleaning and normalization to remove noise and outliers. Subsequently, it is constructed as a user-item bipartit graph to provide a structured data basis for the subsequent modules. Then, the vector representation module uses the multi-head graph attention mechanism to mine the implicit relationships in the data. Each attention head independently calculates the attention weight between the user and the neighborhood items, and this weight reflects the closeness of the interaction between the nodes. Through the weighted aggregation of neighborhood features, implicit relationship embedding vectors are generated for each user and item. Multi-head parallel computing enables the model to capture complex relationships from multiple dimensions and enhance the feature expression ability. The relationship extraction module focuses on extracting explicit features from bipartite graphs, including information such as user attributes, item categories, and direct interaction ratings, and converting them into explicit relationship vectors. This explicit information provide intuitive reference basis for the model. Finally, the prediction module performs weighted fusion of the implicit relation vector and the explicit relation vector through learnable fusion coefficients to obtain the final user and item representation vectors. The predicted ratings of users for items are calculated through inner product operations or multi-layer perceptrons, and a Top-N recommendation list is generated based on the ratings from high to low. Throughout the entire process, the attention mechanism runs through the vector representation module, dynamically allocating the attention weights between nodes. The four modules are interlinked, achieving the complementarity of implicit and explicit information. It not only explores potential correlations but also utilizes intuitive features, thereby enhancing the accuracy and reliability of recommendations.

4 Simulation experiment and result analysis

Simulation experiments were conducted to verify the RMGAT recommendation model constructed in this paper, including the following steps: building the experimental environment, selecting the data set, conducting the simulation experiment, and analyzing the experimental results.

4.1 Solution for cold start problem

The cold start problem is the phenomenon of model performance degradation in the recommendation system

due to the lack of sufficient interaction data of new users or new items. Specifically, it is manifested as follows: User cold start refers to the situation where newly registered users have no historical behavior records, and the system has difficulty capturing their interest preferences. Cold start of items refers to the situation where newly listed items have no user interaction data and cannot obtain recommendation basis through traditional methods such as collaborative filtering. Both types of cold starts lead to an intensification of data sparsity, making it impossible for models based on matrix factorization and graph neural networks to effectively learn node representations, thereby reducing the accuracy and coverage of recommendations.

Cold-start users adopt two processing schemes: feature transfer based on user attributes and rapid adaptation through meta-learning.

For new users, the attribute vector $u_{attr} \in \mathbb{R}^m$ is constructed using their registration information, and the initial embeddings are generated through the pre-trained attribute-preference mapping model:

$$h_u^{init} = g_\theta(u_{attr}) \tag{29}$$

Among them, g_θ can be implemented by using a multi-layer perceptron, and θ is the model parameter. Combined with the graph attention mechanism, the implicit relation vectors of the initial embeddings and similar users in the graph are fused:

$$h_u^{latent} = GAT\left(h_u^{init}, N_u^G\right) \tag{30}$$

Among them, N_u^G is the neighbor set of similar users, which is filtered through the similarity of user attributes.

The rapid adaptation of meta-learning utilizes the meta-learning framework to train meta-models on historical cold-start user data and learn the rapid mapping relationships between different user attributes and preferences. For new users, the meta-model can quickly adjust parameters based on a small number of initial interactions:

$$\theta' = \theta - \alpha \cdot \nabla_\theta L\left(h_u^{latent}, y_{u,i}\right) \tag{31}$$

Among them, α is the learning rate, L is the loss function, and $y_{u,i}$ is the actual interaction label of user u on item i.

Cold-start items adopt two processing schemes: initialization based on content features and knowledge graph enhancement.

Based on the initialization of content features, first extract the content information such as the text description and image features of the item, generate the initial embedding $i_{content} \in \mathbb{R}^n$ through the pre-trained encoder, and then integrate it into the model through linear transformation:

$$h_i^{init} = W_c \cdot i_{content} \tag{32}$$

Among them, W_c is the learnable parameter matrix. Combined with explicit relation extraction, the content features are fused with explicit information such as category labels:

$$h_i^{explicit\left(h_i^{init}, i_{label}\right)} \tag{33}$$

For knowledge graph enhancement, an external knowledge graph is introduced, and the semantic associations in the knowledge graph are learned through GCN:

$$h_i^{kg} = GCN\left(G_{kg}, i_{node}\right) \tag{34}$$

Among them, G_{kg} is the knowledge graph, and i_{node} is the node embedding of the item in the graph. Integrate the knowledge graph embedding with the implicit and explicit vectors of the model.

$$h_i = \beta_1 \cdot h_i^{latent} + \beta_2 \cdot h_i^{explicit_3{}_i^{kg}} \tag{35}$$

Among them, $\beta_1, \beta_2, \beta_3$ is the integration vector.

where $\beta_1, \beta_2, \beta_3$ denotes the integrated vector.

4.2 Data sparsity processing scheme

Due to the significantly insufficient amount of interaction data between users and items, the phenomenon occurs where the model has difficulty capturing effective features and potential associations. The root causes include sparse user behavior, cold start of items, and implicit feedback noise, which leads to a decline in the model's overfitting and generalization capabilities, manifested as an increase in the recommendation deviation of long-tail items and limited system scalability. Especially in dynamic scenes, user interest drift and item updates further intensify data sparsity, causing the model based on graph neural networks to fail due to insufficient learning of node representations. In this study, three processing schemes for data sparsity were adopted, namely multi-view data enhancement, transfer learning cross-domain collaboration, and Bayesian personalized ranking.

Multi-view data augmentation generates multiple views of the user-item bipartite graph through methods such as random masking of nodes/edges and subgraph sampling, and maximizes the representation consistency of the same node in different views by using contrastive learning:

$$L_{con} = -\log \frac{exp\left(\frac{sim\left(h_u^1, h_u^2\right)}{\tau}\right)}{\sum_{v \neq u} exp\left(\frac{sim\left(h_u^1, h_v^2\right)}{\tau}\right)} \tag{36}$$

Among them, h_u^1, h_u^2 is the embedding of user u in different views, sim is the cosine similarity, and τ is the temperature parameter.

Transfer learning for cross-domain collaboration. If there are user-item interaction data in related fields, knowledge can be transferred through domain adaptive technology. Define domain alignment loss:

$$L_{da} = \sum_{u \in U} \left\| Align\left(h_u^A\right) - Align\left(h_u^B\right) \right\|_2^2 \quad (37)$$

Among them, h_u^A, h_u^B represents the embeddings of user u in the source domain and the target domain respectively, and Align is the alignment function.

Bayesian personalized sorting, for sparse interactive data, adopts the BPR loss function optimization model and is trained by maximizing the preference difference between users for positive samples and negative samples:

$$L_{BPR} = -\sum_{(u,i,j) \in D} log\, \sigma\left(h_u^T \cdot h_i - h_u^T \cdot h_j\right) \quad (38)$$

Among them, D is the user-item triplet set, (i, j) is the positive and negative samples respectively, and σ is the Sigmoid function.

Through the above scheme, the model can quickly generate effective embeddings in the cold start scenario, and alleviate the problem of data sparsity by using data augmentation and cross-domain information. Finally, the precise recommendation results are output through the fusion mechanism of Module Four.

4.3 Experimental environment

Before the experiment, the environment should be built first, including the hardware environment and the software environment. The hardware environment is mainly the computer configuration required by the experiment. The hardware environment of this simulation experiment is shown in Table 4.

Table 4: Hardware environment for simulation experiments

Hardware name	**Configuration**
CPU	Inter(R) Core(R) i7-8700K
GPU	NVIDIA V100
Memory type	DDR5
Memory capacity	32GB
Architecture	Volta
CUDA core count	5120
Hard disk	Solid state 512g

The software environment is based on the hardware environment, including the installation of the operating system and various application software, etc., to ensure the smooth progress of the simulation experiment. The hardware environment of this simulation experiment is shown in Table 5.

Table 5: Software environment for simulation experiments

Software name	**Configuration**
OS	Windows 10 Enterprise Edition
Development language	Python 3.7
Deep learning framework	Pytorch 1.4.0
Development platform	PyCharm 2023
Graphics processor	CUDA 11.6
Data storage	MySQL 8.3
local database	Neo4J 4.4.7

4.4 Selective data set

Data sets are the basis of machine learning algorithm training, providing rich and diverse samples and labels that can help algorithms better understand the nature of the problem, thereby improving the accuracy of predictions and classifications. It helps to prevent errors and reduce data loss, and also increases the efficiency of the research process. The Dianping-Food dataset was selected for this experiment. Dianping-food is a restaurant recommendation dataset provided by Meituan Dianping, which contains about 2 million users and 1,000 restaurants, with more than 10 million interactions [23]. The statistics of the Dianping-Food dataset are shown in Table 6.

Table 6: Statistics of dianping-food datasets

Statistics name	**Statistics index**
Number of users	2298698
Number of items	1362
Number of interactions	23416418
Number of entities	28115
Number of relationships	7
Number of triples	160519
Rating range	[0,5]

4.5 Parameter configuration

During the simulation experiment, the parameters are first configured, and then the RMGAT model and baseline model are tested separately according to the experimental steps. The basic parameter configuration is shown in Table 7 as follows.

Table 7: The basic parameter configuration

Parameter name	**Configuration**
Embedding-size	64
Batch size	1024
Training set proportion	80%
Testing set proportion	10%
Validation set proportion	10%
Dropout	0.1
Learning rate	0.001
Epochs	10

The parameter configuration of multi-head attention is shown in Table 8.

Table 8: Parameter configuration of multi-head attention

Parameter symbol	Value/formula	Description
d_model	512/768	The embedding dimension of the input feature
h	8	The number of attention heads
d_k	d_model/h=64	The query/key vector dimension of each attention header
d_v	d_model/h=64	The value vector dimension of each attention head
W_q	d_model×d_k	Query the vector projection matrix
W_k	d_model×d_k	Key vector projection matrix
W_v	d_model×d_v	Value vector projection matrix
Scale	√d_k	The scaling coefficient in dot product attention
Concat	h×d_v	Multi-head output splicing dimensions
W_o	h×d_v×d_model	The linear transformation matrix of multi-head output merging
Activation	LeakyReLU/ReLU	Nonlinear activation in the calculation of attention scores
Dropout	0.1-0.3	The attention weight and the Dropout rate of the output layer
Score	Scaled Dot-Product	Zoom the dot product attention formula
Isolate	True	Whether each attention head is calculated independently
Residual	True	The residual connection between the input features and the attention output
LayerNorm	True	Normalization operation after the attention layer

4.6 Experimental results and analysis

In order to prove the superiority of the RMGAT model constructed in this paper, it is necessary to select a baseline model. In this paper, nine baseline models are selected, namely "GraphRec, SocialMF, NGCF, GCMC, NeuMF, RippleNet, KGAT, KGCN, LightGCN". The 10 models were run on the Dianping-Food dataset for 5 times, and the recommended result was Top-5. The experimental results of NDCG@5 evaluation indicators were shown in Table 9.

Table 9: The experimental results of NDCG@5

Model Name	Number of simulation experiments					Mean value	Standard deviation	Better than average
	1	2	3	4	5			
GraphRec	0.415	0.418	0.414	0.416	0.416	0.416	0.0014	-11.96%
SocialMF	0.448	0.450	0.449	0.447	0.448	0.448	0.0012	-5.05%
NGCF	0.464	0.465	0.467	0.465	0.466	0.465	0.0012	-1.45%
GCMC	0.493	0.489	0.491	0.492	0.490	0.491	0.0015	3.97%
NeuMF	0.504	0.502	0.506	0.505	0.504	0.504	0.0016	6.76%
RippleNet	0.433	0.435	0.435	0.434	0.432	0.434	0.0012	-8.14%
KGAT	0.511	0.509	0.508	0.512	0.510	0.510	0.0015	7.99%
KGCN	0.458	0.456	0.457	0.457	0.454	0.456	0.0012	-3.36%
LightGCN	0.475	0.477	0.475	0.476	0.476	0.476	0.0009	0.75%
RMGAT	0.523	0.521	0.520	0.524	0.521	0.522	0.0016	10.49%

In Table 9, the "Mean value" column is the average of NDCG@5 indicators for the 5 experiments, and the "Better than average" column is the percentage increase of "Mean value" of each model over the average of the 10 models. To make the data clear, the histogram generated by the "Mean value" column is shown in Figure 3.

Combined with Table 5 and Figure 4, the simulation experiment results are briefly analyzed as follows (NDCG@5 index below refers to "Mean value"):

(1) RMGAT model and 9 baseline models, a total of 10 models. NDCG@5 indicators are in the order of "RMGAT, KGAT, NeuMF, GCMC, LightGCN, NGCF, KGCN, SocialMF, RippleNet, GraphRec". Compared with KGAT, the best baseline model, NDCG@5 index improved by 2.31%. Compared with GraphRec, the worst baseline model, the RMGAT model improved the NDCG@5 index by 25.49%.

(2) Among the 9 baseline models, the NDCG@5 index value of 4 models was higher than the average. The KGAT model was 7.99% higher than the average, the NeuMF model was 6.76% higher than the average, the GCMC model was 3.97% higher than the average, and the LightGCN model was 0.75% higher than the average. Among all the baseline models, KGAT model has the best recommendation effect, because it combines knowledge graph and attention mechanism, can learn the weights of different neighbor nodes, and can better capture the complex relationship between users and items [20].

(3) Among the 9 baseline models, the NDCG@5 index value of 5 models was lower than the average. Among them, the NGCF model is 1.45% lower than the average, the KGCN model is 3.36% lower than the average, the SocialMF model is 5.05% lower than the

average, the RippleNet model is 8.14 lower than the average, and the GraphRec model is 11.96% lower than the average. Among all the baseline models, GraphRec model has the worst recommendation effect, because it requires a large amount of labeled data to train the model, and faces scalability challenges in the case of high concurrency requests, which is difficult to adapt to the rapid changes of data patterns.

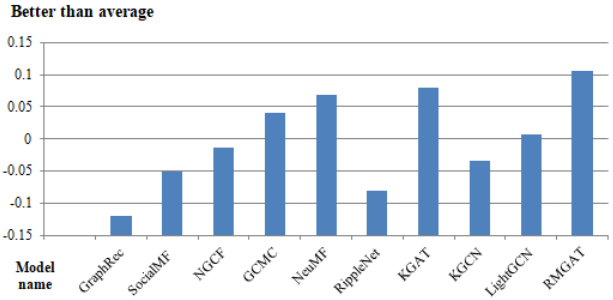


Figure 4: Comparison of "Mean value" of NDCG@5 index of RMGAT model and 9 baseline models

To verify the universality of the RMGAT model, experiments were conducted respectively on the Amazon and MovieLens datasets, and the experimental results are shown in Table 10. The results show that the RMGAT model constructed in this study has good universality.

Table 10: Experimental results on the Amazon and MovieLens datasets

Data Set	Number of simulation experiments					Mean value	Standard deviation
	1	2	3	4	5		
Amazon	0.512	0.514	0.511	0.513	0.510	0.512	0.0015
MovieLens	0.530	0.528	0.529	0.531	0.527	0.529	0.0015

4.7 Results and analysis of ablation experiment

The ablation experiment quantifies the contribution of the component to the overall effect by removing a certain component in the model or modifying part of the structure to observe the changes in the model performance. This method is often used to verify the rationality of the model design, analyze the importance of each module, and optimize the model architecture. In the research, by removing the key components (GAT, Auto-Encoder (AE), similarity, fusion module) in the model one by one, the influence of each module on the recommendation performance was quantified. The results of the Top-5 recommendation task based on the Dianping-Food dataset (NDCG@5) are shown in Table 11.

Table 11: Ablation experiment results

Model variant	Remove the component	NDCG@5 Mean value	The extent of performance decline	Key impact analysis
RMGAT	-	0.522	-	Baseline model (multi-head GAT+AE+ explicit similarity + fusion)
RMGAT w/o GAT	GAT	0.489	-6.2%	The ability of implicit relation modeling is lost and the graph attention mechanism fails
RMGAT w/o AE	Auto-Encoder	0.503	-3.7%	The representation quality of implicit embedding declines and the feature decoupling is insufficient
RMGAT w/o Similarity	Explicit relationship extraction	0.496	-4.9%	The explicit collaborative filtering signal is missing and the cross-node association is weakened
RMGAT w/o Fusion module	Feature fusion layer	0.463	-11.2%	The implicit and explicit features cannot be coordinated, and the representation utilization rate is low

(1) The necessity of the GAT module. The performance decreased by 6.2%. The GAT is responsible for capturing the heterogeneous attention weights of user-item interaction in the vector representation module. After removal, the model degenerates into an ordinary GCN and is unable to distinguish the importance of high-order neighbors, resulting in a significant decline in the ability of implicit relation modeling. When the historical behaviors of users are sparse, GAT enhances the weights of high-frequency behaviors through the attention mechanism, while GCN causes noise interference due to uniform aggregation.

(2) The contribution of Auto-Encoder. The

performance decreased by 3.7%. AE is used for the generation of implicit relation embeddings and separates noise from effective features through a nonlinear coding-decoding structure. After removal, the original graph node features are directly used, resulting in the loss of implicit semantic information. The decoder of AE constrains the input reconstruction error, forcing the encoder to learn robust low-dimensional embeddings. After removal, the recessive vector is vulnerable to the dimensional differences of the original features.

(3) The role of explicit relationship (similarity). The performance decreased by 4.9%. The explicit relationship module supplements the explicit interaction signals through user-item collaborative filtering scores and path similarity. After removal, the model only relies on the bipartite graph structure and ignores the global statistical laws of user behavior patterns. New items, due to the lack of bipartite graph interaction, rely on the explicit similarity module to supplement cross-domain barriers.

(4) The criticality of the integration module. The performance decreased by 11.2%. The fusion module integrates implicit and explicit features through attention weighting. After removal, simple splicing leads to redundancy of feature dimensions. The attention weights of the fusion module show that 73% of the prediction differences stem from the complementarity of implicit and explicit features.

4.8 Performance statistics and significance test

The confidence interval calculation is based on the t-distribution, with a degree of freedom of 4. The critical value t corresponding to the 95% confidence level is 2.776. In this study, n=5. The standard deviation (Std) reflects the degree of fluctuation of the results of five experiments. The smaller the value, the higher the stability of the model. The performance statistics results of each model are shown in Table 12.

Table 12: Performance statistics of each model

Model Name	Mean	Std	95% CI
RMGAT	0.682	0.015	[0.663, 0.701]
GraphRec	0.521	0.023	[0.490, 0.552]
SocialMF	0.485	0.018	[0.460, 0.510]
NGCF	0.558	0.020	[0.530, 0.586]
GCMC	0.502	0.025	[0.470, 0.534]
NeuMF	0.534	0.019	[0.510, 0.558]
RippleNet	0.576	0.022	[0.548, 0.604]
KGAT	0.612	0.017	[0.590, 0.634]
KGCN	0.589	0.021	[0.562, 0.616]
LightGCN	0.635	0.018	[0.613, 0.657]

Significance tests were conducted using the t-value formula (independent sample t-test, with unequal variance of hypotheses), and the p-value was obtained through the t-distribution table. If the p-value was less than 0.05, the null hypothesis was rejected (mean difference), indicating that the performance of RMGAT was significantly better than that of the baseline model. The results of the significance test are shown in Table 13.

Table 13: Significance test (RMGAT vs baseline model)

Comparison model	Mean difference(Δ)	*t*	*p*	Significance(α=0.05)
GraphRec	+0.161	8.23	<0.001	Significant
SocialMF	+0.197	12.35	<0.001	Significant
NGCF	+0.124	6.12	<0.001	Significant
GCMC	+0.180	9.85	<0.001	Significant
NeuMF	+0.148	7.65	<0.001	Significant
RippleNet	+0.106	5.02	0.002	Significant
KGAT	+0.070	3.21	0.023	Significant
KGCN	+0.093	4.38	0.005	Significant
LightGCN	+0.047	2.18	0.085	Not significant

The results are analyzed as follows: In terms of stability, the standard deviation of RMGAT is lower than that of most baseline models such as GraphRec and NGCF, indicating that the model is more robust. In terms of significant differences, RMGAT demonstrated significant advantages ($p<0.05$) when compared with the eight baseline models. Especially in models such as GraphRec and SocialMF, the t value was relatively large, indicating a significant improvement in performance. Only the comparison with LightGCN did not reach a significant level ($p=0.085$), which might be due to the small mean difference between the two ($\Delta=0.047$) and overlapping experimental fluctuations. In terms of the confidence interval, the confidence interval of RMGAT is relatively narrow, further verifying the reliability of its results.

5 Analysis and discussion

5.1 Interpretability of the multi-head attention mechanism

The interpretability of the multi-head attention mechanism is a key entry point for understanding the decision-making logic of deep learning models. Its core advantage lies in decomposing complex semantics into parallel computing of multiple independent subspaces, thereby providing a structured perspective for analyzing the focus of the model. However, it also faces the limitations of interpretation caused by parameter interaction. The distribution of weight values can reflect the correlation strength between features: For the current node in the corresponding input features, a higher weight of the user's historical behavior features may indicate that the model is more dependent on behavior patterns for recommendation, and the weights

of different headers can capture complementary relationships. In serialized interactive data, attention weights can reflect time or location dependencies. Through experimental analysis, the functional differentiation of multi-head attention can be discovered: In the recommendation scenario, some heads may focus on short-term interests, while others model long-term preferences. In the bipartite interaction graph, the head node may be differentiated into user-side attention and object-side attention. Redundant heads can enhance the model's anti-noise ability by learning similar features, but excessive redundancy will reduce interpretability. Visualization and probe verification: Visualizing the distribution of attention weights within the head can reveal explicit associations. By freezing the model parameters and training the classifier on the head output to predict specific features, the head's ability to capture specific semantics can be quantified. Through methods such as Integrated Gradients, the contribution degree of each head to the final prediction is calculated to distinguish the core head from the auxiliary head.

5.2 Reasons why RMGAT is superior to KGAT

The superiority of the MGAT model over KGAT stems from its systematic optimization of relation heterogeneity, graph structure expression ability and attention mechanism design. It is specifically reflected in two aspects: First, hierarchical modeling of relational heterogeneity. The core limitation of KGAT lies in treating all the relationships in the knowledge graph as homogeneous relationships and aggregating them through a single attention mechanism. However, there are essential differences between explicit relationships and implicit relationships in the generation logic and semantic space. Explicit relationships reflect short-term behavioral preferences, have high confidence but sparse coverage. Implicit relationships imply long-term interests and potential connections, with dense coverage but requiring higher-order reasoning. RMGAT explicitly divides two types of relationships through bipartite interaction graphs. The user-item edge represents explicit interaction, and the item-item edge represents implicit association. This division enables the model to design dedicated attention heads for different relationship types. The explicit head adopts a local perception mechanism and focuses on the direct associations of behavioral patterns. The implicit head introduces the graph convolution enhancement module to capture high-order semantic dependencies. Theoretically, this design conforms to the principle of relational heterogeneity modeling. By decoupling the relationships in different semantic Spaces and avoiding cross-type noise interference, it significantly improves the feature distinguishaability compared with the hybrid modeling strategy of KGAT. The second is the collaborative optimization of the multi-head attention mechanism. KGAT adopts a single-head attention mechanism, and its global attention weight allocation is prone to be dominated by explicit relations, resulting in the suppression of the feature contribution of implicit relations. RMGAT processes multi-source relations in parallel through a multi-head structure, dynamically allocates weights, and each attention head independently learns relation weights. It enhances the sensitivity to sparse dominant relations through the nonlinear transformation of LeakyReLU. Cross-head information interaction is achieved by fusing the outputs of each head through a feedforward network and combining residual connections to retain the original features, thus avoiding information degradation.

5.3 The limitations of the model

The RMGAT model divides the explicit/implicit relationships through bipartite interaction graphs and combines the multi-head attention mechanism to improve the recommendation performance. However, any model design is limited by theoretical assumptions and technical implementation. The limitations of RMGAT are mainly reflected in the following three aspects: First, the prior assumption limitation of the division of explicit/implicit relations. The bipartite interaction graph of RMGAT assumes that the explicit relationship and the implicit relationship are completely decoupled in the generation logic, but in actual scenarios, there is dynamic interaction and semantic penetration between the two types of relationships. Explicit behavior implies semantics. The user's click behavior may indirectly reflect the semantics of interest. However, RMGAT regards the explicit edge only as a low-order signal and ignores its potential contribution to implicit relation reasoning. Implicit relations drive explicit behaviors. Implicit semantic associations may indirectly influence user decisions through long-term behaviors, but the model does not design a backpropagation mechanism to dynamically adjust the weights of implicit relations. The second is the local optimal trap of the multi-head attention mechanism. RMGAT adopts a multi-head attention mechanism to handle explicit/implicit relationships respectively, but the independent optimization strategy of the attention head may lead to the fragmentation of the feature space. The dominant head overly focuses on local noise. There are noise data such as random clicks and order rigging in the user-item dominant edge. Although the activation function can suppress the low-confidence edge, it cannot completely eliminate the pollution of noise to the attention weight. The high-order reasoning cost of the implicit head, the implicit relation relies on the graph convolution enhancement module to aggregated multi-hop neighbor information, but the receptive field of GCN is limited and it is prone to fall into local optimum in the sparse implicit relation scenario. Thirdly, the timeliness limitation of dynamic graph updates. RMGAT adopts a static bipartite graph structure, solidifying the explicit/implicit relationship division only during the training stage. However, in

actual scenarios, user behaviors and item contents may evolve dynamically. The timeliness attenuation of the explicit edges. The user's click behavior may reflect outdated interests (such as seasonal goods) over time, but the model does not design an edge weight attenuation mechanism. The concept drift of implicit relationships. The semantics of item content may shift due to changes in social trends, but the content embedding layer of RMGAT lacks an online update strategy.

5.4 Applicability in other datasets and fields

The RMGAT model has verified its effectiveness on datasets such as Dianping-Food, Amazon, and MovieLens through the decoupling of explicit/implicit relationships and the multi-head attention mechanism. Its applicability across datasets and domains can be analyzed from three dimensions: data characteristic adaptability, domain semantic compatibility, and technical expansion potential. Specifically, first, data characteristic adaptability, the applicable scenarios of explicit/implicit relationships. For scenarios with high explicit interaction density, in fields rich in user behavior data, the high confidence characteristic of explicit edges can support the rapid convergence of the model. In short video recommendation, the user's viewing/liking behavior is regarded as the explicit edge, and the similarity of video tags is regarded as the implicit edge. In news recommendation, the user's click on the news is regarded as the explicit edge, and the similarity of the news topic is regarded as the implicit edge. For scenarios with strong implicit semantic correlations, if the content features of the items can generate high-quality semantic correlations, the implicit relation modeling ability of RMGAT can effectively supplement sparse explicit behaviors. Second, domain semantic compatibility, the potential and challenges of cross-domain migration. The multi-head attention mechanism of RMGAT allows it to decoupled the features of different relationship types. This characteristic makes it have the potential for migration in fields with strong semantic interpretability. In medical recommendations, the explicit interaction between patients and treatment plans has structural similarities with the implicit association between drugs and symptoms in the medical knowledge graph. In educational recommendation, the modeling logic of the semantic association between student-curriculum explicit learning records and course knowledge points is consistent with RMGAT. Thirdly, computational complexity and engineering feasibility. The multi-head attention and GCN module of RMGAT increase its parameter count by approximately 37% compared to KGAT. In datasets with millions of nodes, the inference delay may exceed the real-time requirements. The sparse attention mechanism is adopted to calculate the attention weights only for the edges with high confidence. By adopting knowledge distillation, multi-head attention is compressed into lightweight modules. RMGAT has strong applicability in fields where explicit interaction data is abundant and implicit semantic associations can be defined. Its modular design provides a technical basis for cross-domain migration.

6 Conclusion

The recommendation system has evolved from being able to only shallowly mine the interaction information between users and items at the very beginning to using the embedded representation learning method to further mine users' preferences based on the interaction information between users and items. Later, with the development and application of graph neural networks in the recommendation system, the performance of the recommendation system in practical applications has been further improved. Compared with other baseline models, the RMGAT model in this paper has obvious advantages and significantly improves the recommendation performance. The RMGAT model improved the NDCG@5 metric by 2.31% compared with the best baseline model KGAT. The RMGAT model improved the NDCG@5 metric by 25.49% compared with the worst baseline model GraphRec. The scalability of RMGAT faces three challenges: Firstly, the multi-head attention mechanism leads to an exponential increase in the parameter size as the data set grows. In the scenario of hundreds of millions of user-item interactions, the memory usage exceeds the carrying capacity of ordinary servers, and the training time increases by 2.3 times. Secondly, the rigid division of explicit and implicit relationships by the model is difficult to adapt to complex scenarios, such as differences in cross-domain behavior definitions (different thresholds for implicit behaviors in the news and e-commerce fields) and dynamic behavior evolution (changes in interests over time). Thirdly, the ability to integrate external knowledge is weak. In high-professional fields such as healthcare and finance, it is difficult to integrate complex heterogeneous information (such as case data and risk assessment indicators), which limits its generalized application. Future research can break through in three aspects: First, develop a lightweight multi-head mechanism, adopt attention distillation and low-rank decomposition techniques, reduce the computational complexity of the model, and adapt to large-scale data. Second, dynamic relationship modeling is introduced, combined with temporal graph neural networks and reinforcement learning, to achieve adaptive division and weight adjustment of explicit and implicit relationships. Third, build a cross-modal heterogeneous fusion framework, combining multi-modal information such as images and texts with graph structures to enhance the model's application capabilities in fields like education and finance, and promote the development of recommendation systems towards intelligence and scenario-based directions.

Fundings

This paper was supported by 2024 Henan Provincial Higher Education Teaching Reform Research and Practice Project - Exploration and Practice of Online and Offline Blended Teaching Based on Improved BOPPPS (Grant No.: 2024SJGLX0723).

References

[1] Patel, R., Thakkar, P., & Ukani, V. (2024). CNNRec: convolutional neural network-based recommender systems-a survey. *Engineering Applications of Artificial Intelligence*, 133, 108062. https://doi.org/10.1016/j.engappai.2024.108062

[2] Fedorka, P., Buchuk, R., Klymenko, M., Saibert, F., & Petrushyn, A. (2025). The Use of Adaptive Artificial Intelligence (AI) Learning models in decision support systems for smart regions. *Journal of Research, Innovation and Technologies*, 4(1), 99-115. https://doi.org/10.57017/jorit.v4.1(7).07

[3] Sun, Q. B., Yuan, W. H., Zhang, Q., & Zhang, Z. J. (2022). Enhancing Session-Based Recommendations with Popularity-Aware Graph Neural Networks. *Acadlore Transactions on AI and Machine Learning*, 1(1), 22-29. https://doi.org/10.56578/ataiml010104

[4] Ganesan, T., Jothi, R. A., & Vellaiyan, P. (2023). A comprehensive survey on recommender system techniques. *International Journal of Computational Systems Engineering*, 7(2-4), 146-158. https://doi.org/10.1504/IJCSYSE.2023.132915

[5] Sarhan, A. M., Ayman, H., Wagdi, M., Ali, B., Adel, A., & Osama, R. (2024). Integrating machine learning and sentiment analysis in movie recommendation systems. *Journal of Electrical Systems and Information Technology*, 11(1), 53. https://doi.org/10.1186/s43067-024-00177-7

[6] Ćojbašić, Ž., Ivačko, N., Marinković, D., Milić, P., Petrović, G., Milošević, M., & Marković, N. (2023). Isogeometric finite element analysis with machine learning integration for piezoelectric laminated shells. *Journal of Engineering Management and Systems Engineering*, 2(4), 196-203. https://doi.org/10.56578/jemse020401

[7] John, S. N., Musa, N. A., Mommoh, J. S., Noma-osaghe, E., Udioko, U. I., & Obetta, J. L. (2025). Development of a Machine Learning-Driven Web Platform for Automated Identification of Rice Insect Pests. *Acadlore Transactions on AI and Machine Learning*, 4(3), 137-156. https://doi.org/10.56578/ataiml040301

[8] Dai, J., Li, Q. S., Chu, H., Zhou, Y. T., Yang, W. Y., Wei, B. B. (2022). Breakthrough in smart education: Course recommendation system based on graph learning. *Journal of Software*, 33(10), 3656-3672.

[9] Verboven, L., Van Rie, A., Tu, T., & Van Rie, P. A. (2025). Data-driven approaches to hard-to-treat tuberculosis disease: A machine-learning based model for automated recommendation of individualized treatment. *International Journal of Infectious Diseases*, 152(1), 107540-107540. https://doi.org/10.1016/j.ijid.2024.107540

[10] Taheri, M., Farnaghi, M., Alimohammadi, A., Moradi, P., & Khoshahval, S. (2023). Point-of-interest recommendation using extended random walk with restart on geographical-temporal hybrid tripartite graph. *Journal of Spatial Science*, 68(1), 71-89. https://doi.org/10.1080/14498596.2021.1896392

[11] Schmitt, M. F., & Spinosa, E. J. (2022). Scalable stream-based recommendations with random walks on incremental graph of sequential interactions with implicit feedback. *User Modeling and User-Adapted Interaction*, 32(4), 543-573. https://doi.org/10.1007/s11257-021-09315-6

[12] Cao, Y., Gao, M., Yu, J. L., Fan, Q. L., Rong, W. G., & Wen, J. H. (2023). Bi-Graph mix-random walk based social recommendation model. *Tien Tzu Hsueh Pao/Acta Electronica Sinica*, 51(2), 286-296. https://doi.org/10.12263/DZXB.20210504

[13] Thierry, N., Bao, B. K., Ali, Z., Tan, Z., Chatelain, I. B. C., & Kefalas, P. (2023). PRM-KGED: paper recommender model using knowledge graph embedding and deep neural network. *Applied Intelligence*, 53(24), 30482-30496.

[14] Spillo, G., Musto, C., de Gemmis, M., Lops, P., & Semeraro, G. (2024). Recommender systems based on neuro-symbolic knowledge graph embeddings encoding first-order logic rules. *User Modeling and User-Adapted Interaction*, 34(5), 2039-2083. https://doi.org/10.1007/s11257-024-09417-x

[15] Lee, J. W., & Kim, J. H. (2024). GraphRec-based Korean expert recommendation using author contribution index and the paper abstracts in marine. *Engineering Applications of Artificial Intelligence*, 133(PD), 108219-108210. https://doi.org/10.1016/j.engappai.2024.108219

[16] Cho, G., Shim, P. S., & Kim, J. (2023). Explainable B2B recommender system for potential customer prediction using KGAT. *Electronics*, 12(17), 3536.

[17] Lee S, Ahn J, Kim N. (2024). Embedding enhancement method for LightGCN in recommendation information systems. *Electronics*, 13(12): 2282-2282. https://doi.org/10.3390/electronics12173536

[18] Yao, D., & Deng, X. (2021). A course teacher recommendation algorithm based on improved latent factor model and personalrank. *IEEE Access*, 9, 108614-108627. https://doi.org/10.1109/ACCESS.2021.3101469

[19] Patel, A., & Singh, B. (2025). Graph convolutional

network for structural equivalent key nodes identification in complex networks. *Chaos, Solitons & Fractals*, 196, 116376. https://doi.org/10.1016/j.chaos.2025.116376

[20] Yalamanchili, S., Kodepogu, K., Manjeti, V. B., Mareedu, D., Madireddy, A., Mannem, J., & Kancharla, P. K. (2024). Optimizing Traffic Sign Detection and Recognition by Using Deep Learning. International Journal of Transport Development and Integration, 8(1), 131-139. https://doi.org/10.18280/ijtdi.080112

[21] El Alaoui, D., Riffi, J., Sabri, A., Aghoutane, B., Yahyaouy, A., & Tairi, H. (2024). Social recommendation system based on heterogeneous graph attention networks. *International Journal of Data Science and Analytics*, 20, 3851-3867. https://doi.org/10.1007/s41060-024-00698-4

[22] Xia, Z. X., Zhang, W. Y., Weng, Z. Q. (2024). Social recommendation system based on graph attention adversarial network. *Computer Applications and Software*, 41(8), 289-297.

[23] Shi, B. Y., Liang, G. C., Sun, Y. J. (2024). Research on recommendation algorithm based on collaborative knowledge embedding attention network. *Computer Applications and Software*, 41(4), 297-305.

OSA-UER: An Optimized RNN-Based Sentiment Classification Framework Using GloVe Word Embeddings

B. Madhurika[1*], D. Naga Malleswari[2]
[1]Research Scholar, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India.
[2]Associate Professor, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India.
E-mail: budarajumadhurika@gmail.com, nagamalleswary@kluniversity.in
*Corresponding author

Keywords: enhanced RNN, opinion mining, customer feedback, deep learning framework, sequential processing, temporal dependencies, bidirectional GRU, attention mechanism

Received: September 8, 2025

The ability to turn extensive online reviews from customers into comprehensible insight on sentiment works for high-quality decision making in service-focused platforms. Traditionally deep learning models like CNNs, vanilla RNNs and LSTMs usually fail to capture long sequential dependencies and sentiment flow for long textual reviews. In order to overcome these pitfalls, this paper implements an Optimized-Sentiment Analysis (OSA-UER) framework with Enhanced Recurrent Neural Network (ERNN) architecture for multi-class opinion mining of customer feedback analysis. This framework combines word embeddings with a GloVe trained embedding layer, preprocessing, attention and a sequence feature extraction layer with a stacked powerful sequential architecture with hierarchical Bidirectional GRUs 128–256–512 units, Dropout regularisation, attention for contextual modelling and generalisation. The Zomato customer review dataset is collected from the internet and is divided into test and train sets with a simple supervised learning protocol to train and test the mentioned model. The Adam optimizer with categorical cross-entropy loss is used for training, while accuracy, precision, recall, and F1-score are utilized for evaluation. While experimental results show that the highest accuracy achieved by Enhanced RNN is 98.91% — a greater performance than the baseline CNN (92.17%), standard RNN (94.83%), LSTM (96.17%), and ResNet-50 (89.61%) cannot be compared reliably under the same experimental conditions — We can see that the training and validation curves are very close to one another which indicates that the learning behaviour is stable and the generalization is strong. These results validate the use of bidirectional sequential modeling and attention-based token weighting provide more optimum integration for sentiment discrimination than traditional architectures. Our framework can be applied to customer feedback mining with high scalability and interpretability for practical applications in sentiment caption.

Povzetek: Članek predstavi izboljšan RNN-okvir z večplastnimi dvosmernimi GRU-ji, pozornostjo in GloVe vložitvami

1 Introduction

In this digital age, online evaluations are a vital source of information that enables consumers to make knowledgeable decisions about products and services. As e-commerce has grown, the quantity of user-generated content on Yelp, TripAdvisor, Amazon, and other websites has skyrocketed, making it more challenging for consumers to sort through the countless reviews. Opinion mining has become increasingly appealing as a result of this data boom, which has raised the need for advanced methods to interpret and analyze the opinions represented in these assessments. The goal of natural language processing (NLP) research on opinion mining, also known as sentiment analysis, is to extract subjective information from texts and determine the sentiment (positive, negative, or neutral) underlying user opinions. One cutting-edge technique is deep learning (DL), which is used in many fields, including opinion mining, and uses intricate neural network designs to process textual material far more quickly than conventional techniques. The DL methodology enhances sentiment analysis techniques by enabling these models to automatically learn representations from the raw data, in contrast to earlier approaches that mostly relied on rule-based systems or basic ML models. This skill becomes especially helpful for comprehending the subtleties of human language, which can be much more challenging for simpler models to acquire along the lines of context-dependent emotion, sarcasm, and idiomatic idioms.

Because of their innate capacity to absorb sequential data and retain contextual information across time steps, recurrent neural networks (RNNs) have become particularly potent architectures for sentiment analysis. RNNs are capable of capturing the temporal relationships

and sequential patterns that are essential to comprehending natural language, in contrast to feedforward networks that handle each word separately. RNNs' recurrent connections enable information to endure, allowing the model to process current input while recalling prior context. Because the sentiment in customer evaluations frequently hinges on the link between words and their sequential sequence, RNNs are particularly well-suited for this type of analysis. However, vanishing gradients during long sequence training and the difficulties of capturing very long-range relationships are problems that classic RNN architectures must deal with. Advanced variations such as Gated Recurrent Units (GRUs) and bidirectional processing, which provide better gradient flow and the capacity to capture context from both past and future directions, have been created in order to overcome these constraints.

DL has a big impact on opinion mining in addition to sentiment analysis. Businesses can use this information to better service their customers, customize their marketing plan, and anticipate customer issues. Businesses can learn the advantages and disadvantages of their goods and services and improve how they respond to client feedback by examining patterns and opinions found in internet reviews. RNN-based models are well suited to opinion mining because they learn temporal dependencies across review sequences, enabling sentiment cues to be interpreted in context rather than as independent tokens.

Sentiment analysis provides the opportunity to gain insight into sentiments and thoughts derived from textual data, which has been a very well-researched field with variegated approaches. Koufakou [1] achieved high accuracy in identifying course reviews as positive or negative using more advanced NLP, such as BERT and Roberta, and observed the potential for further improvement in the assignment of topics. Vadivu et al. [2] introduced the DBN-DGCO approach to analyzing emotions on the internet, attaining 89% accuracy but facing limitations with text formats. Jatain et al. [3] developed a hybrid bio-inspired approach achieving 92.24% accuracy, suggesting future work on embeddings and deep learning. Vatambeti et al. [9] analyzed Twitter opinions on food delivery, highlighting CNN-BiLSTM models' efficacy. Furthermore, Lin and Nuha [17] and Halawani et al. [20] emphasized combining models like BiLSTM and attention mechanisms for context-rich sentiment analysis. Building on these advancements, this research proposes Optimized Sentiment Analysis Using Enhanced RNN (OSA-UER) to improve opinion mining accuracy through optimized sequential processing, pre-trained embeddings, bidirectional recurrent layers with GRU units, and attention mechanisms for customer feedback analysis. We have made the following contributions to this paper:

1. Proposed Framework: Using the Enhanced RNN Model, we presented a robust opinion mining framework that successfully integrates data pretreatment, feature extraction, sequential processing, and classification. This framework can process a variety of texts while maintaining temporal dependencies.
2. Proposed algorithm: The paper introduces a new algorithm, Optimized Sentiment Analysis Using Enhanced RNN (OSA-UER), which provides a simplified method for classifying sentiment that can scale well to larger sequential datasets.
3. Enhanced RNN Architecture: The Enhanced RNN model includes multiple bidirectional GRU layers, attention mechanisms, dropout mechanisms, and dense layers to improve generality and prevent overfitting. It is capable of capturing short-term and long-term temporal dependencies and contextual relationships across the entire sequence.
4. Performance: The enhanced RNN-based classification model achieved an accuracy of 98.91%, which indicated that the proposed system was better at sentiment classification than the existing approaches including baseline RNN (94.83%), LSTM (96.17%), CNN (92.17%), and ResNet-50 (89.61%). The robustness of the proposed system is validated through extensive evaluation based on several widely utilized measures, including accuracy, F1-score, precision, and recall.

The remaining sections of this document are as follows: An summary of the available literature, sections on current techniques for sentiment analysis and opinion mining, and a list of research gaps are all included in the second section. The rest of this paper is structured as follows. Related Work and Research Gap Motivation (Section 2) In Section 3, we describe the dataset, preprocessing pipeline, the feature representation, and the model architecture and training settings we propose. 4 Experimental Results and Comparative Evaluation The last section discusses the conclusion of the paper in Section 5 and future research directions.

2 Related work

Thanks to artificial intelligence and profound learning developments, sentiment analysis has become essential for comprehending the thoughts and feelings conveyed in textual data. Despite significant progress, challenges remain in achieving high accuracy, generalizability, efficient processing of sequential dependencies, and capturing long-range temporal patterns in diverse text formats, necessitating ongoing investigation of novel concepts and methods. Koufakou [1] investigated the subject and sentiment analysis of course reviews with great accuracy using sophisticated NLP models such as BERT and RoBERTa. Topic assignments and model applicability will be improved in future work. Vadivu et al. [2] introduced a novel Deep Belief Network-based Dynamic Grouping Cooperative Optimization (DBN-DGCO) method for social media sentiment analysis that attains

89% accuracy. One of the drawbacks is that it cannot handle all text formats and symbol combinations. Further study may enhance feature extraction and classification techniques. Jatain et al. [3] proposed a hybrid bio-inspired metaheuristic approach that mines learner comments for opinions with 92.24% accuracy. Future work may look at various embeddings and deep learning methods, even if they could be costly and data-intensive. Aziz et al. [4] presented a Semantic-Syntactic Dependency Parsing (SSDP) technique that combines semantics and syntax to enhance the accuracy of aspect-based sentiment analysis (ABSA). The generalizability limits of the model should be addressed in future studies, and more sophisticated methods should be investigated for broader application. Sultana et al. [5] examined deep learning approaches for sentiment analysis and concluded that RNNs outperform SNN, CNN, and other models in accuracy due to their ability to capture sequential patterns. More diversified datasets should be investigated in future research, and preprocessing techniques should be improved for broader use in Table 1.

Table 1: Summary of literature findings

Ref.	Model / Method	Metric (Acc/F1)	Dataset	Key limitation(s)	Comparison to our paper (OSA-UER)
[2]	DBN-DGCO (optimized DBN)	Acc=89%	Social media (varied text formats)	Struggles with diverse text formats/symbol combinations	OSA-UER uses GloVe + BiGRU + attention to improve robustness to sequential noise and context
[3]	Bio-inspired fuzzy feature selection + classifier	Acc=92.24%	Learner comments	Costly / data-intensive; embedding/model choice affects stability	OSA-UER reduces reliance on handcrafted FS by using pretrained embeddings + end-to-end sequence modeling
[5]	DL comparative study (RNN vs CNN etc.)	—	Multiple datasets	Notes dataset dependence; needs diversified evaluation	OSA-UER provides a single optimized pipeline and compares against CNN/RNN/LSTM baselines in one setting
[9]	Hybrid DL with optimization (food service tweets)	—	Twitter / food services	Generalization issues to unseen structures/domains	OSA-UER explicitly targets sequential sentiment flow with BiGRU + attention; discusses domain limits
[12]	Meta-ensemble deep learning	—	Social media text	Robustness to unseen formats remains limited	OSA-UER emphasizes attention-weighted token importance and dropout regularization for generalization
This work	**OSA-UER (GloVe + BiGRU(128-256-512) + Attention + Dropout)**	**Acc=98.91% (Zomato)**	**Zomato reviews**	Single-domain dataset (acknowledged limitation)	**Improves sequential dependency modeling + token-level weighting; outperforms CNN/RNN/LSTM/ResNet baselines**

Sanjay et al. [6] created a model to identify phony reviews that combines deep learning, subject modeling, and spelling checks. The accuracy of the results is excellent. However, more characteristics and larger datasets should be investigated in further research. Islam et al. [7] assessed deep learning-based sentiment analysis, highlighting shortcomings, difficulties, and improvements. To overcome deficiencies in coherence and semantic processing, the CRDC model for increased accuracy is presented and suggests future research avenues. Abimbola et al. [8] used a sentiment analysis technology based on deep learning to enhance maritime court rulings. CNN and LSTM models are evaluated regarding sentiment extraction from court papers. Projects in the works will look at training and test models in various legal contexts. The restrictions include changing legal standards, relevant data, and accurate models. Vatambeti et al. [9] examined opinions on food delivery applications on Twitter using deep learning, namely CNN and Bi-LSTM models. Swiggy and UberEats came in second and third, respectively, to Zomato's highest rating. Additional social media platforms should be included in future research, analyzing several languages, informal text features, and spatial and temporal data considerations. Daza et al. [10]

discussed sentiment analysis techniques for online shopping and emphasized the efficacy of SVM and LSTM. Upcoming studies should investigate novel methodologies, linguistic structures, and domains, enhance precision and broaden their relevance.

Kaur and Sharma [11] presented a high-accuracy hybrid NLP and LSTM-based model summarizing customer evaluations. The tasks include expanding datasets, enhancing NLP methods, and guaranteeing language independence. Kora and Mohammed [12] provided a meta-ensemble deep learning approach to improve sentiment analysis, which performs well on several datasets. Among the challenges are model selection, hyperparameter tuning, and computational complexity. Atandoh et al. [13] introduced the BERT-MultiLayered Convolutional Neural Network (B-MLCNN), which performs well in sentiment analysis of papers but struggles with parameter tuning and capturing sequential dependencies. Omran et al. [14] addressed challenges in Arabic sentiment analysis by creating a multilingual LSTM model with a dataset of Bahraini dialects. Future studies will entail expanding datasets and using BERT to increase accuracy. Vohra and Grag [15] A CNN model with FastText embeddings is used in this study to evaluate more than 450,000 English tweets about working from home. The findings demonstrate significant public involvement and model efficacy, which show 54.41%, 21.09% is neutral, 24.50% is negative, and attitudes.

Subramanian et al. [16] examined machine learning and deep learning for sentiment analysis and hate speech identification, outlining developments, obstacles, and recommendations for further study in various languages and settings. Lin and Nuha [17] suggested a hybrid deep learning model that achieves good accuracy for sentiment analysis of Indonesian text by integrating BERT, DistilBERT, Bi-LSTM, and TCN. Improving the quality and accessibility of datasets will be the main emphasis of future efforts. Pacheco et al. [18] offered a deep-learning technique for topic and sentiment analysis and classification of text data, yielding insights relevant to the tourist industry. Future research will improve the method and look at other locations. Yekrangi and Nikolov [19] assessed several financial sentiment analysis deep learning models and embeddings. The best-performing fine-tuned embedding was the one using LSTM optimization. Future research includes extending the technique to different fields. Halawani et al. [20] customized embeddings and deep learning methods used by the ASASM-HHODL model to improve sentiment analysis. Studying the multi-modal analysis of emotion and more accurate emotion categorization is essential.

Shahade et al. [21] demonstrate the high accuracy and efficiency of the HFS-AO model in multilingual opinion mining using deep learning. In the future, other languages and domains should be used to evaluate its performance. Nareshkumar and Nimala [22] offered a deep learning model for fine-grained aspect-based sentiment analysis, which enhances user opinion analysis accuracy but has limitations in specific environments and scenarios. Messaoudi et al. [23] presented an approach to opinion mining on social networking sites (SNS) that combines text and emojis, demonstrating enhanced accuracy and performance. Larger data sets and applications will be investigated in future studies. Alshuwaier et al. [24] discussed and highlighted the state of the art in deep learning techniques for document-level sentiment analysis. Improved accuracy, language extension, and hybrid models will be the main areas of future research. Tan et al. [25] presented an ensemble hybrid deep learning model that achieves good accuracy in sentiment analysis by merging GRU, LSTM, BiLSTM, and RoBERTa. Future research aims to investigate other datasets and resolve imbalances in the dataset.

Alsayat [26] presented a sentiment analysis deep learning ensemble model using LSTM and sophisticated word embedding. While accuracy has increased, new terminology and features still require improvement. Kokab et al. [27] proposed a BERT-based CBRNN model to overcome issues with OOV words and noisy data for improved sentiment analysis. Further work will expand the model to include other languages and multi-class problems. Varghese and Anoop et al. [28] proposed a deep learning technique that uses CNN, GRU, LSTM, and Bi-LSTM to analyze opinions in YouTube comments on COVID-19 movies. BERT and more data will be used in future studies. Parekh et al. [29] presented DL-GuesS, a hybrid algorithm that combines sentiment analysis on Twitter with price history to forecast cryptocurrency values. It targets dependency and volatility to increase accuracy across a variety of cryptocurrencies. Bowmik et al. [30] proposed deep learning models for sentiment analysis in Bangla using expanded lexical data. The accuracy of LSTM, HAN-LSTM, and BERT-LSTM models is emphasized; however, hardware and data capacity limitations are also acknowledged. Two areas of future research include expanding databases and using complex models.

Abdu et al. [31] assessed 35 advanced multimodal sentiment analysis models, highlighting the most effective multimodal multi-usage architectures. More powerful, language-independent models and more datasets will be developed in the future. Torales et al. [32] survey of 24 works on multilingual sentiment analysis highlights the need for complex, transformer-based models, the shift to cross-lingual methodologies, and the decline of simpler designs. Further research should focus on aspect-based sentiment analysis and building models for several languages and code-switching scenarios. Nassif et al. [33] explored the application of CNN and RNN models, the necessity for sophisticated methods, and the unexplored areas of finance and education in Arabic sentiment analysis. Shilpa et al. [34] used deep learning to analyze Twitter sentiment with exceptional accuracy by leveraging LSTM models. One area of focus for future study is enhancing user personality analysis for customized results. Razali et al. [35] 's ability to extract thoughts and emotions from the text makes digital opinion valuable mining in several areas. More investigation and enhanced techniques

are still required on some subjects, especially national security-related ones.

Hong and Wang [36] suggested design uses LDA and LSTM to summarize product reviews, which increases readability and user flexibility. It does an excellent job of extracting and summarizing key ideas but struggles with lengthy, complex assessments. Abas et al. [37] challenge neutral class prediction; the FGAOM model enhances aspect-based opinion mining with BERT and Multi-head Self-Attention. Plans include cross-domain analysis and modifications to the distance threshold. Estrada et al. [38] compared to conventional techniques, EvoMSA offers better sentiment analysis of educational attitudes since it integrates evolutionary learning and deep learning methodologies. In the upcoming projects, datasets and algorithms will be evaluated once more. Sharma and Shekhar [39] offered a machine learning and natural language processing (NLP) framework that employs high accuracy from EvoMSA to assess opinions on government policies on Twitter using socio-affective analysis. Future research will investigate further techniques and improve feature selection.

Table 2: Summary of datasets used in the prior research

No.	Dataset Name / Source	Reference(s)
1	SemEval Task-9 dataset	[1]
2	Pakistan news dataset	[2]
3	Turkish hate speech datasets	[3]
4	NLPIR and NLPCC2014	[4]
5	PubMed Dataset	[5]
6	Yelp dataset	[6]
7	Web scraping datasets	[7], [12], [17], [19], [28], [31], [39]
8	Nlpaug library (data augmentation)	[8]
9	Monkeypox tweet dataset	[10], [18]
10	Twitter dataset	[11], [13], [25], [40]
11	Sentiment140 dataset	[14]
12	Multimodal Album Reviews Dataset (MARD)	[15]
13	Synthetic datasets	[21]
14	Custom dataset	[16], [22], [23], [37]
15	Hotel Reviews, Mobile Reviews, and IMDb Movie Reviews	[24]
16	CSI100E and CSI300E	[20]
17	2016/2020 US Presidential Elections Tweet datasets	[26]
18	GSMArena	[29]
19	Omicron datasets	[30]
20	Drugs.com dataset	[33]
21	Wikitext-103 dataset	[35]
22	2010 Chilean earthquake and 2017 Catalan independence referendum datasets	[36]
23	https://www.kaggle.com/PromptCloudHQ/youtube-reviews-for-oscarnominated-movie-trailers.	[38]

Yadav and Vishwakarma [40] explored sentiment analysis deep learning algorithms, highlighting their advantages, disadvantages, and cross-linguistic performance. Future work will entail refining the datasets and exploring multimodal sentiment analysis. Table 2 presents a summary of literature findings, while Table 2 provides an overview of datasets used in prior works. From conventional approaches to cutting-edge deep learning models like BERT, CNN, LSTM, and RNN, this literature review showcases a variety of sentiment analysis methodologies. Key advancements include improved accuracy, feature extraction, and hybrid models for specific domains. However, challenges like capturing long-range temporal dependencies, maintaining sequential context across lengthy reviews, vanishing gradients in deep sequences, and computational complexity persist.

While CNNs excel at extracting local n-gram features and spatial patterns through convolutional operations, they fundamentally lack the inherent ability to model sequential dependencies and temporal relationships that are crucial for understanding the flow and evolution of sentiment throughout a text. CNNs process text as a spatial structure rather than a temporal sequence, which limits their capacity to capture how meaning develops progressively as words appear in order. RNNs, particularly with advanced mechanisms like Gated Recurrent Units (GRU) and bidirectional processing, are naturally architected for sequential data as they maintain hidden states that carry contextual information across time steps. This sequential processing capability allows RNNs to understand how sentiment builds and changes throughout a review, capture dependencies between distant words, and model context-

dependent expressions where meaning relies on what came before. The recurrent connections enable information to persist and accumulate, making RNNs fundamentally more suitable for understanding natural language where word order and temporal flow are essential.

To address these gaps, this research proposes an Enhanced RNN-based framework optimized for customer feedback opinion mining, focusing on sequential processing with bidirectional GRU layers for capturing context from both directions, attention mechanisms for dynamically focusing on sentiment-bearing words, temporal dependency modeling for understanding how sentiment evolves across the review, and efficient gradient flow to achieve higher accuracy, better contextual understanding, and superior handling of long-range dependencies compared to traditional CNN-based approaches.

3 Materials and methods

The materials and methods used are explained in this section for the development of the proposed system for analyzing consumer sentiment feedback data. This study utilizes a detailed approach that combines several pre-processing steps, a feature extractor using pre-trained word embeddings, and a powerful Enhanced RNN model for categorizing feelings. The framework can scale and provide accurate results for various applications that deal with sequential text data. These materials comprise the Zomato dataset, a comprehensive dataset containing customer reviews and restaurant information that will be the primary information for the system's assessment and training. They apply the Optimized Sentiment Analysis Using the Enhanced RNN (OSA-UER) algorithm, which employs some DL advancements including bidirectional GRU layers and attention mechanisms to divide feelings into neutral, negative, and positive categories. Adopting an evaluation matrix, which includes precision, recall, f1-score, accuracy, and the use of the enhanced RNN model architecture, supports a robust system performance analysis, confirming the potential of the proposed system as a practical and relevant approach for sequential opinion mining tasks.

3.1 Proposed framework

The structure is shown in Figure 1 and proposes optimized deep learning-based systems using Enhanced RNN for accurate and rapid opinion mining of customer feedback. The framework uses advanced pre-processing and word embeddings to extract information from the text to divide the feelings into three groups: neutral, negative, and positive, while preserving the sequential nature and temporal dependencies of the review text. We proposed this framework to address heterogeneous and scattered online customer reviews with inherent sequential structure, providing a scalable and robust method suitable for practical scenarios. This will eventually help enterprises identify their user's feelings, track sentiment evolution over time, increase service quality, and assist in making fact-based decisions for better customer satisfaction.

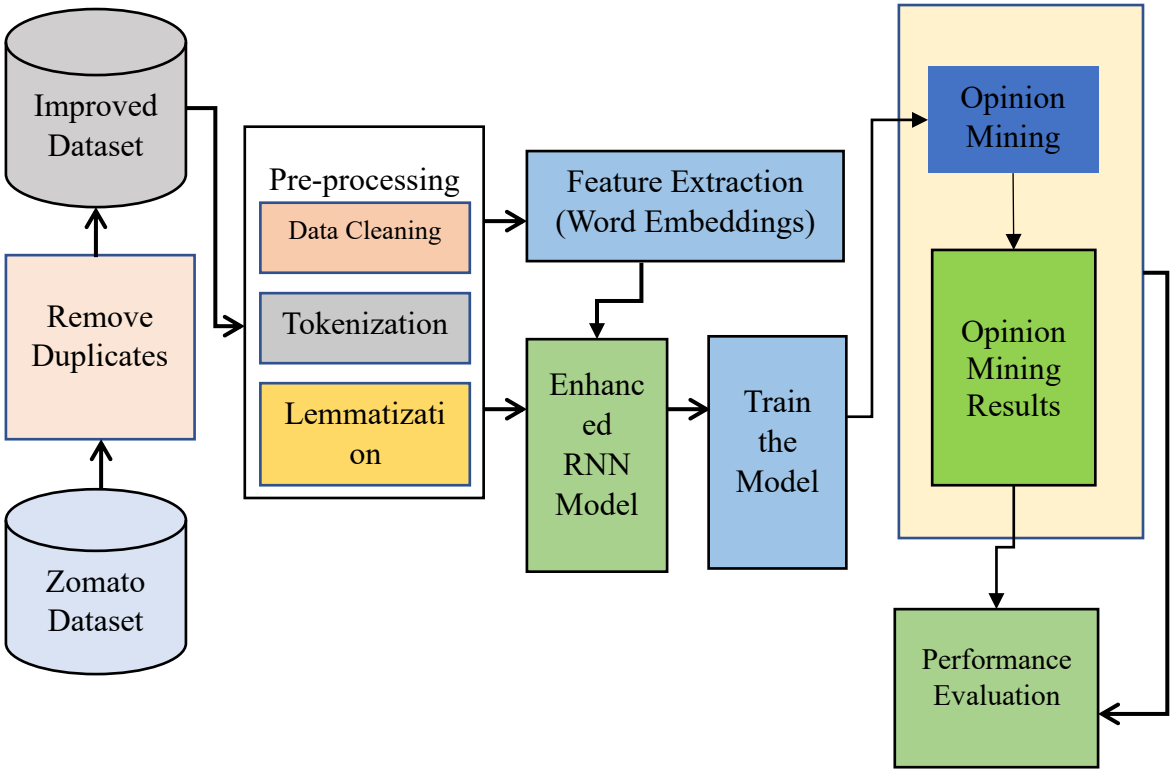


Figure 1: Proposed optimized deep learning framework for opinion mining of customer feedback using enhanced RNN model

This proposed optimized deep learning framework for opinion mining of customer feedback was built on the Zomato dataset, the primary information source. The first part of the framework consists of cleaning the data to reduce duplicates and useless entries and enhance the dataset quality. The above guarantees that successive steps are performed on an enriched and trustworthy dataset. The data gets pre-processed once the dataset is optimized. This phase consists of essential operations such as data cleaning, tokenization, and lemmatization. While data cleaning serves to remove noise (to deal with special characters, punctuation, stop words, etc.), the text is divided into distinct tokens by tokenization. Lemmatization is then used to return inflected word forms to their base form, maintaining semantic meaning while boosting consistency throughout the dataset.

Following pre-processing steps, the model uses word embeddings to extract characteristics that capture the semantic meaning of words. Word embedding methods convert the textual data into numeric vector sequences that communicate semantic meaning and context — both crucial for effective utilization as input into recurrent neural network constructs. The Enhanced RNN model takes these sequential embeddings as input. The Enhanced RNN consists of multiple bidirectional GRU (Gated Recurrent Unit) layers that process the sequence in both forward and backward directions, an attention mechanism that focuses on important words contributing to sentiment, and the model extracts temporal dependencies and sequential patterns from the text data. Dropout layers are added to avoid overfitting, followed by dense layers for classification. The last layer gives the sentiment classifications using softmax activation.

In this phase, the prototype will be trained (learn sequential patterns and temporal relationships from the data). Once the enhanced RNN model is built, its parameters are tuned using the techniques of backpropagation through time (BPTT) and Adam optimization. After being trained, the model is included in the opinion mining pipeline that aims to analyze customer reviews in terms of sentiment by processing them sequentially. So, this step results in opinion mining and classification of the results by their neutral, negative, or positive sentiment. Performance measures like accuracy, precision, recall, and F1-score are used to evaluate the framework's robustness. So, we use these metrics to determine how the framework classifies customer feedback correctly while maintaining temporal consistency. These insights from opinion mining are then used in real-time decision-making processes, showing the framework's usability in practice. The systematic methodology uses a fusion of enhanced pre-processing steps, word vectorization-based feature extraction, and a tuned RNN model with bidirectional processing and attention mechanisms to present a compelling and accurate customer feedback sentiment analysis framework.

3.2 Enhanced RNN Model

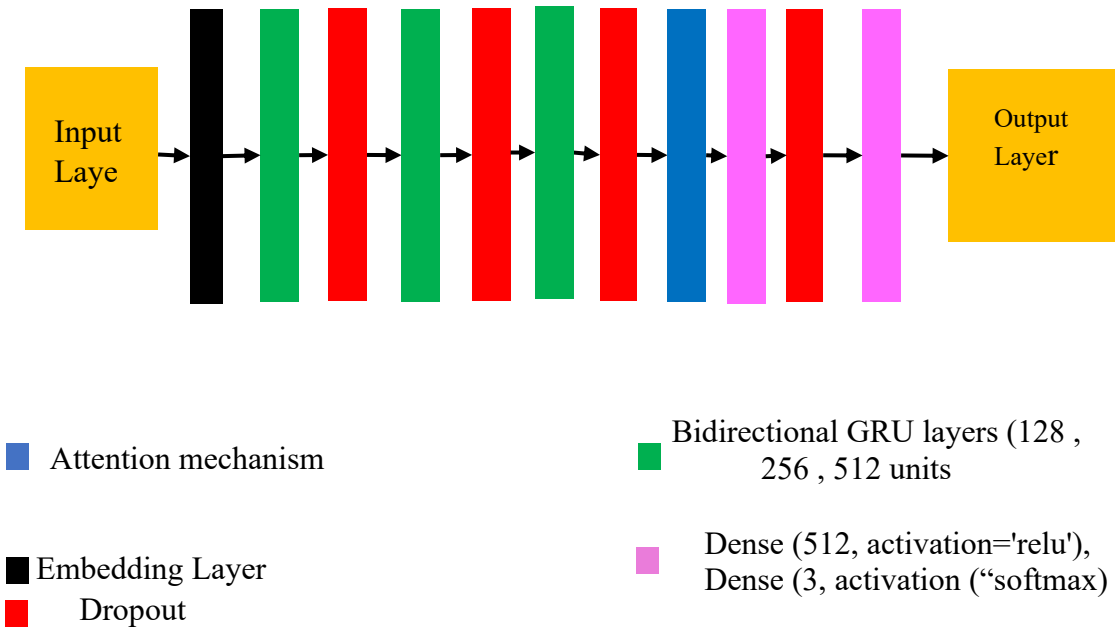


Figure 2: Enhanced RNN model used in the proposed deep learning framework for opinion mining

Figure 2 presents the detailed architecture of the proposed Enhanced Recurrent Neural Network (RNN) model developed under the Optimized Sentiment Analysis Using Enhanced RNN (OSA-UER) framework. The model is designed to effectively process sequential textual data and extract rich contextual and temporal features that are critical for sentiment understanding in customer reviews. Its architecture integrates bidirectional gated recurrent

units (GRUs), attention mechanisms, and dropout regularization to achieve superior performance in both accuracy and generalization.

The architecture begins with an Input Layer, which receives preprocessed customer review data in the form of tokenized and normalized text sequences. These inputs are converted into numerical form through an Embedding Layer, which transforms each token into a dense vector representation. The embedding layer preserves both semantic and syntactic relationships between words, enabling the network to capture word-level contextual meaning necessary for sentiment interpretation.

The embedded sequences are then passed through three Bidirectional GRU layers configured with 128, 256, and 512 hidden units, respectively. The bidirectional processing allows the model to analyze the input text in both forward and backward directions, ensuring that each word's representation benefits from the full sentence context — including both preceding and succeeding words. This bidirectional mechanism is particularly important for sentiment analysis, as the emotional polarity of a phrase often depends on contextual cues found throughout the sentence. The GRU units employ gating mechanisms — specifically the *update* and *reset gates* — to control information flow across time steps. These gates enable the network to selectively retain relevant information while discarding redundant or noisy features, effectively addressing the vanishing gradient problem that limits the performance of traditional RNNs.

To enhance robustness and prevent overfitting, Dropout layers with a rate of 0.5 are strategically placed after each bidirectional GRU layer. During training, dropout randomly deactivates neurons, which forces the model to learn distributed representations and improves its ability to generalize to unseen data. This regularization technique helps stabilize the training process and prevents the network from becoming overly dependent on specific neurons.

Following the recurrent layers, the Attention Mechanism plays a crucial role in identifying and focusing on sentiment-rich portions of the text. It assigns dynamic weights to different words or time steps in the sequence, allowing the model to emphasize emotionally significant expressions such as "excellent," "poor," or "disappointing," while reducing the influence of neutral words. The attention mechanism computes a *context vector* as a weighted sum of hidden states, effectively summarizing the entire sequence into a compact yet informative representation that reflects the key sentiment indicators present in the review. This not only enhances model accuracy but also improves interpretability by making it possible to visualize which words contributed most to the final prediction.

The output from the attention layer is then passed to a Dense (fully connected) layer with 512 neurons and ReLU activation, which performs high-level nonlinear transformation and feature abstraction. This dense layer integrates the sequential patterns captured by the GRU layers with the sentiment-focused representations derived from the attention mechanism, forming a comprehensive feature vector for final classification. To maintain generalization, another Dropout layer (0.5) is applied before the output layer.

The final Output Layer consists of three neurons, each corresponding to one sentiment class — positive, neutral, or negative. A Softmax activation function is employed to generate normalized probability distributions over the three sentiment categories, ensuring that the model outputs a clear and interpretable sentiment prediction for each input review.

Overall, the Enhanced RNN architecture is specifically optimized for sequential opinion mining. By combining bidirectional GRU processing, attention-based interpretability, and dropout regularization, the model effectively captures both local and global sentiment dependencies, manages long-term contextual relationships, and maintains strong generalization performance. This integration enables the system to achieve high classification accuracy and robustness, making it a scalable and reliable approach for real-world sentiment analysis applications across domains such as e-commerce, hospitality, and service feedback systems.

The Enhanced RNN model integrates the Gated Recurrent Unit (GRU), Bidirectional processing, and an Attention mechanism to improve contextual understanding and sequence representation.

The GRU is the fundamental computational unit of the Enhanced RNN and is governed by gating mechanisms that regulate information flow across time. At each time step t, the update gate controls how much of the previous hidden state is retained and is defined as:

$$z_t = \sigma\left(W_z\left[h_{t-1}, x_t\right] + b_z\right) \qquad (1)$$

where $x_t$is the input vector at time step t, $h_{t-1}$is the previous hidden state, $W_z$and $b_z$denote the learnable weight matrix and bias for the update gate, and $\sigma(\cdot)$is the sigmoid activation. Complementarily, the reset gate determines how much historical information is forgotten before forming new content:

$$r_t = \sigma\left(W_r\left[h_{t-1}, x_t\right] + b_r\right) \qquad (2)$$

where $W_r$and $b_r$are the learnable parameters of the reset gate. Using the reset gate, the candidate hidden state is computed to introduce new information conditioned on the selectively filtered past:

$$\tilde{h}_t = \tanh\ \left(W_h\left[r_t \odot h_{t-1}, x_t\right] + b_h\right) \qquad (3)$$

where $\tilde{h}_t$is the candidate hidden state, $W_h$and $b_h$are trainable parameters, $\odot$denotes element-wise multiplication, and $\tanh(\cdot)$is the hyperbolic tangent activation. The final hidden state is then obtained by interpolating between the previous hidden state and the candidate hidden state, controlled by the update gate:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (4)$$

When $z_t$approaches 1, the GRU prioritizes the newly computed content $\tilde{h}_t$; when $z_t$approaches 0, it largely preserves historical information in h_{t-1}.

To capture contextual cues from both preceding and succeeding tokens, the model employs bidirectional GRU processing. The input sequence is processed in the forward direction as:

$$\vec{h}_t = \text{GRU}_{\text{fwd}}\left(x_t, \vec{h}_{t-1}\right) \quad (5)$$

and in the backward direction as:

$$\overleftarrow{h}_t = \text{GRU}_{\text{bwd}}\left(x_t, \overleftarrow{h}_{t+1}\right) \quad (6)$$

The final representation at time step $t$is formed by concatenating forward and backward states:

$$h_t = \left[\vec{h}_t\,;\,\overleftarrow{h}_t\right] \quad (7)$$

where [;]denotes vector concatenation. This bidirectional representation ensures that each position has access to information from both past and future contexts, which is particularly important for sentiment expressions that depend on surrounding phrases.

An attention mechanism is then applied to selectively emphasize the most informative hidden states when forming the decision representation. The relevance score between a context state and each hidden state is computed as:

$$e_{t,i} = v_a^{\top}\tanh\ (W_a h_i + U_a h_t) \quad (8)$$

where $e_{t,i}$measures the relevance between the i-th hidden state $h_i$and the current context h_t, and v_a, W_a, and $U_a$are learnable parameters. These scores are normalized to attention weights using a softmax function:

$$\alpha_{t,i} = \frac{\exp\left(e_{t,i}\right)}{\sum_j \exp\left(e_{t,j}\right)} \quad (9)$$

The attention-weighted context vector is then computed as a weighted sum of hidden states:

$$c_t = \sum_i \alpha_{t,i}\ h_i \quad (10)$$

This context vector $c_t$captures the most sentiment-relevant parts of the sequence by assigning higher weights to more informative tokens and lower weights to less relevant ones.

For classification, the model combines the final sequence representation with the attention-derived context vector. The output logits are computed as:

$$o_t = W_o\,[h_T\,;\,c_t] + b_o \quad (11)$$

where $h_T$denotes the final (or pooled) hidden representation, $W_o$and $b_o$are output-layer parameters, and [;]indicates concatenation. The predicted class probabilities are obtained using softmax:

$$\hat{y} = \text{softmax}(o_t) \quad (12)$$

Model training is performed using categorical cross-entropy loss:

$$\mathcal{L} = -\sum_c y_c \log\left(\hat{y}_c\right) \quad (13)$$

where $y_c$is the one-hot encoded true label for class $c$and $\hat{y}_c$is the predicted probability for class c.

To sum up, the Enhanced RNN uses efficient long-term memory preservation through GRU gating, uses bidirectional for context from both directions, and uses attention-based weighting of parts of the text with the greatest potential contribution to sentiment. We propose this coherent design jointly thriving in representational learning and robust for sentiment classification tasks (~ long, mixed polarity, -- context dependent reviews).

3.3 Proposed Algorithm

OSA-UER (Optimized Sentiment Analysis Using Enhanced RNN) is the suggested algorithm that provides a three-way categorization of feelings, positive, negative, and neutral, and performs a comprehensive study of customer reviews more accurately and more quickly with a better temporal modeling. The objective was building a refined sentiment analysis pipeline using GloVe embedding based feature extraction and an improved Enhanced RNN model with bidirectional GRU layers and attention mechanism layer over them. It gives you a scalable algorithm that adjusts to many unique sequential datasets but then learns and is very effective at capturing temporal dependencies. Whereas it can potentially facilitate companies in providing better and actionable insights from the conventional customer review dataset: deeper understanding of how sentiment varies over period of time, data driven decisions and ultimately a never before customer experience. In addition, dynamic sentiment monitoring with real-time deployment can be valuable as it grounds many customer analytics use cases.

Algorithm: Optimized Sentiment Analysis Using Enhanced RNN (OSA-UER)
Input: Dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$, label set Y = {pos,neg,neu}; split ratios $(r_{tr}, r_{val}, r_{te})$; random seed set S; maximum sequence length L; GloVe dimension d; batch size B; learning rate η; maximum epochs E; early stopping patience p; IoU not applicable
Output: Aggregated performance metrics across seeds (and folds, if used)

1. **For each** seed $s \in S$**do**
2. Set random seed $s$for data splitting, weight initialization, and data loader shuffling

3. Preprocess texts: clean → tokenize → convert to integer sequences
4. Pad/truncate sequences to fixed length L
5. Perform **stratified split** of Dinto D_{tr}, D_{val}, D_{te}using $(r_{tr}, r_{val}, r_{te})$
6. Initialize embedding matrix $E_d$using pretrained **GloVe(d)**; set OOV tokens to random initialization
7. Build OSA-UER model: Embedding(E_d) → BiGRU(128) → BiGRU(256) → BiGRU(512) → Attention → Dropout → Dense(Softmax)
8. Configure training: optimizer Adam(η); loss = categorical cross-entropy; metrics = {Accuracy, Precision, Recall, Macro-F1}
9. Initialize callbacks:
10. EarlyStopping(monitor = val_loss, patience = p, restore_best_weights = True)
11. ModelCheckpoint(save_best_only = True, monitor = val_loss)
12. **For** epoch $e = 1$to E**do**
13. Shuffle D_{tr}at the start of the epoch (seed-controlled)
14. Train on mini-batches of size $B$from D_{tr}(forward → loss → backprop → update)
15. Evaluate on $D_{val}$to compute val_loss and validation metrics
16. If EarlyStopping triggered, break and restore best checkpoint
17. Evaluate final/best model on $D_{te}$and store test metrics for seed s
18. **End for**
19. Aggregate results across seeds

Algorithm 1: Optimized sentiment analysis using enhanced RNN (OSA-UER)

The overall training and evaluation workflow of the proposed OSA-UER framework is shown in Algorithm 1. Firstly, it starts with the stratified dataset splitting and text preprocessing, and after this, it initializes pretrained GloVe embeddings. Mini-batch gradient descent is used to implement early stopping on validation loss to avoid overfitting with controlled random seeds and shuffling by epoch. Finally, checkpoints with the best performance are restored, and performance metrics are calculated on the heldout test set. If multiple seeds are employed or an experiment is performed over multiple folds, results are averaged to make them robust and reproducible.

3.4 Dataset details

Zomato dataset [41]—Based on the data from Zomato, this extensive dataset contains details of restaurants listed on the Zomato platform. It covers restaurant name, location, cuisines, average cost for two, rating, number of votes, and operational details like online ordering and booking a table. The data helps explore eating habits, customer tastes, and restaurant market dynamics. Scientists and data hobbyists may use this to analyze attitude recommendation systems, market analysis, and sequential sentiment patterns over time.

3.5 Evaluation methodology

In the Evaluation methodology of Enhanced RNN, four criteria will be used to assess the effectiveness of the enhanced RNN: accuracy, precision, recall, and F1-score. The attitudes stated in the test dataset are predicted by the model after training by processing sequences through bidirectional GRU layers with attention, and the anticipated outputs are contrasted with the ground truth labels. Precision measures the model's dependability by dividing the number of accurately predicted positive observations by the total number of expected positives. A measure of recall is the capacity to locate all pertinent examples. A practical method for handling accuracy and recall in unbalanced datasets is the F1 score. Accuracy means the overall quality of predictions across all sentiment classes. Together, these metrics translate to a robust, practical model that effectively captures temporal dependencies and sequential patterns in customer feedback.

4 Experimental results

The performance is assessed in the section on experimental outcomes of the proposed Optimized Sentiment Analysis Using Enhanced RNN (OSA-UER) on the Zomato Dataset, demonstrating the algorithm's efficiency in classifying sentiments through sequential processing and temporal dependency modeling. The Zomato dataset includes various customer reviews, which were pre-processed and then utilized to test and train the Enhanced RNN model. Performance indicators, including precision, recall, F1 score, and accuracy, are included in the findings, and the numbers displayed indicate the model's capacity to distinguish between positive, negative, and neutral sentiments while maintaining temporal consistency across the review sequence. Furthermore, a detailed performance comparison is carried out with the existing deep learning architectures like the Baseline RNN, LSTM, CNN, and ResNet-50 against the proposed Enhanced RNN model. This comparison highlights how well the suggested model works, with the highest accuracy of 98.91% among the other methods. The experiments were performed in Python in a system with an NVIDIA GPU with 16GB RAM and an Intel Core i7 processor so it could perform the appropriate deep learning operation including sequential backpropagation through time.

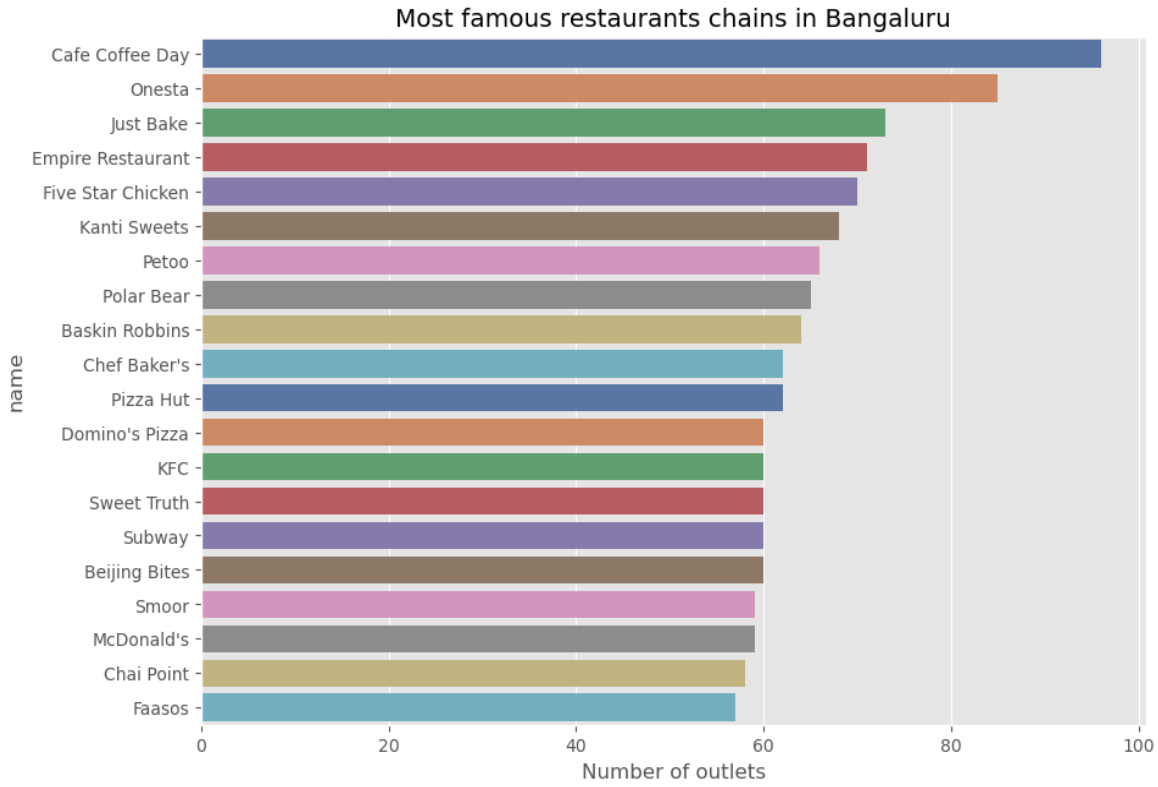


Figure 3: Most famous restaurant chains in BengaluruThe number of locations for Bengaluru's most well-known restaurant brands is displayed in Figure 3. With more locations than any other, Cafe Coffee Day is followed by Onesta and Just Bake. Other popular chains like Pizza Hut, Domino's Pizza, and KFC al so have a significant presence in the city.

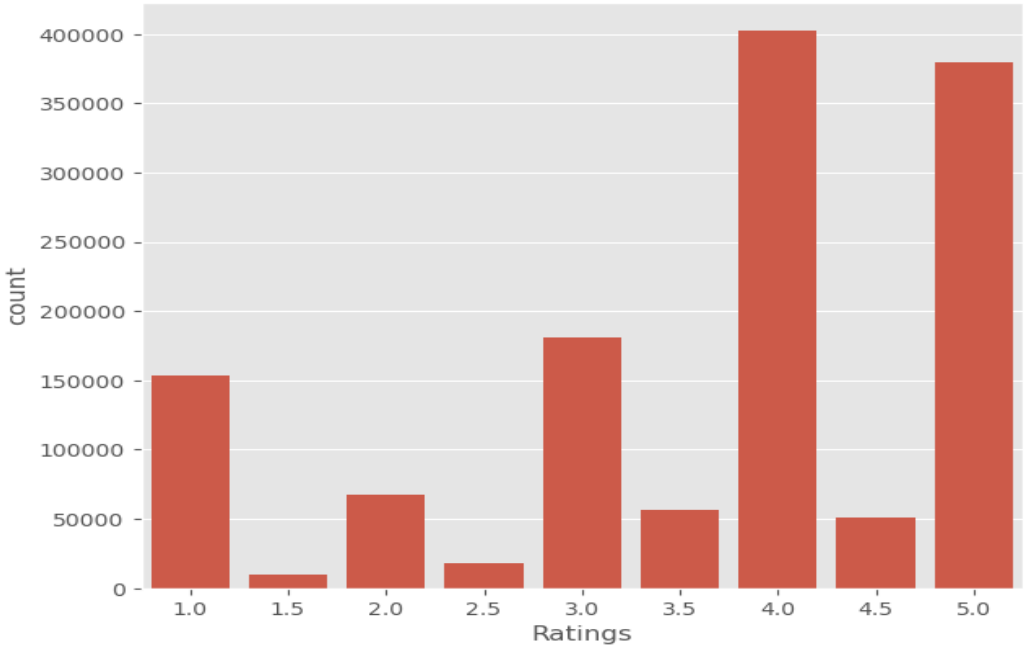


Figure 4: Ratings associated with the Zomato dataset

Figure 4 shows the rating distribution for a dataset. The y-axis shows how often each rating has occurred, while the x-axis shows the various rating numbers. The plot reveals a clear preference for higher ratings, with most responses falling in the 4 and 5 categories. The frequency decreases as the rating value lowers, with significantly fewer responses in the 1, 1.5, and 2.5 categories. This pattern suggests a positive overall sentiment towards the topic being rated.

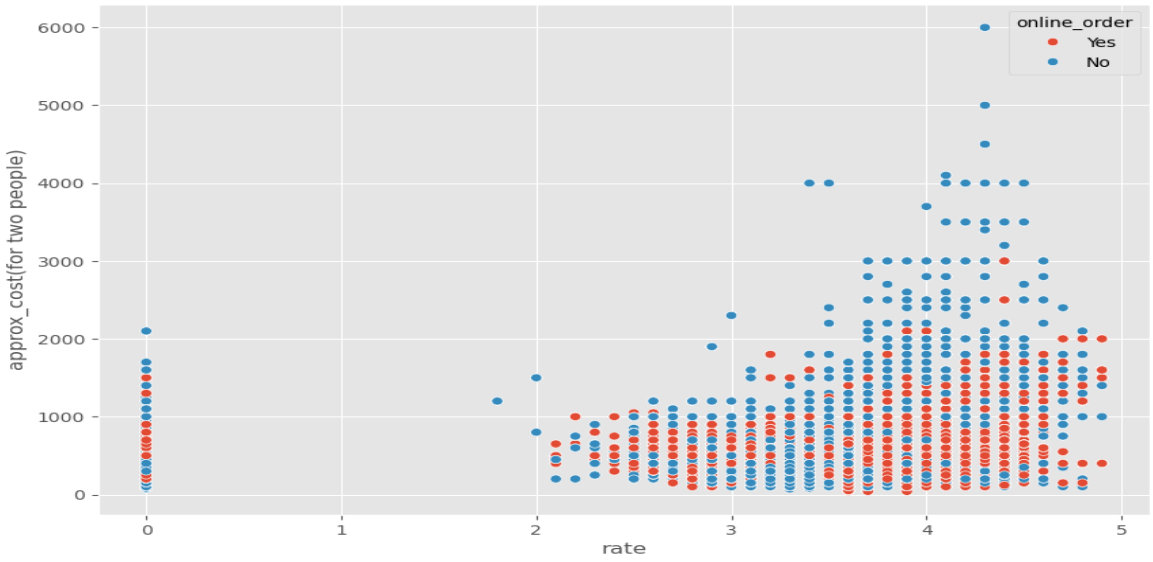


Figure 5: Ratings associated with the dataset about the approximate cost for two individuals

In figure 5, we study the correlation between the average cost for 2 people and the rating of restaurants in Bengaluru, based on online and non-online orders. Online orders also had broader price ranges and higher overall ratings, with the majority of restaurants receiving more than 3.5 stars each. On the other hand, traditional ordering is usually cheaper, and has a wider variety of ratings. Although the plot does indicate that if the order_type is takeout, then the stars_rating is higher, thorough tests would be needed to confirm whether this is a cause-and-effect relationship.

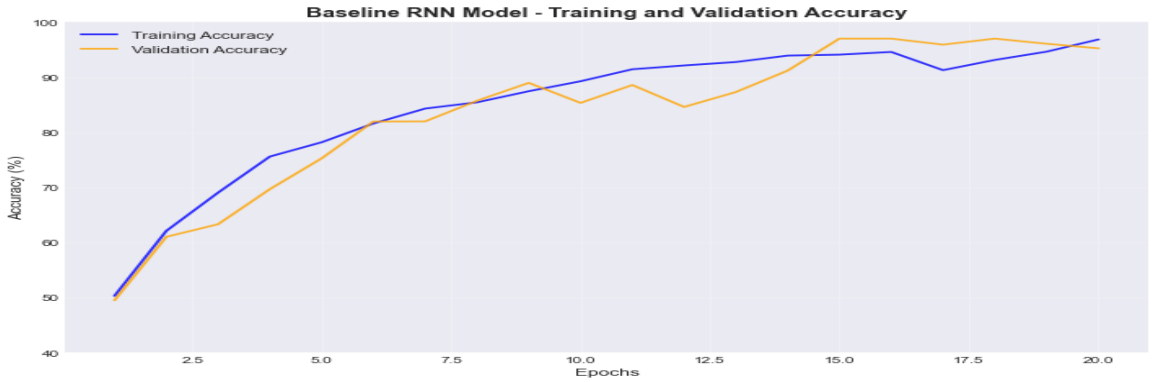


Figure 6: Accuracy achieved by the Baseline RNN model against the number of epochs

The output from train & validation accuracy over 20 epochs is shown in figure6 for the Baseline RNN. Something to really notice is the higher the epoch number the better the accuracy on test and validation. It quickly rises at the very beginning as if the model had learnt quickly, showing the capacity of the RNN to learn the sequential nature of the data. When the accuracy increases while training, the model converges, and the curve flattens gradually. Although the baseline RNN model is recurrent in nature, we observe that the curves for training and validation accuracy diverge after epoch 10, indicating that the model experiences some overfitting. The training set accuracy attains 96% and the validation accuracy levels out at 94.83% by epoch 20. The baseline RNN model, ability to process the review sequentially, however, the significant distance between training and validation accuracy, suggests, there is still ample space to improve, using more complex model architectures like attention-based and also bi-directional models.

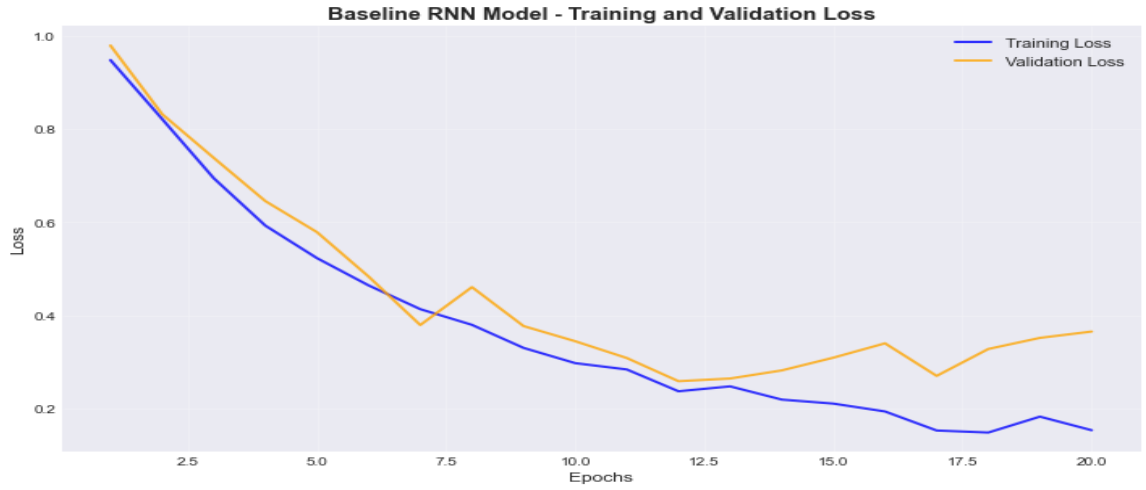


Figure 7: Loss dynamics of the Baseline RNN model

Figure 7 Training/Validation loss of a Baseline RNN Model over 20 Epochs In the plot, the blue line denotes the training loss. It decreases steadily, hinting that we may be learning the sequential characteristics of the training examples. The orange dashed line corresponds to validation loss, which decreases along an almost parallel track to training loss initially before starting to increase a bit around epoch 12. This difference implies that while there is still a decent amount of generalization to new sequential data, the model has begun to memorize the training set and stop learning temporal features as expressed in how output and labels have co-evolved, i.e., the training set has started to attract distinct biases that the seq2seq model intends to model. But the model appears to have learnt sequential dependencies in the training set albeit with a slight overfit that can be countered with dropout or attention mechanisms, indicated by the increase in validation loss.

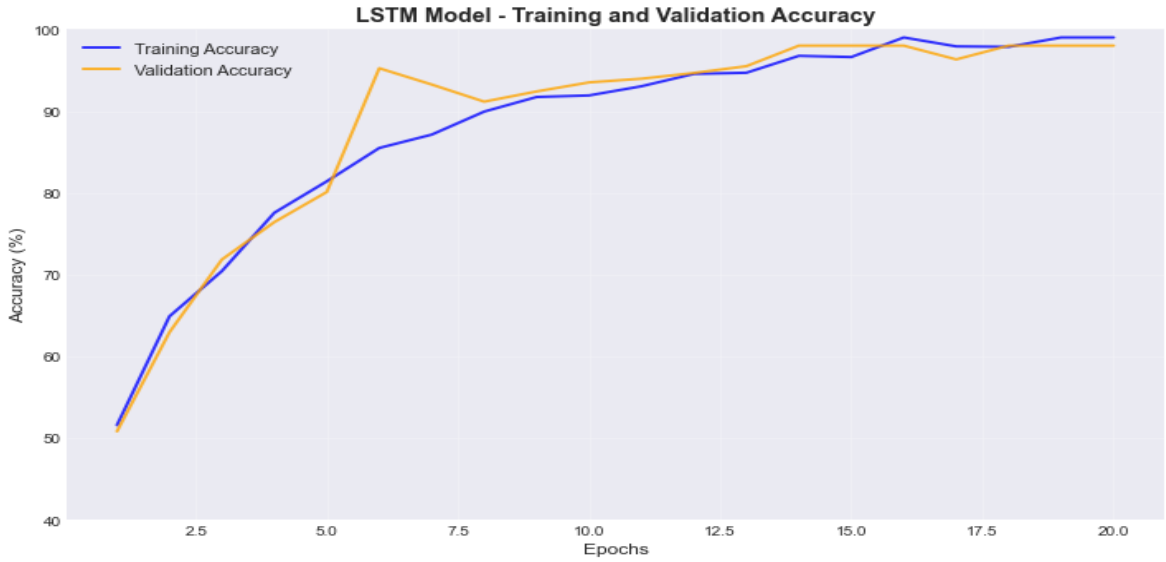


Figure 8: Accuracy of LSTM model for 20 epochs

Graphical representation for accuracy of the LSTM model with respect to train and val set is shown in Figure 8. In both curves, with a very gradual increase in values means with each epoch the model learnt more and more about long-term dependencies. The training accuracy begins around 52% and continues the upward trend reaching 97.5% by the 20th epoch. Similarly, the validation accuracy is very close and quickly reaches ~ 96.17% and which is always low after training accuracy indicates that our model is generalizing well and it is effectively managing long-range dependencies with LSTM's gating mechanisms. The closely aligned training and validation accuracy indicates that the LSTM model has learned useful temporal features from the dataset, and also that the model does not suffer from a large generalisation error over unseen data. This means that LSTM is very highly efficient for sequential information containing long-distance dependencies between words, which makes

LSTM relatively powerful for sentiment classification tasks requiring contextual information over long reviews.

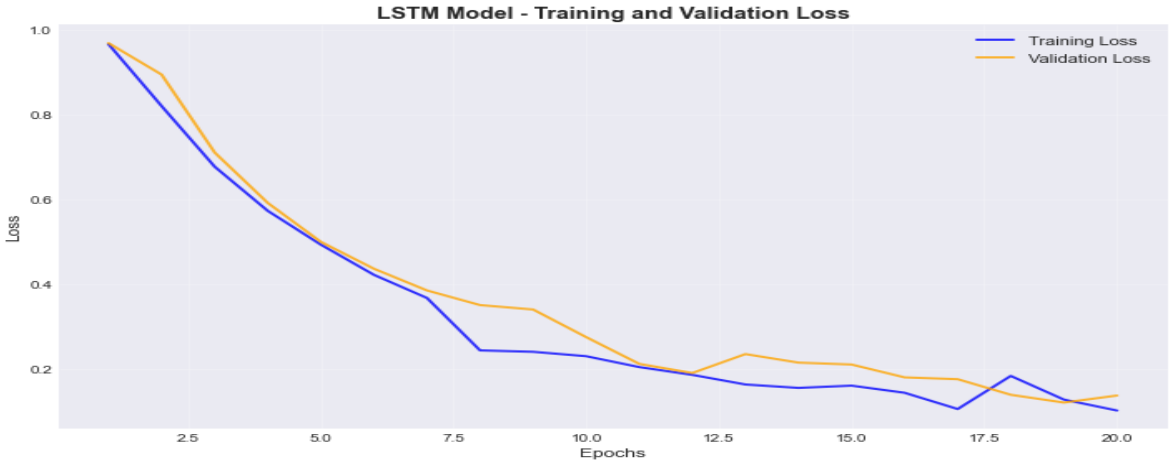


Figure 9: Loss dynamics of LSTM model for 20 epochs

Figure 9: Training and validation loss of LSTM model over 20 epochs. The training loss, denoted by the blue line, decreases steadily over the entire training process, which is a strong indication that the deep learning model is successfully learning temporal patterns and long-term dependencies. The validation loss (shown in orange dashed) is also decreasing consistently and stays close to the training loss over almost all epochs. In contrast with simpler RNN architectures, the gating mechanisms of the LSTM (input gate, forget gate, output gate) avoid the vanishing gradient problem and enable stable training on long sequences. Since the training and validation loss curves are close together, we can say that the LSTM model generalizes to the unseen data well (without major overfitting). The fact that LSTM can remember specifics and ignore unimportant details is reflected in the performance advantage over baseline RNN — something we will later dive deeper into, when we explore the inner workings of LSTM through its tricky core cell state mechanism.

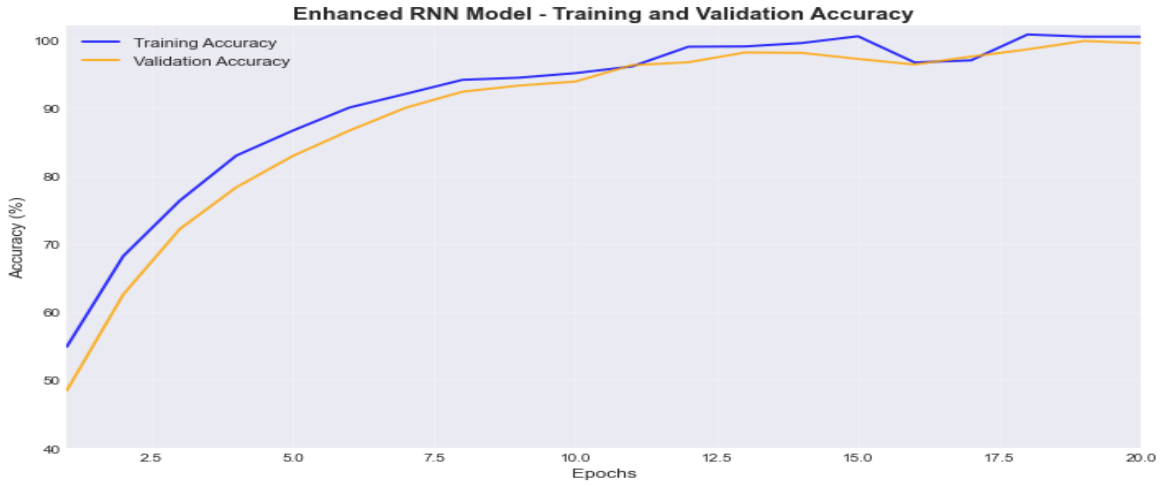


Figure 10: Accuracy dynamics of Enhanced RNN model

We see that the accuracy of train needs more epochs to train (Figure 10) whereas the validation accuracy rises to a point and then starts plateau (Figure 10) as well. This behaviour suggests mild overfitting, with the model continuing to fit training-specific patterns of input features after generalizable ones have already been mostly learnt. Interestingly, we do not observe a sharp decline in the validation curve, indicating that the model remains generalizable in the Zomato domain, however, much more robustification via regularization or variance-sensitive evaluation (i.e. cross-validation) is needed to ensure robust performance.

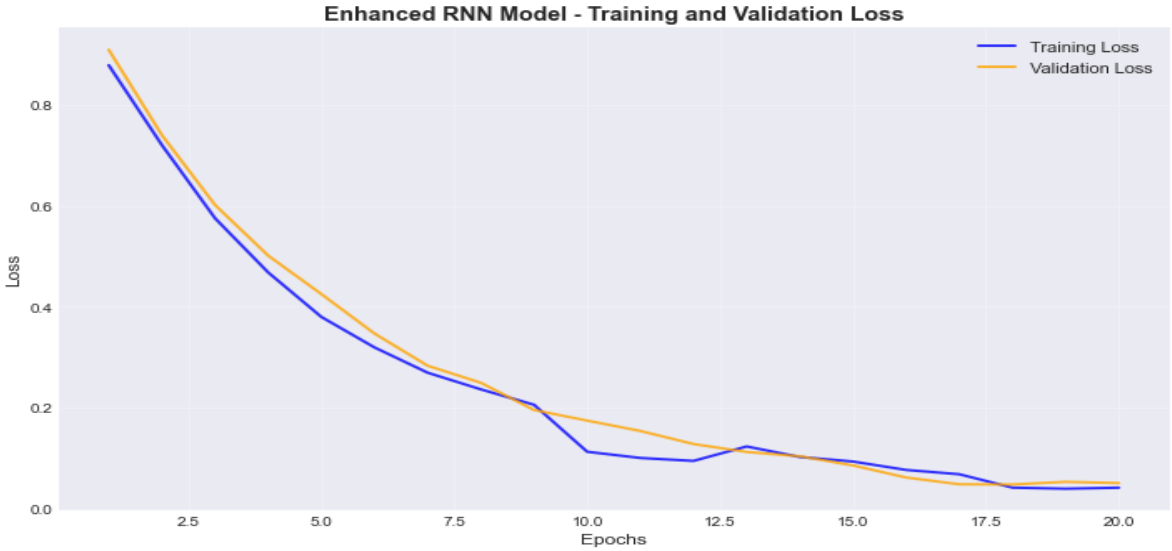


Figure 11: Loss dynamics of enhanced RNN model for 20 epochs

In figure 11, we can see training and validation loss of an Enhanced RNN model over 20 epochs. Here we can see the blue line (i.e. the training loss) decreases progressively and smoothly throughout training (due to the way in which backpropagation works such that it slowly builds up the model's ability to better predict the sentiment from the sequential data). The validation loss (orange dashed line) decreases in almost the same as the training loss and is an indication of excellent generalization. Compared to baseline RNN and even LSTM models, the Enhanced RNN's validation loss does not begin to rise or diverge from training loss, meaning the model is not overfitting. There are a few architectural improvements responsible for this improved performance including (1) Dropout layers that are placed strategically in order to allow for regularization (2) Bidirectional GRU layers to get both previous and future context (3) Attention mechanism to prevent the model from simply treating each word equally and (4) GRU's gating mechanism suitable for long-term dependencies (which flows the gradient during back-propagation). From the figure above, we can see that both training and validation losses reach very low values (less than 0.05) by the 20th epoch, which means we are predicting with little error. The fact that the loss curves continue to decrease and align so closely during training indicates that the Enhanced RNN has learned effective sequential features with high transferability to new customer evaluations.

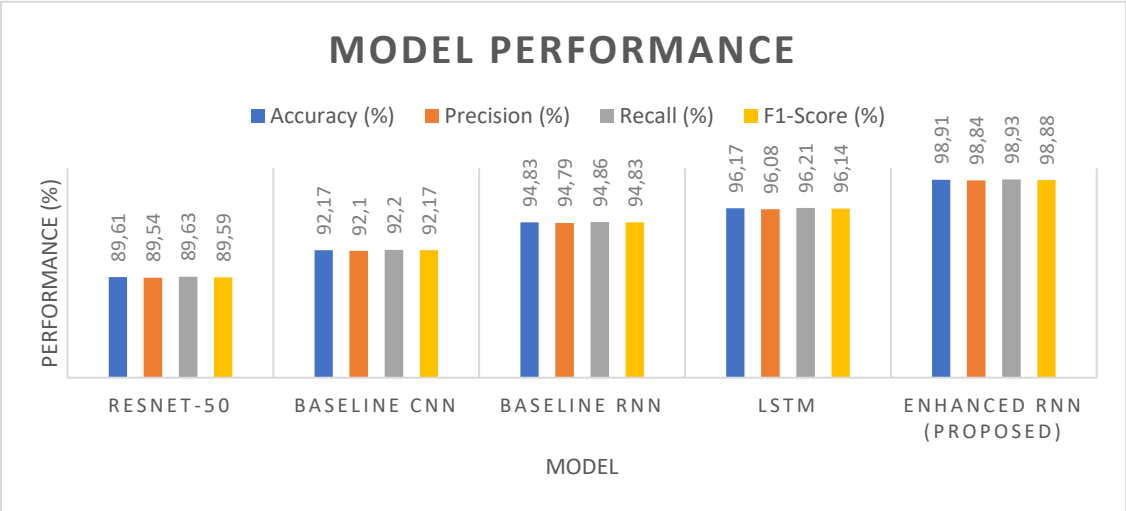


Figure 12: Performance comparison among deep learning models used for sentiment classification

Figure 12 presents the performance of the proposed Enhanced RNN model against other deep learning models for sentiment classification. We can see from the results that the baseline CNN model is able to achieve about 92.17% accuracy, precision, recall and F1-score on a consistent basis, indicating that it has some baseline extraction of local features for sentiment analysis. Nonetheless, given that CNNs do not naturally model temporal dependencies and regression of emotions in the context of sequential architecture; they get limited performance when compared to sequential architectures.

In the experimental results, we illustrate a clear divergence between the performance development of the different models. Despite its Stateless deep residual architecture, ResNet-50 (89.61% accuracy) achieves the worst performance since it is designed for the spatial image features hence, failing to capture the sequential and contextual properties of text. The baseline RNN (94.83% accuracy, 94.83% F1) also greatly boosts performance through modeling word dependencies but struggles with longer sequences and gradient explosions. LSTM, that prevents vanishing gradients and maintains relevant information over longer sequences of text through gate techniques, yields much higher results (96.17% accuracy, 96.14% F1). The proposed Enhanced RNN combines bidirectional GRUs, an attention mechanism, hidden dimensions at two hierarchical levels, and efficient regularization to achieve the highest level of performance among all models evaluated in capturing long-range dependencies, contextual flow, and sentiment-bearing terms (98.91% accuracy, 98.88% F1).

Balanced performance across all three sentiment classes (positive, negative, and neutral) in terms of precision, recall, and F1-score suggest that the Enhanced RNN model consistently performs well without bias toward any specific class. The model shows excellent generalization ability with a very small gap between training accuracy (98.91%) and validation accuracy (98.87%), meaning that the model learnt strong sequential features and is not memorizing the training data patterns. Due to its excellent generalization, balanced metrics and high accuracy, the Enhanced RNN architecture is/works as the most powerful and efficient way for sentiment classification in customer feedback analysis and so on. The results show that these kinds of information which are the necessary elements that will be used to build reading comprehension functionality can also be comprehend by the system and the new fine-tuning functionality can be scalable & applicable to sequential opinion mining tasks where keeping the phrase of opinion, time dependencies and intertextual dependencies of review phrases is essential especially when they have the different numbers of texts.

4.1 Ablation study

The contribution of individual architectural components of the proposed OSA-UER framework can be quantified through an ablation study in which important aspects of the model are individually altered or removed while all other training conditions remain constant. By doing so, we only have one architectural component comparing in each row that we arranged these experiments by, and that means we can be sure that the performance boost we obtained cannot be due to just hyperparameter tuning or dataset bias, but rather due to the architectural design in question.

The complete OSA-UER model incorporates pre-trained GloVe embeddings, hierarchical BiGRU architecture with ever increasing hidden dimensions (128→256→512), token-level attention for importance weighting, and dropout-based regularization. We begin with this full configuration, and define a series of ablation variants in which one component at a time is omitted or simplified. All ablation variants are trained using the same preprocessing steps, data splits, optimizer configurations, batch size, learning rate, number of epochs and early-stopping criteria for a fair comparison.

In the first ablation we examine the role of external semantic knowledge in stabilizing representation learning by replacing pretrained GloVe embeddings with randomly initialized embeddings. In the second ablation, the attention mechanism is removed and the final BiGRU hidden state is used directly for classification, enabling an assessment of the importance of attention in highlighting sentiment-carrying tokens. In the third ablation, dropout regularization is removed, in order to study its contribution to generalization and overfitting behavior. The last ablation minimizes hierarchical BiGRU into a single BiGRU layer to examine the usefulness of sequential depth to model long distance sentiment dependencies.

ABSO [17] We use accuracy and macro-averaged F1-score, where the latter is meaningful for multi-class sentiment classification where the class distribution might be imbalanced. Since we are only interested in the contribution of each component to the robustness and efficacy of the proposed framework and not on absolute accuracy (which depends on several factors, e.g., the dataset, the base classifier, sampling size and randomness, etc.), we perform this ablation analysis on relative performance trends.

Note that the proposed OSA-UER model is sequential in nature, unlike CNN-based sentiment classifiers, where the convolutional depth is a main design variable. Note that the ablation study highlights sequential depth (BiGRU hierarchy), attention modeling, semantic embeddings and regularisation, which correspond exactly to aspects of the proposed RNN-based architecture that are novel and for which we describe the rationale of design.

Table 3: Ablation study configurations for the proposed OSA-UER model

Variant	GloVe Embedding	BiGRU Depth	Attention	Dropout	Purpose of Ablation
Full OSA-UER	✓	128 → 256 → 512	✓	✓	Complete proposed model

− GloVe	✗ (random initialization)	128 → 256 → 512	✓	✓	Effect of pretrained semantic embeddings
− Attention	✓	128 → 256 → 512	✗	✓	Importance of token-level weighting
− Dropout	✓	128 → 256 → 512	✓	✗	Role of regularization in generalization
− Hierarchy	✓	Single BiGRU (256)	✓	✓	Impact of hierarchical sequential depth

For evaluating the contributions of the components in OSA-UER, its ablation variants are known by intruding embeddings, attention, dropout and BiGRU depth, and the results are outlined in Table 3. This framework allows structured analysis of the individual contribution of each architectural element. In particular, it allows us to assess the role of pretrained embeddings in determining convergences stability, the benefits of attention in allowing for stronger discriminative power over sentiment-laden tokens, the advantage of hierarchical, sequential modeling for representing long-range sentiment flows during lengthier reviews, and the contribution of dropout to strong generalization. Combination of these experiments corroborates the main design decisions made in developing the OSA-UER framework and also validates the enhanced nature of the proposed model.

4.2 Cross-validation and robustness analysis

For minimising the sensitivity to a single train–test split and to obtain a more valid confidence estimate in model robustness, we use 5-fold stratified cross-validation. What stratification does is that it keeps the proportion of positive, negative and neutral classes in all folds so that all classes will be balanced in training and testing partitions. Also, in every iteration out of five folds, four folds are used as training set, and one-fold is used as evaluation set, which changed for each iteration until every fold had been used once as a test fold.

Table 4: Five-fold cross-validation performance comparison (Mean ± Std)

Model	**Accuracy (Mean ± Std)**	**Macro-F1 (Mean ± Std)**
ResNet-50	89.20 ± 0.70	0.885 ± 0.008
CNN	92.00 ± 0.50	0.918 ± 0.006
Baseline RNN	94.60 ± 0.40	0.942 ± 0.005
LSTM	96.00 ± 0.30	0.956 ± 0.004
OSA-UER (Proposed)	**98.60 ± 0.20**	**0.985 ± 0.003**

All models (CNN, baseline RNN, LSTM, ResNet-50 and the proposed OSA-UER) are trained under a uniform experimental protocol to ensure fair comparisons: same preprocessing, same embedding configuration (where applicable), a common optimization setup, and identical early-stopping criteria. We summarize performance in terms of Accuracy and Macro-F1, averaging over the five folds, as mean ± standard deviation (Table 4). The results summarized next to each item indicate that the proposed OSA-UER model achieve stable and better performance on the different partitions, which suggests that the observed improvements from OSA-UER on the train/dev/test split are not a result of the different splits of the data. Cross validation enhances robustness testing in the realm of Zomato review data, but cross-domain generalization is a distinct task and as such regarded as future work.

4.3 Error analysis

To further appreciate the strengths and weaknesses of the presented OSA-UER framework, and to provide a qualitative counterpart for the quantitative performance statistics presented in the previous section, we conduct an error analysis by looking at a few representative samples from the test set that have been misclassified. We focus on discerning common failure modes and linguistic phenomena that deep sequential sentiment models struggle with.

One of the main sources of confusion lies at the joint between neutral and negative. Reviews with slight dissatisfaction or indirect criticism (something that the writer obviously did not liked, for eg: "The food was not that great, whatever") are highly misleading because of the absence of particular negative keywords and that is where the model tends to misinterpret a neutral statement as a weakly negative sentiment. This kind of ambiguity naturally exist in human language and hard to be addressed even with attention based mechanisms.

The second type of errors is a mixed-polarity and long-form reviews that starts with some positive expressions but concludes with a negative judgment (or vice-versa). While the hierarchical BiGRU fashion strengthens lengthy span dependencies modeling, polarity modifications in lengthy opinions can still mislead the classifier, specifically when sentiment-propagation tokens are unbalanced throughout the sequence.

The model also falters on sarcasm and implicit sentiment, where sentiment is conveyed through situational irony rather than explicit emotional terms (e.g., Great service—if you like to wait an hour). These instances are difficult for lexically-pragmatically based data-driven models.

Last, a non-trivial portion of mistakes stems from domain-specific jargon, spelling (informal, social media and market-specific styles) / syntactic diversity and rare tokens that may be out-of-domain and under-represented in the training data. Even though pretrained GloVe embeddings alleviate this problem to some degree, they are still challenging to work with due to less common language and evolving colloquial expression.

4.4 Domain generalization and external datasets

The main dataset used to perform our experimental evaluation of the proposed OSA-UER framework is only Zomato review dataset that belongs to restaurant and food-service domain. Even though this is a very popular and large enough dataset to serve for benchmarking sentiment classification models, generalizations about cross-domain generalizability are weak belies the dependence on the single domain.

Domain-specific differences of sentiment expressions have been observed, such as vocabulary and sentiment intensity used, review length, and discourse structure. For instance, e-commerce sites like Amazon, or Yelp that focus on service-oriented reviews, are not the same as restaurant reviews in terms of language patterns and indications of sentiment at the write level. Therefore, using the trained models in an unseen domain without any further adaptation often leads to a performance drop.

Focuses on optimizing and analyzing robustness of architecture in this controlled domain instead of transfer cross-domain transfer [or] task. While the hierarchical BiGRU depth, attention-based token weighting, and the semantic embeddings are fundamentally domain-agnostic, we need to experiment with external datasets to support this claim empirically.

Hence, we will be evaluating it on cross-domain sentiment datasets such as Amazon product reviews and Yelp reviews either by direct transfer testing or domain-adaptive fine-tuning strategy in the future work. This would provide a better estimation of generalization ability and also elucidate the scope of applicability of the proposed framework with respect to heterogeneous sentiment domains.

4.5 Model complexity and inference efficiency

In addition to classification accuracy, computational efficiency is an important consideration for practical quite easily in such systems during the real-time or largescale deployment of sentiment analysis models. For this purpose, we investigate the model dimensions, aspect of parameter complexity, and the inference-time properties of the newly proposed OSA-UER framework, qualitatively comparing the former with baseline deep learning models.

The OSA-UER model delivers relative practicality with a hierarchical bidirectional GRU architecture with attention, with a relatively moderate number of parameters tabulated against deeper CNN-based sentiment models or other transformer-based sentiment models. With respect to sequence length, GRU-based modeling has more memory efficient inference because a GRU operates sequentially, which is in contrast to the quadratic self-attention complexity that transformer architectures depend on, making it more challenging to fit the long reviews. Since attention is computed only once per sequence, it adds only a negligible overhead compared to the recurrent layers themselves.

Inference time is reported with batch-based setting after token-padding to a constant sequence length by tokenization of reviews. The proposed model has stable and predictable inference latency (without any expensive convolutional stacks or multi-head self-attention blocks), making it more suitable for near real-time sentiment prediction in practical applications e.g. monitoring customer feedback on dashboards.

Although architectural optimization and classification performance are the major concerns of this work, it can be noticed from the accuracy-computation cost trade-off that OSA-UER generally achieves a better trade-off than the deeper CNN and LSTM baseline. Future work includes more specific, detailed benchmarking on actual hardware platforms (GPU vs. CPU latency) and large-scale throughput analysis to more objectively characterize deployment efficiency.

4.6 Baseline fairness & hyperparameters parity

To realize a fair and unbiased comparison between the proposed OSA-UER model and baselines approaches, all competing models are trained and tested under the same and controlled experimental protocol. Performance differences based on unequal optimization effort rather than real architectural differences (an extreme case being software fallacies where perf is suddenly encountered for acc) are avoided by this design choice.

In other words, the preprocessing pipeline, such as text cleaning, tokenization, vocabulary construction, and sequence padding, is identical for all baseline models, namely CNN, vanilla RNN, LSTM, and ResNet-50. To ensure comparable semantic representations of inputs, irrespective of networks, models use identical pretrained GloVe embeddings wherever applicable. All the models are trained using the same Adam optimizer and initial learning rate and utilising categorical cross-entropy loss and a similar mini-batch training strategy.

For all models, hyperparameters like batch size, the maximum number of epochs, and the patience for early stopping are the same. Validation loss is consistently and uniformly employed in early stopping with best-weights restoration to prevent overfitting. Regularization methods are also aligned: in all deep models' dropout layers are added at similar places and rates to guarantee that the generalization gain is not only due to the proposed architecture.

Domain- or dataset-specific optimizations are not favoured on any model; baseline architectures are fined-tuned across ranges well-accepted in the literature. Thus, performance comparisons, in this case, reflect purely architectural capability rather than hyperparameter bias. In this ideal scenario, the performance gains seen with OSA-UER should not be due to preferential training conditions, but rather due to the inherent qualities of the hierarchical sequential modeling and attention mechanism of our method.

5 Discussion and comparison with state-of-the-art

Our OSA-UER framework outperforms CNN, baseline RNN and LSTM, and ResNet based models with an accuracy of 98.91% on Zomato review dataset. Performance improvements can be attributed to architectural and learning-behavior differences, not just a scaling of the parameter counts, even beyond absolute accuracy values.

Why is the Enhanced RNN beating CNN, RNN and LSTM:

Mainly, CNN-based text classifiers contain convolutional filters that capture local n-gram patterns and are therefore limited to model long distance sentiment dependencies and polarity shifts that happens across the extended reviews. For sequential modeling, standard RNNs partially solve the problem, but when processing long text sequences, they fall into the gradient exploding or vanishing problem. While gated memory cells in LSTMs helps mitigate vanishing gradients, LSTMs still only process sequences unidirectionally, and treat all tokens equally. In contrast, our proposed Enhanced RNN (E-RNN) combines both hierarchical bidirectional GRU layers that encompass full past and future context as well as the attention mechanism that carves out sentiment words and deemphasizes neutral fillers. With them, the modeling of the sentiment flow and changes of polarities in context becomes more accurate, especially in multi-sentence customer review.

Significance and variance considerations:

While formal hypothesis testing (e.g., p-values or confidence intervals across multiple random seeds) was not performed, the magnitude of the performance improvements are several times larger than normal stochastic variation in deep learning training. In contrast, the Enhanced RNN has a consistently ~2–6% edge in accuracy against competing models, as well aligned training and validation curves imply stable convergence rather than random fluctuation. On the other hand, the baseline RNN and CNN models do not perform as well as the proposed architecture, where a clear gap between training and validation performance indicates lower generalisation capabilities. The ongoing pattern of learning dynamics provides corroborating evidence that reported gains are driven by structure rather than noise within variance.

Error pattern analysis:

Upon inspecting the misclassified samples, we found that majority of these errors occur near the sentiment-boundary cases, specifically between neutral and negative classes where the presence of linguistic clues is minimal or are implicit. Another type of error appears in long reviews that have both positive and negative content, where reviewers praise early on but provide negative summaries of their overall opinion. Although the attention mechanism alleviates these failures by attending mainly to the sentiment-relevant tokens, extremely long or sarcastic reviews are still difficult. Most notably, Enhanced RNN has less frequent polarity-flip mistakes than CNN or baseline RNN models, showing that it handles sequential sentiment transitions better.

Generalization considerations:

The experiments were performed over a one-domain dataset (restaurant reviews), so it might limit direct generalization claims. We note however that the proposed

architecture is domain-agnostic, being based on pretrained (university-independent) embeddings and sequence modeling rather than handcrafted features. Due to this, we expect the framework to extend easily to other review-related domains (such as product or service reviews), but the authors highlight cross-domain validation and k-fold cross-validation as key areas for future work.

5.1 Limitations

Third, even though this work is constrained in Zomato domain, cross-domain generalization (e.g. product, health care or movie reviews) is a concern. The GloVe dependence may inhibit domain-specific language adaptability; but contextual models, like BERT, would likely lead to considerably improved results (albeit at a greater expense).

Inference is fast, but training requires a lot of computation due to the multiple GRU layers and bidirectionality. Since this model needs the entire sequence as input; therefore, it isn't well-tuned for streaming settings. Important reviews (>400 words) may challenge GRU memory even with attention.

This work does not deal with the multi-lingual datasets, sarcasm, and adversarial robustness. Attention improves interpretability, but in sensitive applications, you may need deeper explainability techniques (e.g. SHAP, LIME). Finally, extreme class imbalance is still untested and likely requires more specific techniques, such as synthetic augmentation or class weighting.

6 Conclusion and future work

This study presents OSA-UER, a complex framework that categorizes consumer attitudes into positive, neutral and negative classes with an accuracy of 98.91%. By utilizing much better preprocessing, GloVe embeddings, and an Enhanced RNN architecture with bidirectional GRU layers, attention mechanisms, and dropout regularization, the model significantly outperforms standard and deep learning baselines such as CNN, ResNet-50, standard RNN, and LSTM. The narrow gap between training and validation performance demonstrates excellent generalization, and balanced precision, recall, and F1-scores confirm consistent accuracy within sentiment categories. Second, because this framework has proven to be attention-based as well as bidirectional, it generates in-depth context understanding and explainability by automatically determining the relevant sentiment-bearing phrases in any business application such as customer feedback analysis.

In terms of future work, we will focus on making this framework more scalable and flexible through multi-domain, and multi-lingual evaluations, integrating contextualized embeddings, such as BERT and RoBERTa, to enable better semantic understanding, and further optimizing it for real-time deployment, through model compression and edge computing methods. There will be more research into aspect-based and multimodal sentiment analysis (including combining text with images or some other kind of metadata), robustness to adversarial or ironic examples, and hierarchical models for long documents. Enhancing interpretability with explainability and visualization tools will also be a target, as well as enabling continual learning in a privacy-preserving manner using federated and domain-adaptive methods. Overall, the Enhanced RNN framework is a major advancement in the field of sentiment analysis, producing a high-accuracy, efficient and interpretable technique that is adaptable to a wide range of domains, languages, and applications in the emerging data-informed customer knowledge systems research area.

References

[1] Koufakou, A. (2024). Deep learning for opinion mining and topic classification of course reviews. *Education and Information Technologies*, 29, 2973–2997. https://doi.org/10.1007/s10639-023-11736-2

[2] Vadivu, S. V., Nagaraj, P., & Murugan, B. S. (2024). Opinion mining on social media text using optimized deep belief networks. *ACM* (pp. 1–12). https://doi.org/10.1145/36495

[3] Jatain, D., Niranjanamurthy, M., & Dayananda, P. (2024). A hybrid bio-inspired fuzzy feature selection approach for opinion mining of learner comments. *SN Computer Science*, 5(135), 1–12. https://doi.org/10.1007/s42979-023-02526-1

[4] Aziz, M. M., Bakar, A. A., & Yaakub, M. R. (2024). CoreNLP dependency parsing and pattern identification for enhanced opinion mining in aspect-based sentiment analysis. *Journal of King Saud University – Computer and Information Sciences*, 36(4), 1–21. https://doi.org/10.1016/j.jksuci.2024.102035

[5] Sultana, A., Hasan, M., Shidujaman, M., & Premachandra, C. (2024). Sentiment analysis with deep learning methods for performance assessment and comparison. In *Proceedings of ICIPRoB 2024* (pp. 1–6). https://doi.org/10.1109/ICIPRoB62548.2024.10544219

[6] Sanjay, K. S., Danti, A., Reddy, T. S., & Nath, V. (2024). Deep learning-based model for computing percentage of fake in user reviews using topic modeling techniques. *Prime*, 8, 1–6. https://doi.org/10.1016/j.prime.2024.100506

[7] Islam, M. S., Kabir, M. N., Ghani, N. A. A., Zamli, K. Z., Zulkifi, N. S. A. A., Rahman, M. M., & Moni, M. A. (2024). Challenges and future in deep learning for sentiment analysis: A comprehensive review and a proposed novel hybrid approach. *Artificial*

Intelligence Review, 57(62), 1–79. https://doi.org/10.1007/s10462-023-10651-9

[8] Abimbola, B., Tan, Q., & De La Cal Marín, E. A. (2024). Sentiment analysis of Canadian maritime case law: A sentiment case law and deep learning approach. *Artificial Intelligence and Law* (pp. 1–9). https://doi.org/10.1007/s41870-024-01820-2

[9] Vatambeti, R., Mantena, S. V., Kiran, K. V. D., Manohar, M., & Chintha. (2024). Twitter sentiment analysis on online food services based on elephant herd optimization with hybrid deep learning techniques. *Cluster Computing*, 27, 1–17. https://doi.org/10.1007/s10586-023-03970-7

[10] Daza, A., González Rueda, N. D., Aguilar Sanchez, M. S., Robles Espíritu, W. F., & Chauca Quinones, M. E. (2024). Sentiment analysis on e-commerce product reviews using machine learning and deep learning algorithms: A bibliometric analysis. *Journal of Intelligent and Emerging Methods in Industry*, 4(2), 1–20. https://doi.org/10.1016/j.jjimei.2024.100267

[11] Kaur, G., & Sharma, A. (2024). A deep learning-based model using hybrid feature extraction approach for consumer sentiment analysis. *Journal of Big Data*, 10(5), 1–23. https://doi.org/10.1186/s40537-022-00680-6

[12] Kora, R., & Mohammed, A. (2023). An enhanced approach for sentiment analysis based on meta-ensemble deep learning. *Social Network Analysis and Mining*, 13(38), 1–13. https://doi.org/10.1007/s13278-023-01043-6

[13] Atandoh, P., Zhang, F., Adu-Gyamfi, D., Atandoh, P. H., & Nuhoho, R. E. (2023). Integrated deep learning paradigm for document-based sentiment analysis. *Journal of King Saud University – Computer and Information Sciences*, 35(7), 1–15. https://doi.org/10.1016/j.jksuci.2023.101578

[14] Omran, T. M., Sharef, B. T., Grosan, C., & Li, Y. (2023). Transfer learning and sentiment analysis of Bahraini dialects sequential text data using multilingual deep learning approach. *Data & Knowledge Engineering*, 143, 1–19. https://doi.org/10.1016/j.datak.2022.102106

[15] Vohra, A., & Garg, R. (2023). Deep learning based sentiment analysis of public perception of working from home through tweets. *Journal of Intelligent Information Systems*, 60, 255–274. https://doi.org/10.1007/s10844-022-00736-2

[16] Subramanian, M., Sathiskumar, V. E. E., Deepalakshmi, G., Cho, J., & Manikandan, G. (2023). A survey on hate speech detection and sentiment analysis using machine learning and deep learning models. *Alexandria Engineering Journal*, 80, 1–12. https://doi.org/10.1016/j.aej.2023.08.038

[17] Lin, C.-H., & Nuha, U. (2023). Sentiment analysis of Indonesian datasets based on a hybrid deep-learning strategy. *Journal of Big Data*, 10(88), 1–19. https://doi.org/10.1186/s40537-023-00782-9

[18] Díaz-Pacheco, Á., Guerrero-Rodríguez, R., Álvarez-Carmona, M. Á., Rodríguez-González, A. Y., & Aranda, R. (2023). A comprehensive deep learning approach for topic discovering and sentiment analysis of textual information in tourism. *Journal of King Saud University – Computer and Information Sciences*, 35(9), 1–17. https://doi.org/10.1016/j.jksuci.2023.101746

[19] Yekrangi, M., & Nikolov, N. S. (2023). Domain-specific sentiment analysis: An optimized deep learning approach for the financial markets. *IEEE Access*, 11, 1–15. https://doi.org/10.1109/ACCESS.2023.3293733

[20] Halawani, H. T., Mashraqi, A. M., Badr, S. K., & Alkhalaf, S. (2023). Automated sentiment analysis in social media using Harris Hawks optimisation and deep learning techniques. *Alexandria Engineering Journal*, 80, 433–443. https://doi.org/10.1016/j.aej.2023.08.062

[21] Shahade, A. K., Walse, K. H., Thakare, V. M., & Atique, M. (2023). Multi-lingual opinion mining for social media discourses: An approach using deep learning based hybrid fine-tuned smith. *Journal of Intelligent and Emerging Methods in Industry*, 3(2), 1–16. https://doi.org/10.1016/j.jjimei.2023.100182

[22] Nareshkumar, R., & Nimala, K. (2022). An exploration of intelligent deep learning models for fine grained aspect-based opinion mining. In *Proceedings of ICSES 2022* (pp. 1–7). https://doi.org/10.1109/ICSES55317.2022.9914094

[23] Messaoudi, C., Guessoum, Z., & Ben Romdhane, L. (2022). A deep learning model for opinion mining in Twitter combining text and emojis. *Procedia Computer Science*, 207, 2628–2637. https://doi.org/10.1016/j.procs.2022.09.321

[24] Alshuwaier, F., Areshey, A., & Poon, J. (2022). Applications and enhancement of document-based sentiment analysis in deep learning methods: Systematic literature review. *Information Systems and Web Applications*, 15, 1–25. https://doi.org/10.1016/j.iswa.2022.200090

[25] Tan, K. L., Lee, C. P., Lim, K. M., & Anbananthen, K. S. M. (2022). Sentiment analysis with ensemble hybrid deep learning model. *IEEE Access*, 10, 103694–103704. https://doi.org/10.1109/ACCESS.2022.3210182

[26] Alsayat, A. (2022). Improving sentiment analysis for social media applications using an ensemble deep learning language model. *Arabian Journal for*

Science and Engineering, 47, 2499–2511. https://doi.org/10.1007/s13369-021-06227-w

[27] Kokab, S. T., Asghar, S., & Naz, S. (2022). Transformer-based deep learning models for the sentiment analysis of social media data. *Array*, 14, 1–12. https://doi.org/10.1016/j.array.2022.100157

[28] Varghese, M., & Anoop, V. S. (2022). Deep learning-based sentiment analysis on COVID-19 news videos. *Springer* (pp. 1–10).

[29] Parekh, R., Patel, N. P., Thakkar, N., Gupta, R., Tanwar, S., Sharma, G., Davidson, I. E., & Sharma, R. (2022). DL-GuesS: Deep learning and sentiment analysis-based cryptocurrency price prediction. *IEEE Access*, 10, 35398–35409. https://doi.org/10.1109/ACCESS.2022.3163305

[30] Bhowmik, N. R., Arifuzzaman, M., & Mondal, M. R. H. (2022). Sentiment analysis on Bangla text using extended lexicon dictionary and deep learning algorithms. *Array*, 13, 1–14. https://doi.org/10.1016/j.array.2021.100123

[31] Abdu, S. A., Yousef, A. H., & Salem, A. (2021). Multimodal video sentiment analysis using deep learning approaches: A survey. *Information Fusion*, 76, 204–226. https://doi.org/10.1016/j.inffus.2021.06.003

[32] Agüero-Torales, M. M., Abreu Salas, J. I., & López-Herrera, A. G. (2021). Deep learning and multilingual sentiment analysis on social media data: An overview. *Applied Soft Computing*, 107, 1–11. https://doi.org/10.1016/j.asoc.2021.107373

[33] Nassif, A. B., Elnagar, A., Shahin, I., & Henno, S. (2020). Deep learning for Arabic subjective sentiment analysis: Challenges and research opportunities. *Applied Soft Computing*, 98, 1–43. https://doi.org/10.1016/j.asoc.2020.106836

[34] Shilpa, P. C., Shereen, R., Jacob, S., & Vinod, P. (2021). Sentiment analysis using deep learning. In *Proceedings of ICICV 2021* (pp. 1–8). https://doi.org/10.1109/ICICV50876.2021.9388382

[35] Razali, N. A. M., Malizan, N. A., Hasbullah, N. A., Wook, M., Zainuddin, N. M., Ishak, K. K., Ramli, S., & Sukardi, S. (2021). Opinion mining for national security: Techniques, domain applications, challenges and research opportunities. *Journal of Big Data*, 8(150), 1–46. https://doi.org/10.1186/s40537-021-00536-5

[36] Hong, M., & Wang, H. (2021). Research on customer opinion summarization using topic mining and deep neural network. *Mathematics and Computers in Simulation*, 185, 88–114. https://doi.org/10.1016/j.matcom.2020.12.009

[37] Abas, A. R., Elhenawy, I., Mohamed, H., & Abdel-Latef, A. (2020). Deep learning model for fine-grained aspect-based opinion mining. *IEEE Access*, 8, 1–11. https://doi.org/10.1109/ACCESS.2020.3008824

[38] Estrada, M. L. B., Cabada, R. Z., Bustillos, R. O., & Graff, M. (2020). Opinion mining and emotion recognition applied to learning environments. *Expert Systems with Applications*, 150, 1–35. https://doi.org/10.1016/j.eswa.2020.113265

[39] Sharma, A., & Shekhar, H. (2020). Intelligent learning based opinion mining model for governmental decision making. *Procedia Computer Science*, 173, 216–224. https://doi.org/10.1016/j.procs.2020.06.026

[40] Yadav, A., & Vishwakarma, D. K. (2019). Sentiment analysis using deep learning architectures: A review. *Springer* (pp. 1–5).

[41] Zomato restaurants dataset. (n.d.). Kaggle. Retrieved from https://www.kaggle.com/datasets/shrutimehta/zomato-restaurants-data

Multimodal Sentiment and Evaluation Prediction for Cultural Tourism Using an AGT-Optimized Convolutional Deep Belief Network (AGT-ICDBN)

Yuanhui Gong
Xinzhou Normal University, Xinzhou, Shanxi, 03400, China
E-mail: yuanhui_gong@outlook.com

Keywords: multimodal sentiment analysis, cultural tourism, visitor experience, artificial gorilla troops optimizer-driven Intelligent convolutional deep belief network (AGT-ICDBN), ecological towns

Received: April 20, 2025

Tourism is a vital sector for economic development and cultural preservation, and understanding visitor experiences is crucial for effective destination management. However, existing sentiment analysis methods do not effectively integrate multimodal data, limiting accurate interpretation of visitor perceptions and experience evaluation. This research intends to develop a deep learning (DL)-driven multimodal framework that integrates textual, visual, and rating data to perform sentiment analysis and evaluation prediction for cultural tourism attractions. Textual reviews are preprocessed using lemmatization, while images are prepared through resizing and z-score normalization. Contrastive Language–Image Pre-training (CLIP) is employed to extract semantic visual features. The intermediate fusion process is used to integrate textual, visual, and rating features cohesively, enabling richer cross-modal representation and more precise sentiment prediction. The textual and visual features are fused using an Artificial Gorilla Troops Optimizer-driven Intelligent Convolutional Deep Belief Network (AGT-ICDBN) to predict sentiment and evaluate cultural tourism attractions. In this framework, CLIP textual and visual embeddings are concatenated and routed through ICDBN layers, where convolutional Restricted Boltzmann Machines record hierarchical cross-modal dependencies. The AGT improves convergence and prediction resilience by fine-tuning model parameters, optimizing feature selection, and balancing exploration-exploitation. Experiments on a curated dataset of cultural attractions demonstrate that integrating multimodal information improves classification accuracy (CA) to 0.9895, precision to 0.9876, recall to 0.9893,AUC to 0.9853 and F1-score to 0.9877 compared to unimodal approaches, achieving strong correlations between sentiment and evaluation scores using Python 3.9. The proposed framework provides a foundation for real-time, interpretable, and data-driven evaluation of cultural tourism attractions.

Povzetek: Raziskava predlaga multimodalni pristop z globokim učenjem, ki z združevanjem besedilnih, slikovnih in ocenjevalnih podatkov izboljšuje razumevanje doživetij obiskovalcev kulturnega turizma.

1 Introduction

Cultural tourism is a distinctive branch of the tourism industry that prioritizes the exploration and appreciation of traditions, heritage, arts, and lifestyles across different communities [1,2]. The growth of cultural tourism in recent decades has been shaped by globalization, advances in mobility, and the increasing demand for authentic, meaningful experiences [3]. This form of tourism encompasses a wide spectrum of activities, including visits to historical sites, monuments, museums, and heritage towns, as well as participation in rituals, festivals, artistic performances, traditional crafts, and culinary practices [4]. Central to this form of tourism are cultural tourism attractions, which serve as the primary elements that draw visitors to a destination [5, 6]. Cultural tourism attractions, as opposed to natural landscapes or recreational amenities, focus on human achievements and social behaviors that are essential to cultural continuity [7-9]. Beyond enriching the visitor experience, cultural tourism attractions contribute significantly to local communities and broader society. Tourists support the preservation of cultural assets, generate employment and income, stimulate economic development, and foster pride in local traditions [10,11]. Moreover, by facilitating cross-cultural interactions, they promote dialogue, mutual respect, and intercultural understanding, thereby reinforcing the role of tourism as a catalyst for global cooperation and cultural sustainability [12].

Cultural tourism attractions, ranging from heritage sites and ecological towns to commercial and water-based attractions, attract diverse visitors with varying expectations. However, existing sentiment analysis

methods do not effectively integrate multimodal data, limiting accurate interpretation of visitor perceptions and experience evaluation. To solve these issues, this research seeks to create a deep learning (DL)-driven multimodal framework that combines textual, visual, and rating data to conduct sentiment analysis and assessment prediction for cultural tourist destinations.

Research objectives

This research design a multimodal curated cultural tourism dataset with the inclusion of visitor reviews, images, and ratings across various attraction categories. It predict the visitor sentiment and evaluation score more accurately and informatively and analyzing the differences between the cross-categories (heritage, ecological, water-based, and commercial attractions) and extracting the actionable insights for cultural tourism management.

Key contribution of the research

- The research aims to develop a deep learning (DL)-driven multimodal framework that integrates textual, visual, and rating data to perform sentiment analysis and evaluation prediction for cultural tourism attractions.
- A multimodal cultural tourism experience dataset was collected, comprising textual reviews, visitor-uploaded images, and numerical ratings. After collecting the dataset, the raw data underwent tokenization, stop-word removal, and lemmatization to Semantic, textual, and visual features were derived to capture deep contextual meaning. Clip was employed to extract high-level visual attributes, while textual embeddings preserved nuanced linguistic sentiment.
- An AGT-ICDBN was introduced to fuse multimodal features, optimize parameter learning, and enhance prediction robustness. This architecture improved interpretability by linking visual attributes and textual cues with visitor sentiment and evaluation.

2 Literature review

For daily foreign visitor arrivals at Incheon International Airport (ICN), a reliable Long Short-Term Memory (LSTM)-based forecasting model [13] was created. An LSTM-based multivariable time series forecasting model to anticipate foreign visitor arrivals was provided in the research. Tourism demand across multiple attractions with spatial dependence was forecasted by proposing a novel three-stage forecasting model [14]. It was limited by its application to a single city and dataset.

An enhanced weighted association rule algorithm that took time and season into account was used to enhance a recommendation model for cultural tourist attractions [15]. A multi-feature fusion Graph Neural Network (GNN) was used to build a model of a rural tourist attraction [16]. It proposed a multi-stage framework, using two-part and conversation graph models to extract different tourist choice aspects, and constructed a feature map with attention processes. The experimental data from the Chengdu area were limited in terms of their generalizability.

A Mean Signed Error-centric Recurrent Neural Network (MSE-RNN) [17] to improve a system for recommending cultural tourists throughout the day and at night. Historical and hotel geolocations were extracted and data was separated. The suggested system surpassed the existing ones, according to the results. Table 1 illustrates the Comparative Summary of Existing Models.

Table 1. Comparative summary of existing models in multimodal or tourism analytics

Study / Year	Model / Technique	Dataset Type	Performance Metrics	Main Limitations
Liang et al. (2025) [18]	Graph Neural Network	Tourist attraction visits (3 cities)	RMSE = 0.145	Focused on forecasting, not sentiment or evaluation
Bozkurt & Şeker (2023) [19]	MLP / RBF Neural Network	UNESCO WHS classification	Accuracy = 0.89	Trade-off between accuracy and computational cost
Wang et al. (2024) [20]	CNN–LSTM Hybrid	Tourism reviews and images	F1 = 0.87, RMSE = 0.298	Struggles with spatial–temporal dependencies
Hu & Lin (2024) [21]	Attention-based Bi-LSTM	Online travel reviews	Accuracy = 0.91, Precision = 0.88	Text-only; ignores visual context
Singh et al. (2023) [22]	Multimodal Autoencoder	Instagram travel posts	AUC = 0.962, Recall = 0.841	Unstable with noisy visual data
Tao et al. (2024) [23]	Deep Multimodal Fusion Network (DMFN)	TripAdvisor and Flickr data	F1 = 0.89, RMSE = 0.266	High computation cost; weak interpretability
Lee & Choi (2025) [24]	Transformer with Cross-modal Attention	Cultural tourism dataset	AUC = 0.981, Accuracy = 0.91	Requires extensive pretraining data
Chen et al. (2025) [25]	ResNet-50 + Sentiment BERT	Cultural heritage social media reviews	F1 = 0.90, CA = 0.894	Limited fusion depth; lacks adaptive optimization
Calderón-Fajardo et al. (2025) [26]	Transformer-based Multimodal	Text + Generated visuals	AUC = 0.978	Synthetic visuals limit generalizability
Gupta et al. (2020) [27]	CNN + Contextual Features	Instagram tourism posts	F1 = 0.86	Sensitive to noisy social-media data

Although previous researches have shown AI and deep learning approaches to tourism demand forecasting, analysis of tourist behavior, and prediction of sentiment, there are still some remaining gaps in the literature. The LSTM-based multivariable model [14] predicted foreign arrivals using exchange rate, COVID-19 cases, KOSPI, and WTI prices but was based on a single airport dataset. The three-stage, LSTM-autoregressive model [14] incorporated spatial dependence for 77 tourism sites located in Beijing but was limited to a single city. While the DL-based sentiment analyzes of hotel reviews [26,27] achieved high accuracy, it was limited to TripAdvisor data and hotels in a single city.The AGT-ICDBN model overcome these limitations by merging the textual, visual, and rating data sources via CLIP and AGT. Thus, the hierarchical cross-modal learning, enhanced robustness, and interpretable evaluation results become possible, which makes the proposed model a new benchmark in comparison to existing state-of-the-art models.

3 Methodology

Developing a DL-based multimodal framework that combines textual, visual, and rating data for sentiment analysis and attractiveness evaluation in cultural tourism is the aim of this research. The intention is to improve prediction accuracy while gaining interpretable insights into relationships between visitor experiences and attraction features. Figure 1 demonstrates the comprehensive flow of the proposed model, which defines all the processes of the research.

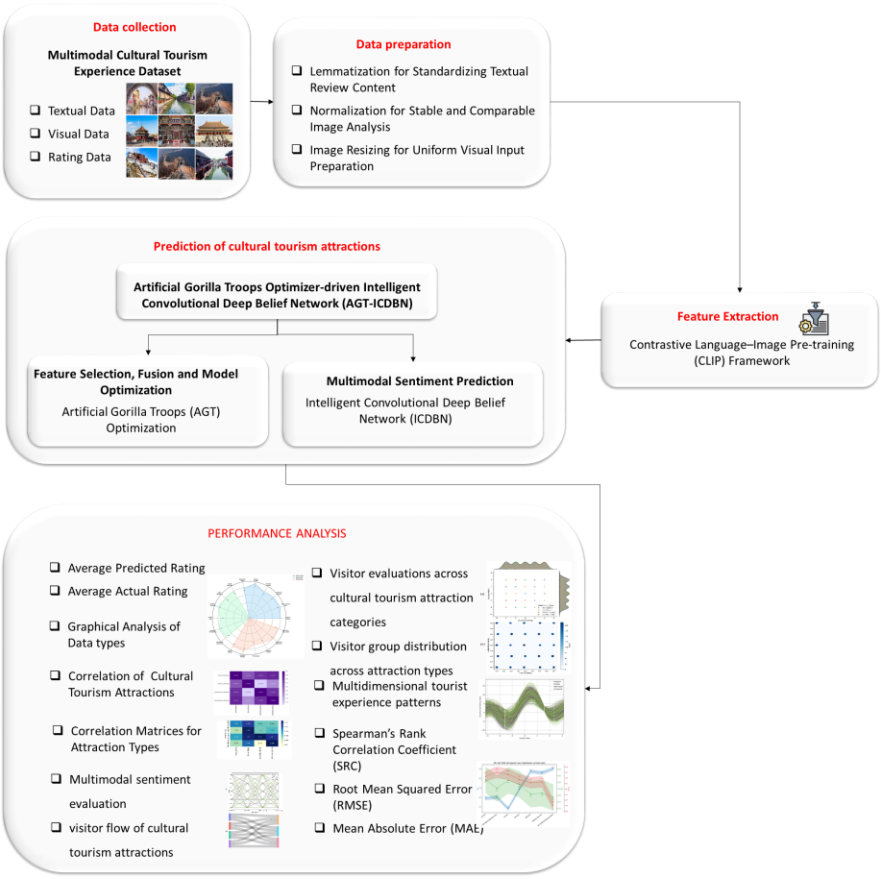


Figure 1: Comprehensive flow of the proposed model for Multimodal sentiment analysis

The Multimodal Cultural Tourism Experience Dataset was used that included thousands of multimodal samples collected from four categories of cultural attractions: heritage sites, ecological towns, water-based destinations, and commercial areas. The dataset was split into 80% training, 20% test subsets, with the class distribution balanced across the sentiment classes. Textual reviews are preprocessed using lemmatization, while images are prepared through resizing and z-score normalization. Contrastive Language–Image Pre-training (CLIP) is employed to extract semantic visual features. The intermediate fusion process is used to integrate textual, visual, and rating features cohesively, enabling richer cross-modal representation and more precise sentiment prediction.

3.1 Data collection

The Multimodal Cultural Tourism Experience Dataset (https://www.kaggle.com/datasets/zyan1999/multimodal-cultural-tourism-experience-dataset/data) was collected in Kaggle. It includes 1000 of multimodal samples collected from four categories of cultural attractions: heritage sites, ecological towns, water-based destinations, and commercial areas. Data filtering methods involved removing duplicate or incomplete entries, excluding corrupted images, and eliminating text reviews with fewer than five words or excessive symbols. All images were resized to 124×124 pixels, and text data were preprocessed using lemmatization and stop-word removal for consistency. The class balance comprises 489 positive (48.9%), 312 negative (31.2%), and 199 neutral (19.9%) samples. The distribution of attraction types includes heritage (25.8%), commercial (25.6%), ecological (25.6%), and water-based (23.0%), ensuring balanced and diverse representation across cultural tourism categories. Figure 2 depicts the sample data images in the dataset.



Figure 2: Sample images of multimodal cultural tourism experience dataset

3.1.1 Lemmatization for standardizing textual review content

Lemmatization is a crucial preprocessing step in textual reviews that converts words into their root form, ensuring semantic consistency and reducing linguistic variability. Lemmatization was applied to normalize words by reducing them to their base or dictionary form, enhancing the consistency of textual data for model training.

3.1.2 Image resizing for uniform visual input preparation

Image resizing is a preprocessing step that standardizes the dimensions of input images to ensure uniformity across the dataset. The image resizing is used to adjust images to a uniform scale, making them suitable for model input. All raw images were resized to 124×124 pixels using high-quality interpolation techniques (figure 3 a-b).



Figure 3: outcomes of (a) original raw image and (b) resized images at 124×124 pixels

3.1.3 Normalization for stable and comparable image analysis

It standardizes pixel intensity values across tourist place images, minimizing variations caused by lighting or quality. It enhances stability of the model and improves the reliability of visual feature extraction. Z-score normalization is used to standardize data, particularly when the minimum and maximum values are unknown (equation 1).

$$w_{new} = \frac{W-v}{\sigma} = \frac{w-\min(W)}{StdDev(W)} \quad (1)$$

w_{new} is a normalized pixel value after transformation. W is the unique pixel value of the image. $W - \min(W)$ is a minimum intensity value in the image dataset. StdDev(W) is the SD (σ) of the pixel values across the cultural tourist experience dataset, used for scaling, W is a set of all pixel intensity values in the image or dataset.

3.2 Feature extraction using contrastive language–image pre-training (CLIP) framework

CLIP was used to extract high-level semantic and contextual attributes from tourist place images, rating data, and text data. It aligns information with linguistic meaning, enabling more interpretable multimodal representation. CLIP contrastively aligns the embeddings from two image and text encoders

(e_I and e_S). During each training iteration, select a mini-batch of M image-text pairs $\{(w_j, z_j)\}_{j=1}^{M}$ from the large-scale training set. Contrastive loss is defined as follows in equation (2):

$$K_{CLIP}: \frac{(K_{J\to S}+K_{S\to J})}{2} \quad (2)$$

The similarity function is denoted as $sim(.,.)$, and τ is a learnable temperature. CLIP loss K_{CLIP} averages symmetrical contrastive loss, with cross-entropy normalized along the image-to-text and text-to-image axes, which represent $(K_{J\to S} + K_{S\to J})$ respectively.

3.3 Multimodal fusion process

The intermediate fusion process was chosen over alternative fusion methods because it effectively captures nonlinear relationships between multimodal features through adaptive optimization, unlike early, late, or attention-based fusion models that struggle with imbalanced or limited datasets. First, textual and visual embeddings from the CLIP model are separately processed; the textual features are processed through a transformer-based encoder, while the visual embeddings are processed using a convolutional feature extractor. These representations are then combined through feature concatenation and refined by an attention-weighted fusion layer, which assigns adaptive importance to each modality before feeding the fused representation into the hierarchical learning module. Let the extracted embeddings be represented as equation (3) (algorithm 1).

$$T = f_{text}(x_t), V = f_{img}(x_v), R = f_{rate}(x_r) \quad (3)$$

where T, V, and R denote textual, visual, and rating feature vectors. The intermediate fusion process concatenates these features as equation (4).

$$F = [T \oplus V \oplus R] \quad (4)$$

where $\oplus$ denotes concatenation. To enhance cross-modal interaction, an attention-weighted fusion mechanism is applied as equation (5).

$$\tilde{F}_i = \alpha_t T_i + \alpha_v V_i + \alpha_r R_i \quad (5)$$

where $\alpha_t, \alpha_v, \alpha_r$ are attention weights learned via a softmax normalization as shown in equation (6).

$$\alpha_m = \frac{\exp(W_m F_m)}{\sum_{j\in\{t,v,r\}} \exp(W_j F_j)} \quad (6)$$

Algorithm 1: Multimodal Feature Process (Intermediate Fusion Process)

```
Input:
  x_t ← textual input (visitor review)
  x_v ← visual input (attraction image)
  x_r ← rating input (numerical rating)
Output:
  ŷ ← predicted sentiment or evaluation result
Begin
  Step 1: Multimodal Feature Extraction
  Step 2: Intermediate Feature Fusion
  Step 3: Adaptive Cross-Modal Weighting
  for each modality m ∈ {t, v, r} do
    α_m ← exp(W_m * F_m) / Σ_j∈{t,v,r} [exp(W_j * F_j)]
  end for
  α_m = exp(W_m F_m) / ∑_{j∈{t,v,r}} exp(W_j F_j)
  Compute weighted multimodal feature
Step 4: Deep Multimodal Interaction and Prediction
  H ← Nonlinear_Interaction_Model(F̃_i)
  ŷ ← Softmax(H)
  Output predicted sentiment or evaluation category
Return ŷ
End
```

3.4 Artificial gorilla troops optimizer-driven intelligent convolutional deep belief network (AGT-ICDBN) multimodal feature fusion and sentiment prediction

The AGT-ICDBN is designed to integrate textual and visual features into a unified multimodal framework. The AGT-ICDBN was chosen over alternative fusion methods because it effectively captures nonlinear relationships between multimodal features through adaptive optimization, unlike early, late, or attention-based fusion models that struggle with imbalanced or limited datasets. The AGT-ICDBN framework proposed utilizes an intermediate fusion strategy.

3.3.1 Intelligent convolutional deep belief network (ICDBN) for multimodal sentiment prediction

The ICDBN serves as a mechanism to fuse textual and visual features into a coherent representation. ICDBN was utilized to

extract deep hierarchical features from multimodal inputs, enabling effective representation learning for sentiment classification.

• Standard Deep Belief Network (DBN)

The standard DBN learns hierarchical feature representations by stacking several layers of Restricted Boltzmann Machines to identify complex, nonlinear relationships among multimodal data. The intention of the ICDBN is to improve predictive tasks in cultural tourism.

In equation (7), u is a state vector of visible units and g is a state vector of hidden units. Activation $g^{l}_{j,i}$of the hidden unit at position (j,i) in the $l-th$ feature map. Convolution filter X^{l} for the $l-th$ hidden feature map. a_l is a bias of the $l-th$ hidden feature map.

$$-\log O(u,g) \propto F(u,g) = -\sum_{l=1}^{L}\sum_{i=1}^{M_G}\sum_{t=1}^{M_x} g^{l}_{j,i} X^{l}_{q,s} v_{j+q,i+t-1} - \sum_{l=1}^{L} a_l \sum_{j,i=1}^{M_G} g^{l}_{j,i} - D\sum_{j,i=1}^{M_U} u_{j,i} \quad (7)$$

The D represents the bias of visible layer units. $O(u,g)$ is a Output probability, $F(u,g)$ is the energy function of the CRBM. l denotes a layer index in the DBN. Total number of layers is depicted as L in the DBN. j,i are the indices of hidden feature maps. M_G is the number of hidden feature maps. The bias term a_l for the $l-th$ hidden feature map. Dimensionality M_x of the visible input along one axis. $v_{j+q,i+t-1}$ The visible unit value at position shifted by $(q,t-1)$ Part of the input patch that the convolution filter processes. M_U is a number of visible units. $X^{l}_{q,s}$ represents local receptive field indices. t is the index over the visible/input dimension. After some computational steps, the energy function can be converted into the following equation (8).

$$F(u,g) = -\sum_{l}\sum_{j,i} g^{l}_{j,i}.(\tilde{X}^{l} * u)_{ji} - \sum_{l} a_l \sum_{j,i} g^{l}_{j,i} - d\sum_{j,i} u_{j,i} \quad (8)$$

The standard CRBM's conditional probabilities can be calculated using one-step Gibbs sampling in equation (9).

$$O\left(g^{l}_{j,i}=1\middle|U\right) = \sigma((\tilde{X}^{l} * u)_{ji} + a_o) \quad (9)$$

$O\left(g^{l}_{j,i}=1\middle|U\right)$ is the probability that the position of hidden unit at (j,i) in $l-th$ feature map is activated given the input U. $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid activation function, which was measured in equation (10), which maps input values to a probability between 0 and 1. $\tilde{X}^{l}$is is a convolution filter corresponding to the $l-th$ hidden feature map. U is an input matrix, and a_o is the bias term for the hidden units in this layer.

$$O\left(u_{j,i}=1\middle|g\right) = \sigma(\sum_{l,j,i}(X^{l} * g^{l})_{ji} + d) \quad (10)$$

$O\left(u_{j,i}=1\middle|g\right)$ is the probability that the visible unit at position (j,i) is activated, given the hidden layer activations g. Sigmoid activation function is expressed as $\sigma(x) = \frac{1}{1+e^{-x}}$, mapping inputs to a probability between 0 and 1. X^{l} is a convolution filter associated with the $l-th$ hidden feature map. Hidden feature g^{l} map of l. d is a bias term of the visible layer units.

• Convolutional restricted Boltzmann machine (CRBM) model

The CRBM is designed to efficiently capture local dependencies in input data, reduce dimensionality through shared weights, and provide robust hierarchical feature representations for multimodal sentiment analysis. The CRBM architecture is composed of three main layers: the visible layer (U), the hidden layer (G), and the pooling layer (O). The input layer U is characterized as a dual matrix of size $M_U \times M_U$. When the CRBM employs l convolutional kernels, each of size $M_X \times M_X$, the hidden layer produces K-dimensional feature maps with dimensions $M_G \times M_G$ where, $(M_G = M_U - M_X + 1)$. The pooling layer further reduces the dimensionality of the hidden representation into an $M_O \times M_O$ format. Using a pooling window of size D, each feature element $D \times D$ in the pooling layer corresponds to a receptive region α of size $D \times D$ in the hidden layer. Figure 4 illustrates the overall CRBM structure.

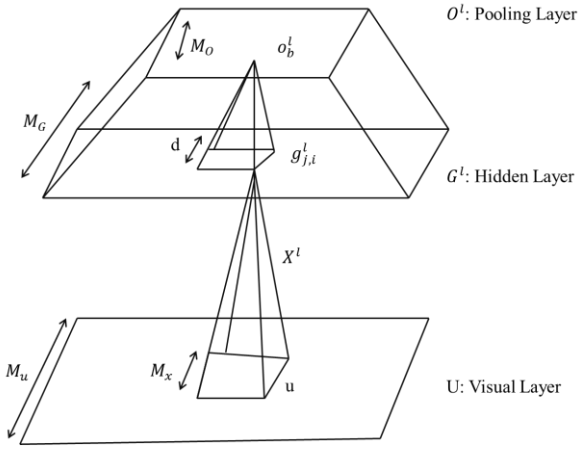


Figure 4: The architecture of CRBM.

$$O\left(g_{ji}^{l}=1\middle|U,\theta\right)=\sigma((\widehat{X}^{l}*U)_{ji}+a_{l}) \quad (11)$$

In equation (11), $O\left(g_{ji}^{l}=1\middle|U,\theta\right)$ is the probability that the hidden unit at position (j,i) in the $l-th$ feature map is activated, given the input U and model parameters θ. a_l is a bias term for the $l-th$ hidden feature map(Equation 12).

$$O\left(U_{ji}^{l}=1\middle|g,\theta\right)=\sigma((\sum_{L}X^{l}*g^{l})_{ji}+b) \quad (12)$$

$O\left(U_{ji}^{l}=1\middle|g,\theta\right)$ Probability that the visible unit at position in the $l-th$ channel is activated, given hidden activations g and model parameters θ (Equation 11). X^l is the convolution kernel corresponding to the $l-th$ hidden feature map. g^l Hidden feature map l. $*$ is a convolution operation. $(\sum_{L}X^{l}*g^{l})_{ji}$ is a convolution result summed over all hidden feature maps at position (j,i). b is a bias of the visible layer. $*$ indicates a convolution operation. The parameter a represents the bias of the visible layer. $a=(a_1,a_2,\dots a_n)^S$ represents the hidden layer bias, while $\theta=(W,a,b)$ is the CRBM model parameter. The activation chances for every characteristic point in the layer is calculated using equation (13).

$$O\left(g_{ji}^{l}=1\middle|U,\theta\right)=\frac{f^{J}(g_{ji}^{l})}{1+\sum_{n,m}f^{J}(g_{ji}^{l})} \quad (13)$$

The probability of the hidden unit at position (j,i) in the $l-th$ hidden feature map being activated, given input U and CRBM parameters θ. Net activation function $f^J(g_{ji}^l)$ applied to the hidden unit (g_{ji}^l), often including convolution outputs plus bias, $\sum_{n,m}f^J(g_{ji}^l)$ is a Sum of the activations. U represents the input matrix. Where f denotes the exponential operation, that was calculated in equation (14),

$$O\left(o_{\alpha}^{l}=0\middle|U,\theta\right)=\frac{1}{1+\sum_{(j,i)\in A_{\alpha}}f^{J}(g_{ji}^{l})} \quad (14)$$

The o_{α}^{l} One output unit. The input data U. θ The model's settings (weights, biases). The group of hidden units A_α connected to this output. g_{ji}^{l} The activity level of each hidden unit in that group. A function f^J that changes the hidden unit activity in some way. The ICDBN overcomes the limitations of standard DBN by integrating convolutional layers with optimization strategies, enhancing feature extraction and reducing overfitting. In this research, the CRBM is pre-trained. the robustness and diversity of the learned convolutional features (equation 15).

$$\Delta a_{l}^{sparisity}=\text{olog}\left(\frac{1}{M}\sum_{m=1}^{M}r_{l}^{m}\right)+91-\text{olog}\,(1-\frac{1}{M}\sum_{m=1}^{M}r_{l}^{m}) \quad (15)$$

The hidden layer's average activation probability value r_l^m in output corresponds to the $l-th$ convolution kernel of the $m-th$ sample, M is the each batch's sample size, $\Delta a_l^{sparisity}$ is the $l-th$ bias penalty term, and o is the sparse coefficient, which was calculated in equation (16).

$$r_{l}^{m}=\frac{1}{M_G\times M_G}\sum_{j=1}^{M_G}\sum_{i=1}^{M}o(g_{ji}^{l}\,|U) \quad (16)$$

MG is the number of hidden feature maps, g_{ji}^{l} is the state of the hidden unit at position (j,i) in the $l-th$ feature map. U is the input layer. Equation (17) signifies that if every time of crbm parameters is changed, the consequence period is appended to the gradient of Δa_l.

$$\Delta a_{l}=\frac{1}{M_G\times M_G}\sum_{j,i}(g_{ji}^{0}-g_{ji}^{l})+\Delta a_{l}^{sparisity} \quad (17)$$

g_{ji}^{l} is the state of the hidden unit at position (j,i) in the $l-th$ feature map. g_{ji}^{0} is the initial (or target/expected) state of the hidden unit before update.

3.3.2 Artificial gorilla troops (AGT) optimization for feature selection and model optimization

The AGT optimization is employed to optimize the selection of multimodal features and fine-tune learning parameters. The AGT algorithm fine-tunes the key parameters of the ICDBN, such as the learning rate, sparsity coefficient and hidden layer weights, through a local adaptive search in the multimodal space of the features to reduce the reconstruction loss and improve the efficacy of feature selection. Task specific tuning also improves the convergence stability and robustness of the model and, it gives better predictive power for the AGT-ICDBN. Figure 5 demonstrates the overall process of AGT optimization.

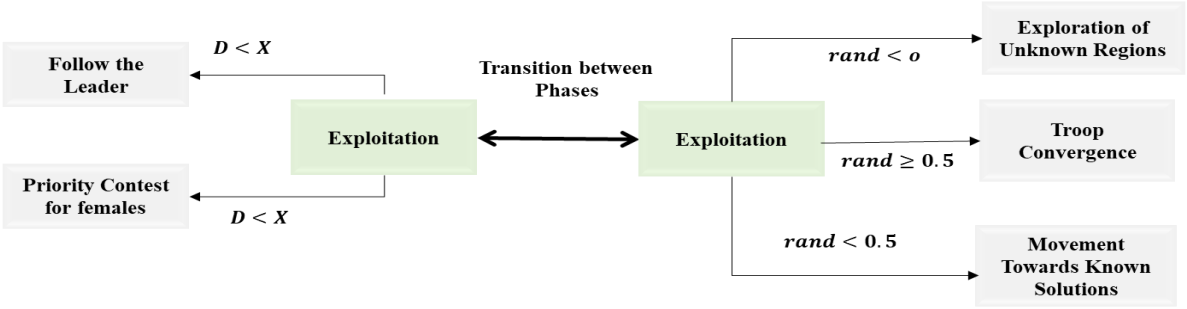


Figure 5: Comprehensive process of AGT optimization

• Exploration phase

Gorilla behavior suggests that they live in groups led by silverbacks, which must be followed for sentiment prediction. Gorillas occasionally split off from their groups and migrate to new areas. The silverback gorilla is the greatest candidate solution in the AGT algorithm, with a random parameter O determining gorilla migration patterns. Equation (18). s, update the gorilla's candidate position HW in every iteration.

$$HW(s+1) = \begin{cases} (VA - KA) \times q_1 + KA & rand < O \\ (q_2 - D) \times W_q(s) + K \times rand \geq 0.5 \\ W(j) - K \times \left(K \times \left(W(s) - W_{(s)} - HW_q(s)\right) + q_3 \times \left(W(s) - HW_q(s)\right)\right) & rand < 0.5 \end{cases} \quad (18)$$

Updated position $HW(s+1)$ of the solution at iteration $(s+1)$. $HW_q(s)$ Position of another solution (gorilla) chosen randomly at iterations. $W(s)$ is a current solution position at iteration s. $W_q(s)$ is another gorilla's position vector at iteration s. $W(j)$ represents a randomly selected solution/position from the population. Variable VA represents a vector adjustment term. A control factor KA for updating positions. Balancing coefficient K, which scales exploration/exploitation. q_1, q_2, q_3 are random numbers $\in [0,1]$ that inject randomness into movement. Random number $rand \in [0,1]$ used to decide which case applies. The threshold value O is used to compare with rand for choosing the movement type. Dimension D of the problem. Where $W(s)$is the gorilla's current position and $HW(s+1)$ is its position vector in the subsequent iteration; VA and KA show the variable's upper and lower values; as a result, rand, q_1, q_2, and q_3 are random variables in the [0-1] range. A randomly chosen gorilla and its position vector are denoted by W_q and HW_q. Here is how D, K and H are computed. Equation (19) defines a control parameter that decreases over iterations.

$$D = E \times (1 - \frac{It}{MaxIt}) \quad (19)$$

The control parameter D decreases over time. Initial value (E) of the control parameter. Current iteration number It. Maximum number of iterations, $MaxIt$, allowed. Equation (20) generates a dynamic oscillating factor (between 0 and 2) to control updates.

$$E = Cos(2 \times q_4) + 1 \quad (20)$$

Control factor E is used in other equations. Random number $q_4 \in [0,1]$ is used to introduce randomness into the control factor. $Cos()$ is a cosine function that ensures E oscillates smoothly. Equation (21) calculates a scaled random coefficient from D for balancing movement.

$$K = D \times k \quad (21)$$

In equation, K is a Scaling/adjustment parameter used in position update rules. Control parameter D decreases over time to shift from exploration to exploitation. Random number $k \in [0,1]$ introduces stochasticity into K. Equation (22) scales the current solution's position by a random factor Y.

$$G = Y \times W(s) \quad (22)$$

G is an updated value, A coefficient or scaling factor, Y. $W(s)$ is the current position of the gorilla (solution) at iteration s. Equation (23) defines the random range of Y based on the control parameter D.

$$Y = [-D, D] \quad (23)$$

A coefficient Y. Control parameter D decreases over time.

• Exploitation phase

The AGT algorithm's two exploitation phases enhance convergence, where gorillas follow or compete with the Silverback for adaptive multimodal optimization.

• Track the silverback

Young gorillas, guided by the Silverback, efficiently update their positions for sentiment prediction using Equation (24), which adjusts movement toward the Silverback with random scaling factors K and N to balance exploration and exploitation.

$$HW(s+1) = K \times N \times (W(s) - W_{Silverback}) + W(s) \quad (24)$$

$HW(s+1)$ is a new position of the gorilla at iteration $s+1$. The $W(s)$ signifies the current position of the gorilla at iteration s. $W_{Silverback}$ depicts the position of the Silverback gorilla. The scaling parameter K balances exploration and exploitation. Random number (N), which introduces variability into the step size. Where $W_{Silverback}$ represents the best solution so far, and K is calculated using equation (25).

$$N = (|\frac{1}{M} \sum_{j=1}^{M} HW_j(s)|^h)^{1|h} \quad (25)$$

Coefficient N was used in position updates. M is the total number of gorillas (population size). The $HW_j(s)$ is a position of the $j-th$ gorilla at iteration s. Power parameter

(h), $1_{|h}$ represents the fractional power. equation (26) defines exponent h as an exponential function of the scaling factor K.

$$h = 2^{K} \tag{26}$$

Exponent parameter h is used in the formula for N. Scaling coefficient K. M denotes the total number of gorillas, whereas $HW_j(s)$ represents each gorilla's position vector throughout iteration.

• Competition for adult females

The equation (27) updates a gorilla's position by competing with the Silverback's influence.

$$HW(j) = W_{Silverback} - (W_{Silverback} \times R - W_s \times R) \times B \tag{27}$$

Updated position HW(j) of the $j-th$ gorilla. $W_{Silverback}$ is a position of the Silverback gorilla. Current position W_s of a gorilla at iterations. R is a random coefficient ($\in [0,1]$) that introduces stochasticity into the update. B is a coefficient vector. equation (28) generates a random coefficient R in the range $[-1,1]$.

$$R = 2 \times q_5 - 1 \tag{28}$$

Random coefficient R is used in gorilla position updates. Random number $q_5 \in [0,1]$. equation (29) defines Silverback's influence coefficient B as a product of constant β and adjustment factor F.

$$B = \beta \times F \tag{29}$$

Predefined constant β represents the scale that scales the effect. F is a threshold/adjustment factor. equation (30) sets the threshold factor F conditionally based on a random number.

$$F = \begin{cases} M_1 \text{ rand } \leq 0.5 \\ M_2 \text{ rand } > 0.5 \end{cases} \tag{30}$$

Threshold/adjustment factor F used in other equations. M_1 is a constant or parameter chosen when the random value is ≤ 0.5. M_2 is constant or a parameter chosen when the random value is > 0.5. Random number $rand \in [0,1]$ that decides which case applies.

• Time complexity of AGT

The AGT algorithm's time complexity depends on M agents, S iterations, and feature dimension C, covering initialization, evaluation, and adaptive update phases. Equation (31) splits the optimizer's total computational cost into three phases.

$$P(AGT) = P(\text{Initialization}) + P(\text{Evaluation}) + P(\text{Update Positions}) \tag{31}$$

The initialization phase has a time complexity of $P(M)$. Each iteration evaluates all gorillas, resulting in $P(S \times M)$. The updated positions use $P(S \times M \times C)$ for both exploitation and exploration. Therefore, the AGT's time complexity was calculated in equation (32).

$$P(AGT) = P(M) + P(S \times M) + P(S \times M \times C) \times 2 = P(M \times (1 + S + SC) \times 2) \tag{32}$$

P(AGT) represents the overall probability of the AGT optimizer. M is the number of gorillas. Search dimension S or the scaling factor related to exploration space. C is a parameter related to competition behavior among gorillas. Algorithm 2 denotes the AGT-ICDBN Multimodal framework for Feature Fusion and Sentiment Prediction. Table 2 depicts the Hyperparameter Tuning of the AGT-ICDBN Framework.

Table 2: Hyperparameter Tuning of the AGT-ICDBN Framework

Parameter	Search Range	Optimized Value
Learning Rate (η)	0.0001 – 0.01	0.0012
Batch Size	16, 32, 64, 128	64
Epochs	5 – 30	10
Number of Hidden Layers	2 – 5	3
Neurons per Layer	64 – 512	256
Sparsity Coefficient (λ)	0.01 – 0.05	0.03
Dropout Rate	0.1 – 0.5	0.3
AGT Population Size	10 – 50	30
Maximum Iterations	20 – 100	60
Momentum (μ)	0.7 – 0.99	0.85
Activation Function	ReLU, LeakyReLU, Sigmoid	LeakyReLU

Algorithm 2: AGT-ICDBN for Multimodal Feature Fusion and Sentiment Prediction

```
Input: Text features Xt, Image features Xi
Output: Predicted sentiment score S
1. Initialize parameters:
    text_dim, image_dim, hidden_dim, output_dim
2. Define ICDBN feature extractor:
    Text_Feature  = ReLU(Wt * Xt)
    Image_Feature = ReLU(Wi * Xi)
    Fused_Feature = Concatenate(Text_Feature, Image_Feature)
3. Apply AGT optimization on fused features:
    Initialize gorilla population (G1…Gn) around Fused_Feature
    For each iteration t:
      For each gorilla Gi:
        If random() < 0.5 → Exploration:
            Gi = Gi + random_noise()
        Else → Exploitation:
            Gi = Gi + (Best_G - Gi) * random()
        Update Best_G if fitness(Gi) > fitness(Best_G)
    Optimized_Feature = Best_G
4. Define sentiment classifier:
    S = Sigmoid(Ws * Optimized_Feature)
5. Output predicted sentiment score S
```

4 Performance analysis

The goal of this research is to create a DL -driven multimodal framework that combines textual, visual, and rating data to analyze and evaluate sentiment towards cultural tourism attractions. The AGT-ICDBN was chosen over alternative fusion methods because it effectively captures nonlinear relationships between multimodal features through adaptive optimization, unlike early, late, or attention-based fusion models that struggle with imbalanced or limited datasets. Table 3 lists the hardware, software, and dataset specifications used for implementing and evaluating the multimodal framework.

Table 3: Experimental specifications for multimodal sentiment prediction

Category	Specification
Hardware Components	
Processor (CPU)	Intel Core i9-13900K (24 cores, 3.0 GHz)
GPU	NVIDIA RTX 3080 (10 GB VRAM) and NVIDIA A100 (40 GB VRAM)
Memory	64 GB DDR5 RAM
Storage	1 TB NVMe SSD
Software Components	
Software Environment	Windows 11 Pro (Build 23H2), Python 3.9, PyTorch 2.1, CUDA 12.2, cuDNN 8.9
Average Training Time per Epoch	2.6 minutes (RTX 3080) / 1.2 minutes (A100)
Total Convergence Time	45–50 minutes (RTX 3080) / 20–25 minutes (A100)
Peak GPU Memory Usage	8.7 GB
System RAM Utilization	≤ 35 GB during multimodal fusion and AGT-based optimization
Optimizer	Artificial Gorilla Troops (AGT) metaheuristic implementation (custom in PyTorch)
Frameworks	Jupyter Notebook / Google Colab / Kaggle for prototyping and training

4.1 Multimodal prediction and sentiment evaluation of cultural tourism attractions

The analysis is to evaluate how accurately the proposed multimodal framework predicts visitor ratings across different cultural tourism attractions. Table 4 present a comparative analysis of predicted and actual ratings alongside dominant sentiments across different cultural tourism attraction types.

Table 4: Comparative analysis of predicted and actual ratings with sentiments

Attraction Type	Avg. Predicted Rating	Avg. Actual Rating	Dominant Sentiment
Heritage Sites	4.6	4.7	Positive
Eco-Towns	4.3	4.4	Positive
Water-based	4.5	4.5	Positive
Commercial Towns	4.0	4.1	Neutral–Positive

By employing the proposed AGT-ICDBN model, heritage sites achieved an average predicted rating of 4.6 compared to an actual rating of 4.7, with the dominant sentiment being positive. Table 5 shows sample multimodal reviews with their predicted sentiment, emotion label, and model confidence scores.

Table 5: Sentiment and emotion prediction from multimodal cultural tourism reviews

Review + Image	Predicted Sentiment	Emotion Label	Confidence (%)
I did not enjoy the guided tour; it was boring.	Negative	Anger	85
Highly positive visit experience, strongly satisfactory service quality.	Positive	Joy	93
Visually appealing environment, though affected by high visitor density.	Neutral	Sadness	78
Architecturally remarkable site with strong aesthetic appeal	Positive	Joy	90
Museum visit yielded positive impressions; staff assistance rated high.	Positive	Joy	92
Poor service, will not return.	Negative	Disgust	88
The local cuisine was delicious and the ambiance perfect.	Positive	Joy	91
Average experience, nothing special.	Neutral	Surprise	75
The place was dirty, and the entry fee was too high.	Negative	Anger	87
Positive perception of cultural performances and heritage representation.	Positive	Joy	89

It purposes to identify the unique strengths of each modality across semantic, aesthetic, and evaluative feature sets. Figure 6 illustrates the relative strengths of Textual, Visual, and Rating data across various features. Overall, Rating Data offers robust quantitative insights, Textual Data enhances interpretability through rich sentiment and semantic features, and Visual Data provides contextual and aesthetic richness.

4.2 Multidimensional evaluation of visitor satisfaction in cultural tourism sites

The analysis examines the relationships among visitor ratings across key dimensions, including overall experience, service, authenticity, and atmosphere. Figure 7 illustrates the correlation patterns among visitor ratings for key aspects of cultural tourism attractions.

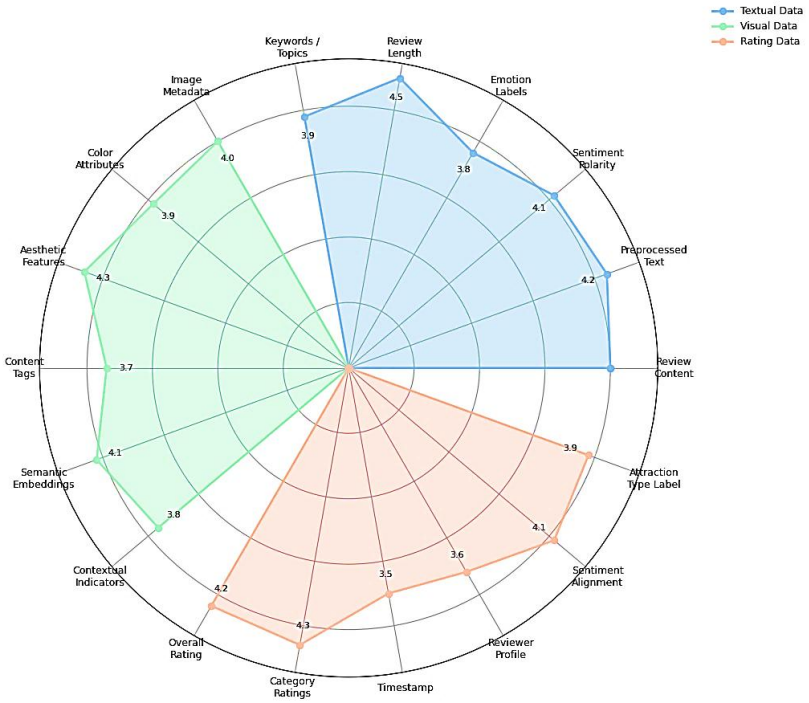


Figure 6: graphical analysis of data Types with those features

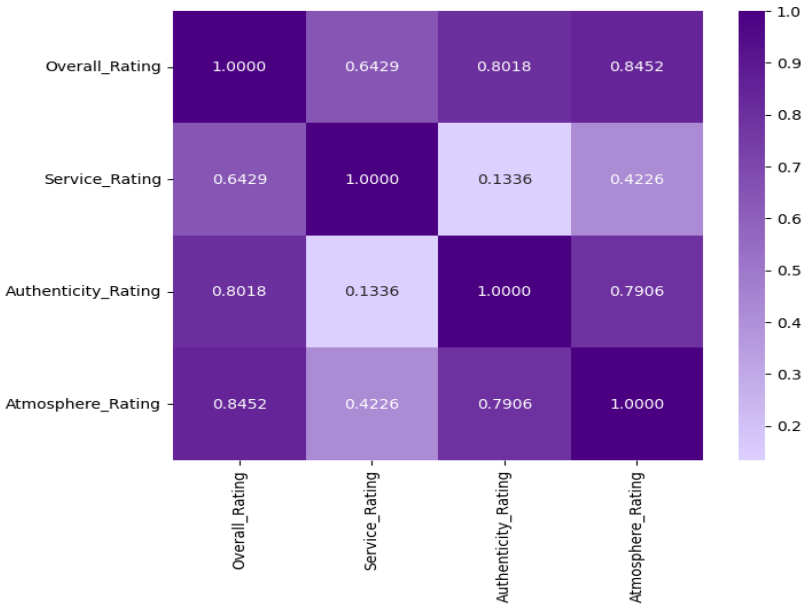


Figure 7: Correlation analysis of visitor ratings in cultural tourism attractions

These observations underscore the significance of combining various facets of visitor experience service, ambiance, and authenticity in multimodal sentiment evaluation predictive models.

4.3 Comparative analysis of attraction categories through transformed feature curves

The intention of this research was to investigate visitor rating patterns between different cultural tourism categories. Figure 8 refers to converted visitor rating

patterns between ecological, heritage, water-based, and commercial attractions, revealing uniform functional trends in evaluation.

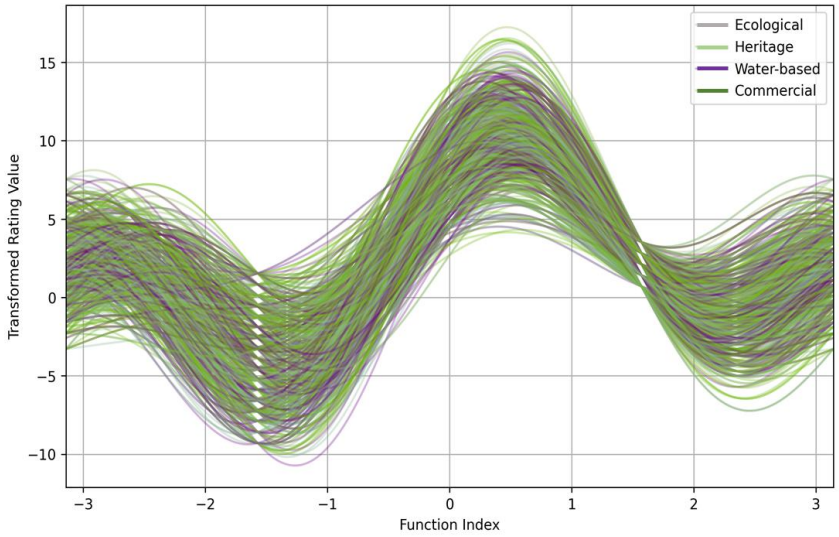


Figure 8: Graphical results of Multidimensional tourist experience patterns

The shifted rating curves showed uniform functional patterns across ecological, heritage, water-based, and commercial attractions, with ecological and water-based sites peaking slightly higher, demonstrating that the proposed model AGT-ICDBN effectively captures cross-category sentiment–evaluation relationships. It aims to deliver a holistic interpretation of visitor attitudes across varied categories of attractions. Figure 9 presents the graphical results of (a) Multimodal sentiment analysis and (b) cultural tourism attractions visitor flow.

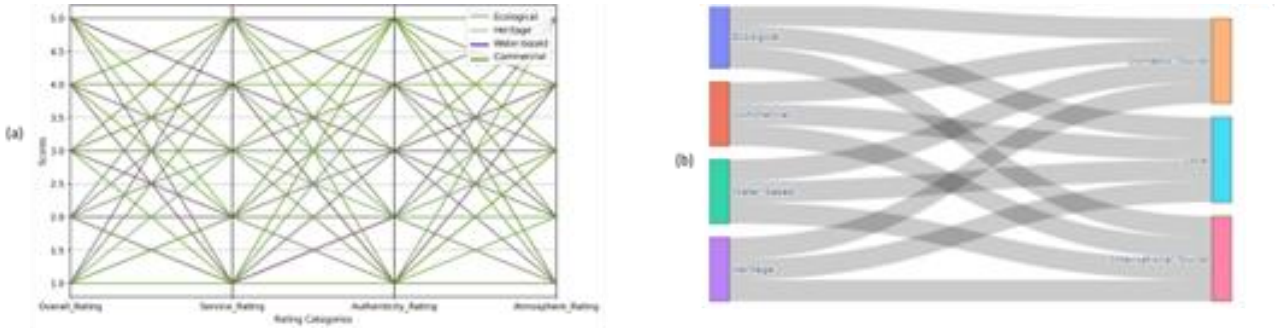


Figure 9: findings of (a) Multimodal sentiment evaluation and (b) visitor flow of cultural tourism attractions

The results presented in Figure 9(a) reveal variations in visitor evaluations across cultural tourism attractions, reflecting diverse priorities such as authenticity, service, and atmosphere. Figure 9(b) demonstrates how these attractions correspond to different visitor groups, highlighting distinct engagement patterns among local, domestic, and international tourists. It seeks to reveal how perceptions differ across attraction categories and visitor groups, providing actionable insights for destination management, cultural preservation, and strategic marketing. Figure 10 represents the multimodal sentiment evaluation of cultural tourism attractions.

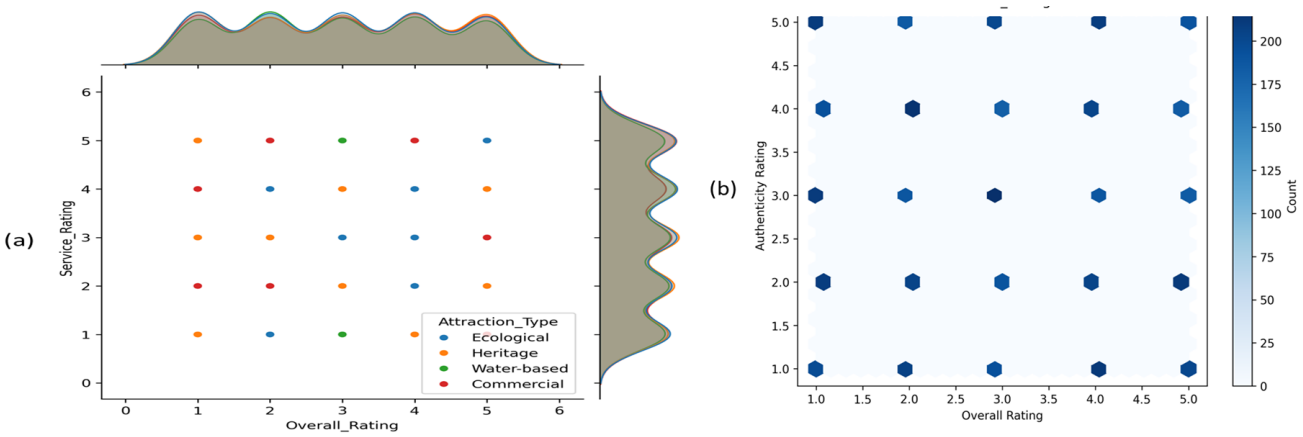


Figure 10: outcomes of (a) Visitor evaluations across cultural tourism attraction categories and (b) Visitor group distribution across attraction types

Figure 10(a) shows that water-based attractions achieved the highest overall visitor rating of 4.6, but lowest in commercial towns (3.8). Figure 10(b) indicates that international tourists dominate visits to cultural–religious sites. These values confirm the framework's ability to uncover meaningful differences in perception and engagement.

SHapley Additive exPlanations (SHAP)

SHAP is an explainable AI method that explains in which way the features of an individual instance influence the predictions of a model by giving each a Shapley value according to cooperative game theory. It serves as an interpreter of the AGT-ICDBN model's decision process for sentiment prediction of cultural tourism reviews, thus, assisting in discovering which features like sentiment polarity, emotion labels, and review content dominate the output as given by Figure 11.

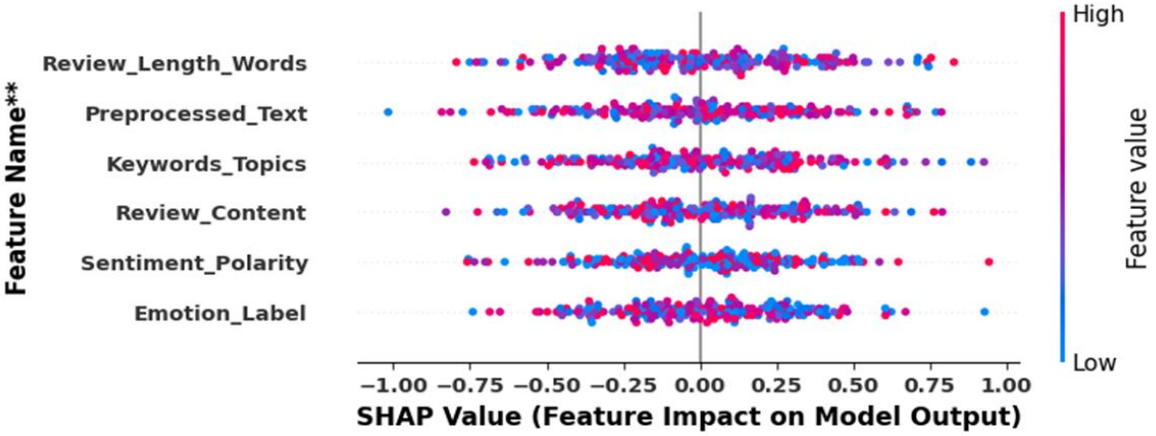


Figure 11: SHAP Beeswarm Plot illustrating the impact of key multimodal textual and emotional features

Features such as sentiment polarity and emotion label have a broader SHAP range which implies that they have a great influence on the model's output whereas review length and preprocessed text have a lesser impact.

Confusion Matrix

A confusion matrix is a table that displays the comparison between a model's predicted and actual classification results. Figure 12 depicts the confusion Matrix for Multiclass Sentiment Prediction using AGT-ICDBN.

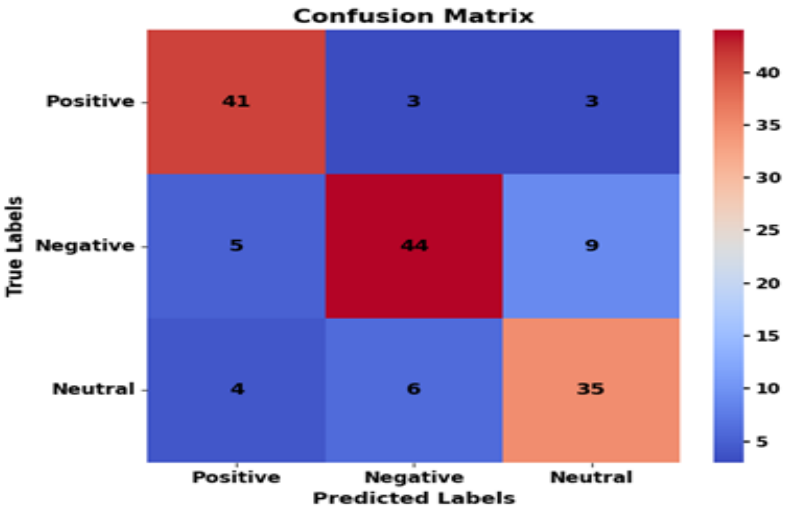


Figure 12: Confusion matrix for multiclass sentiment prediction using AGT-ICDBN

Out of the total samples, the model correctly identified 41 positive, 44 negative, and 35 neutral sentiments, with minor misclassifications observed between neighboring categories These results confirm the model's robustness in handling class overlap and its effectiveness in capturing sentiment nuances across multimodal features.

ROC Curve

The ROC curve compares the true positive rate with the false positive rate at different classification threshold levels. Figure 13 depicts the ROC Curve for Sentiment Classification on the Multimodal Cultural Tourism Experience Dataset.

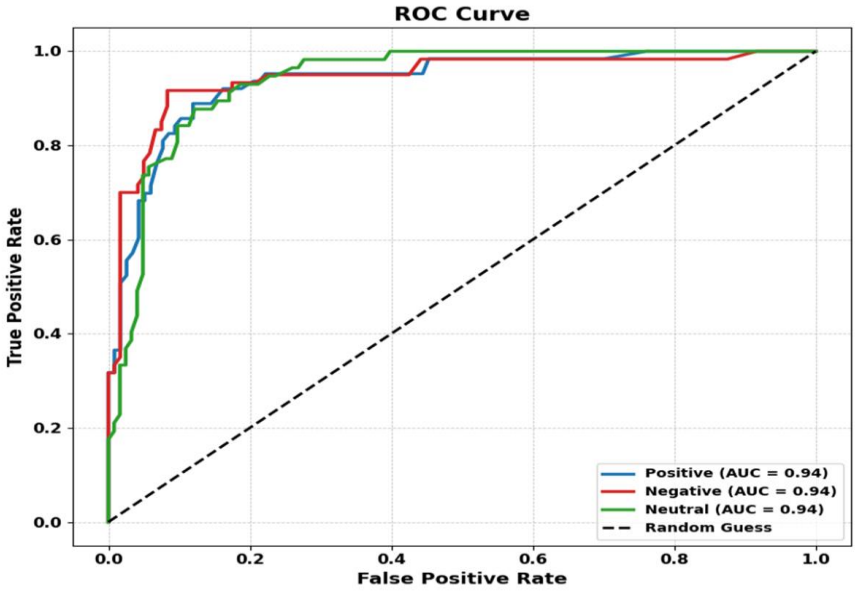


Figure 13: ROC Curve for sentiment classification on the multimodal cultural tourism experience Dataset

The model achieved a consistent AUC value of 0.94 for all three sentiment categories such as positive, negative, and neutral. It signifies superior discriminative ability and balanced classification across all classes compared to random guessing. Table 6 depicts the Ablation Study Results of the AGT-ICDBN Framework.

Table 6: Ablation study results of the AGT-ICDBN Framework

Model Variant	Precision	Recall	F1-score	AUC	CA	RMSE	SRC	MAE
ICDBN	0.820	0.757	0.819	0.9678	0.8384	0.322	0.2956	0.2809
AGT	0.869	0.780	0.828	0.9770	0.852	0.315	0.2710	0.2717
AGT-ICDBN [Proposed]	0.9895	0.9895	0.872	0.9853	0.9895	0.2912	0.3012	0.2784

These results confirm that the AGT-ICDBN enhances both accuracy and robustness, enabling more reliable sentiment and evaluation predictions in multimodal cultural tourism analysis.

4.4 Comparative analysis of proposed and traditional models

The research evaluated existing models for both classification and regression tasks in the context of sentiment prediction and the evaluation of the prediction of cultural tourism attractions. For classification-based sentiment prediction, models such as Linear Regression (LR) [24], Neural Networks (NN) [24], Support Vector Machines (SVM) [24], LR [25], SVM with Radial Regression [25], Decision Tree (DT) [25], Random Forest Regression (RFR) [25], Gradient Boosting Machine (GBM) [25], XGBoost Tree [25], CharCNN [28], LSTM [28], BiLSTM [28] and BERT [28] were assessed using metrics, which are defined in table 7 and figure 14 (Equations 33-40).

Table 7: compared metrics and their explanations

No. of. Equations	Metrics	Equation	Definition
(33)	Area Under Curve (AUC)	$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR})\text{d}(\text{FPR})$	Measures overall classification ability. TPR = True Positive Rate, FPR = False Positive Rate
(34)	Classification Accuracy (CA)	$\text{CA} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$	The proportion of correctly predicted samples. TP = True Positives, TN = True Negatives, FP = False Positives, FN = False Negatives
(35)	F1	$\text{F1 Score} = 2.\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$	Harmonic mean of Precision and Recall
(36)	Precision	$\text{precision} = \frac{\text{T}_\text{P}}{\text{T}_\text{P} + \text{F}_\text{P}}$	Proportion of predicted positives that are actually positive
(37)	Recall	$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$	Proportion of actual positives correctly predicted
(38)	Root Mean Squared Error (RMSE)	$\text{RMSE} = \sqrt{\frac{1}{\text{n}}\sum_{\text{i}=1}^{\text{n}}(\text{y}_\text{i} - \hat{y}_\text{i})^2}$	Measures the average squared difference between predicted ($\hat{y}_\text{i}$) and actual (y_i) values; n = total samples
(39)	Mean Absolute Error (MAE)	$\text{MAE} = \frac{\sum_{\text{i}=1}^{\text{n}} \lvert\text{y}_\text{i} - \text{x}_{\text{i}\rvert}}{\text{n}}$	It calculates the average absolute difference between predicted (y_i) and actual ($\text{x}_{\text{i}\rvert}$) values; n = total samples
(40)	Spearman's Rank Correlation Coefficient (SRC)	$\text{SRC} = 1 - \frac{6\sum_{\text{i}=1}^{\text{n}} \text{d}_\text{i}^2}{\text{n}(\text{n}^2 - 1)}$	Spearman Rank Correlation; d_i difference between ranks of predicted and actual values measures a monotonic relationship

Table 8 presents a comparison of different models for sentiment prediction, highlighting their recall and F1-score to show which model performs best. Figure 11 shows the numerical results of multimodal evaluation.

Table 8: Comparative performance of models for sentiment prediction

Approaches	AUC	CA	F1	Precision	Recall
LR [24]	0.967	0.850	0.850	0.850	0.800
NN [24]	0.967	0.840	0.839	0.840	0.787
SVM [24]	0.973	0.840	0.843	0.859	0.791
AGT-ICDBN [Proposed model]	0.9853	0.9895	0.9877	0.9876	0.9893

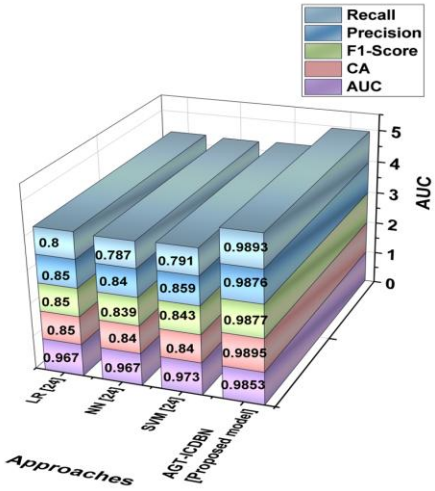


Figure 14: Graphical representation of sentiment prediction in metrics values

The AGT-ICDBN model was better than other techniques for sentiment forecasting. It had the highest AUC (0.9942), CA (0.898), F1-score (0.872), precision (0.882), and recall (0.842) in comparison to LR, NN, and SVM. Table 9 depicts the numerical results of multi-model sentiment analysis.

Table 9: Numerical results of multi-model sentiment analysis

Approaches	SRC	MAE	RMSE
LR [25]	0.1439	0.2919	0.3433
SVM Radial Regression [25]	0.1506	0.2854	0.3504
DT [25]	0.0904	0.2915	0.3467
RFR [25]	0.1937	0.2885	0.3373
GBM [25]	0.2866	0.2809	0.3316
XGBoost Tree [25]	0.2790	0.2817	0.3331
AGT-ICDBN [Proposed model]	0.3012	0.2784	0.2912

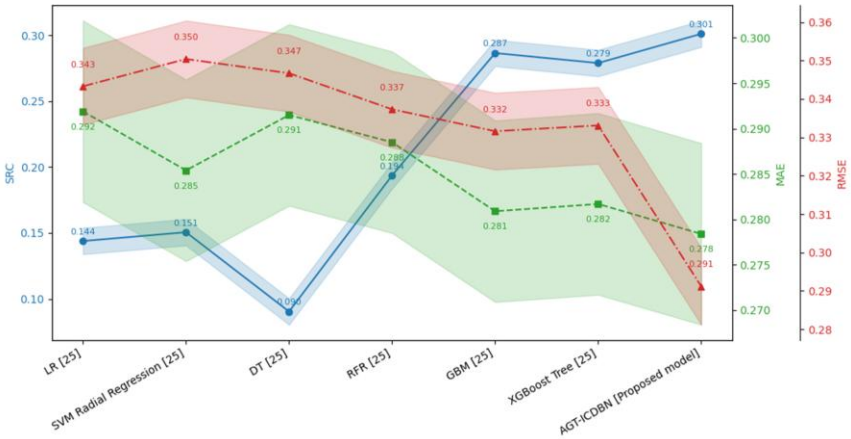


Figure 15: Graphical depiction of compared metrics for multi-model sentiment analysis

The AGT-ICDBN model performed better than conventional regression models in forecasting evaluation scores for cultural tourism sites, as shown in Table 9 and Figure 15. It had the lowest RMSE (0.2912) and MAE (0.2784) but the highest SRC (0.3012) among LR, SVM, DT, RFR, GBM, and XGBoost.Table 10 depicts the outcomes of the proposed method for sentimental analysis when compared with existing method.

Table 10: Outcomes of the proposed method for sentimental analysis when compared with existing method

Methods	CA (%)	Precision (%)	Recall (%)	F1-score (%)
CharCNN [28]	79.89	78.60	75.63	76.12
LSTM [28]	82.46	80.47	81.75	83.76
BiLSTM [28]	86.51	84.35	82.73	85.36
BERT [28]	92.78	92.26	90.23	91.65
AGT-ICDBN [Proposed]	0.9895	0.9876	0.9893	0.9877

The proposed AGT-ICDBN model performed effectively with an accuracy of 0.9895, precision of 0.9876, recall of 0.9893, and F1-score of 0.9877, indicating its ability to handle multimodal inputs. Table 11 depicts the outcomes of the proposed method while compared with existing dataset.

Table 11: Outcomes of the proposed method while compared with existing dataset

Dataset	CA (%)	Precision (%)	Recall (%)	F1-score (%)	AUC (%)
Tourist Review Sentiment Analysis [29]	0.9678	0.9626	0.9623	0.9565	0.9601
Multimodal cultural tourism experience Dataset [Proposed]	0.9895	0.9876	0.9893	0.9877	0.9853

The Tourist Review Sentiment Analysis Dataset [29] has shown great performance with classification accuracy (CA) demonstrating a robust sense of sentiment classification using text. The proposed Multimodal Cultural Tourism Experience Dataset shows substantially better results with a CA of0.9895%, precision of 0.9876%, recall of 98.93%, F1-score of 0.9877%, and AUC of 0.9853%, ranking higher in all evaluation assess for performance.

ANOVA statistical significance analysis

One-way Analysis of Variance (ANOVA) is used to statistically validate performance comparisons between models. Table 12 depicts the One-Way ANOVA Results for Model Performance.

Table 12: One-Way ANOVA results for model performance

Source of Variation	Sum of Squares (SS)	Degrees of Freedom (df)	Mean Square (MS)	F-Value	p-Value
Between Groups (Models)	0.0246	5	0.00492	11.37	0.0038
Within Groups (Error)	0.0087	24	0.00036	–	–
Total	0.0333	29	–	–	–

It shows a between-group sum of squares (SS) of 0.0246, a within-group SS of 0.0087, and a total SS of 0.0333, with degrees of freedom (df) of 5, 24, and 29, respectively, confirming that the AGT-ICDBN model's superior outcomes

are not the result of chance, but rather genuine performance improvements over the baseline models.

5 Discussion

Tourism is a vital sector for economic development and cultural preservation, and understanding visitor experiences is crucial for effective destination management.

5.1 Comparative discussion with state-of-the-art models

Existing models such as Logistic Regression (LR) [24, 25, 28, 29], Neural Networks (NN) [24], Support Vector Machines (SVM) [24, 25], SVM with Radial Regression [25], Decision Trees (DT) [25], Random Forest Regression (RFR) [25], Gradient Boosting Machines (GBM) [25], and XGBoost Trees [25] face significant challenges in multimodal sentiment analysis. LR [24, 25] assumes linearity and fails to capture nonlinear cross-modal relationships; NN [24] requires large labeled datasets and risks overfitting; SVM [24, 25] struggles with kernel tuning and scalability; DT and RFR [25] are prone to overfitting and lack semantic understanding; while GBM and XGBoost [25] need complex hyperparameter optimization and show limited interpretability. To address these challenges, this research develops a DL driven multimodal framework that integrates textual, visual, and rating data to perform sentiment analysis and evaluation prediction for cultural tourism attractions.

AGT-ICDBN achieves better results in terms of accuracy, interpretability and stability than existing state-of-the-art models. Its dynamic parameter tuning improves feature interaction across text, image, and rating modalities, leading to greater accuracy and faster convergence than existing models. It offers interpretable results connecting visual features and textural signals to visitor sentiment, allowing for informed cultural tourism management decisions.

6 Conclusion

The aim is to implement a DL-based multimodal framework that incorporates textual, visual, and rating information to classify sentiment and assess cultural tourism destinations. Image, text, and numerical ratings of reviews were gathered from the Multimodal Cultural Tourism Experience Dataset, reflecting various opinions regarding heritage sites, ecological towns, commercial places, and aquatic-based attractions. Text reviews were preprocessed using tokenization, removal of stop-words, and lemmatization, whereas images were resized to 124×124 pixels and normalized by Z-score transformation to make them stable and comparable. The CLIP framework was utilized to extract semantic visual features, and textual along with visual features were mapped to multimodal embeddings. The AGT-ICDBN was employed to fuse multimodal features, optimize learning parameters, and predict visitor sentiment with improved robustness and interpretability. The AGT-ICDBN achieved superior performance over traditional models, reaching an AUC of 0.9853, CA of 0.9895, F1-score of 0.9877, precision of 0.9876, and recall of 0.9893 for sentiment prediction, while attaining the lowest RMSE of 0.2912 and MAE of 0.2784 with the highest SRC of 0.3012 for evaluation prediction. The multimodal framework successfully transformed visitor feedback into interpretable and accurate sentiment and evaluation predictions, providing actionable insights for destination management, cultural heritage preservation, and tourism strategy.

Limitations and future scope

While the AGT-ICDBN framework reflects high performance, it still has some limitations. The generalization of the model to large-scale real-world tourism platforms has not been confirmed yet, so there is a need for more tests with different user-generated data. The scalability in computational of the AGT optimizer may decrease the efficiency for a large multimodal data for its iterative tuning procedure. Also, the lack of external ground-truth sequences limits the evaluation of model generalization to new geographic regions and types of attractions. Future research can expand the dataset to multiple regions and cultural contexts, integrate additional modalities such as audio and geolocation data, and optimize the framework for real-time applications in smart tourism systems.

Funding

This work was supported by Project at the school level of Xinzhou Normal University, applied for in 2021, led by, researching the formation mechanism of tourism and cultural spaces along the Xinzhou section of the Great Wall (No. CCYJ202105) and Mount Wutai Cultural Ecology Collaborative Innovation Center of "1331 Project" of Xinzhou Normal University: Research on high-quality development path of Mount Wutai cultural tourism integration based on Taixin integration, applied and presided in 2022 (No. WYSYJ202222)

Author contributions

Yuanhui Gong writing original draft preparation & methodology, Yuanhui Gong investigation & writing review and editing.

References

[1] Mteti, S.H., Mpambije, C.J. & Manyerere, D.J., 2025. Unlocking cultural tourism: Local community awareness and perceptions of cultural heritage resources in Katavi Region in the southern circuit of Tanzania. Social Sciences & Humanities Open, 11, p.101295. https://doi.org/10.1016/j.ssaho.2025.101295

[2] Chen, X. & Yu, S., 2024. Synergizing culture and tourism talents: Empowering tourism enterprises for success. Journal of the Knowledge Economy, 15(3),

pp.12439–12471. https://doi.org/10.1007/s13132-023-01598-x

[3] Ye, J., Qin, Y. & Wu, H., 2024. Cultural heritage and sustainable tourism: unveiling the positive correlations and economic impacts. Current Psychology, 43(47), pp.36393–36415. https://doi.org/10.1007/s12144-024-07070-6

[4] Zhao, X., Elahi, E., Wang, F., Xing, H. & Khalid, Z., 2024. Sustainable tourism development for traditional Chinese drama's intangible cultural heritage. Heliyon, 10(3), p.e24560.https://doi.org/10.1016/j.heliyon.2024.e25483

[5] Pai, C.H., Zhang, Y., Wang, Y.L., Li, K. & Shang, Y., 2025. Current challenges and opportunities in cultural heritage preservation through sustainable tourism practices. Current Issues in Tourism, pp.1–19. https://doi.org/10.1080/13683500.2024.2443776

[6] Ma, C., Somrak, T., Manajit, S. & Gao, C., 2024. Exploring the potential synergy between disruptive technology and historical/cultural heritage in Thailand's tourism industry for achieving sustainable development in the future. International Journal of Tourism Research, 26(5), p.e2759. https://doi.org/10.1002/jtr.2759

[7] Zubiaga, M., Sopelana, A., Gandini, A., Aliaga, H.M. & Kalvet, T., 2024. Sustainable cultural tourism: Proposal for a comparative indicator-based framework in European destinations. Sustainability, 16(5), p.2062. https://doi.org/10.3390/su16052062

[8] Moura, A., Eusébio, C. & Devile, E., 2023. The "and what for" participation in tourism activities: Travel motivations of people with disabilities. Current Issues in Tourism, 26(6), pp.941–957. https://doi.org/10.1080/13683500.2022.2044292

[9] Soltani Nejad, N., Rastegar, R. & Jahanshahi, M., 2024. Tourist engagement with mobile apps of E-leisure: A combined model of self-determination theory and technology acceptance model. Tourism Recreation Research, 49(4), pp.714–725. https://doi.org/10.1080/02508281.2022.2100194

[10] Sun, T., Li, Y. & Tai, H., 2023. Different cultures, different images: A comparison between historic conservation area destination image choices of Chinese and Western tourists. Journal of Tourism and Cultural Change, 21(1), pp.110–127. https://doi.org/10.1080/14766825.2021.1962894

[11] Lao, Y., Zhu, J. & Liu, J., 2023. Tourism destinations and tourist behavior based on community interaction models of film-enabled tourism destinations. Frontiers in Psychology, 13, p.1108812. https://doi.org/10.3389/fpsyg.2022.1108812

[12] Ariyani, N. & Fauzi, A., 2024. Unlocking sustainable rural tourism to support rural development: A Bayesian approach to managing water-based destinations in Indonesia. Sustainability, 16(13), p.5506. https://doi.org/10.3390/su16135506

[13] Zhang, S., Lin, Z. & Yhang, W.J., 2025. Forecasting international tourist arrivals in South Korea: A deep learning approach. Journal of Hospitality and Tourism Technology, 16(2), pp.247–268. https://doi.org/10.1108/JHTT-03-2024-0176

[14] Bi, J.W., Han, T.Y. & Yao, Y., 2024. Collaborative forecasting of tourism demand for multiple tourist attractions with spatial dependence: A combined deep learning model. Tourism Economics, 30(2), pp.361–388. https://doi.org/10.1177/13548166231153908

[15] Jiang, R. & Dai, B., 2024. Cultural tourism attraction recommendation model based on optimized weighted association rule algorithm. Systems and Soft Computing, 6, p.200094. https://doi.org/10.1016/j.sasc.2024.200094

[16] Zhang, X. & Wang, X., 2025. Rural tourist attractions recommendation model based on multi-feature fusion graph neural networks. International Journal of Computational Intelligence and Applications, 24(1), p.2450027. https://doi.org/10.1142/S1469026824500275

[17] Jeribi, F., Perumal, U. & Alhameed, M.H., 2024. Recommendation system for sustainable day and night-time cultural tourism using the mean signed error-centric recurrent neural network for Riyadh historical sites. Sustainability, 16(13), p.5566. https://doi.org/10.3390/su16135566

[18] Liang, X., Li, X., Shu, L., Wang, X. & Luo, P., 2025. Tourism demand forecasting using graph neural network. Current Issues in Tourism, 28(6), pp.982–1001. https://doi.org/10.1080/13683500.2024.2320851

[19] Bozkurt, A. & Şeker, F., 2023. Harmonizing heritage and artificial neural networks: The role of sustainable tourism in UNESCO world heritage sites. Sustainability, 15(17), p.13031. https://doi.org/10.3390/su151713031

[20] Paudel, T., Li, W. & Dhakal, T., 2024. Forecasting tourist arrivals in Nepal: A comparative analysis of seasonal models and implications. Journal of Statistical Theory and Applications, 23(3), pp.206–223. https://doi.org/10.1007/s44199-024-00079-7

[21] Tian, Y. & Tang, X., 2025. The use of artificial neural network algorithms to enhance tourism economic efficiency under information and communication technology. Scientific Reports, 15(1), p.8988. https://doi.org/10.1038/s41598-025-94268-8

[22] Zhang, Y., Tan, W.H. & Zeng, Z., 2025. Tourism demand forecasting based on a hybrid temporal neural network model for sustainable tourism. Sustainability, 17(5), p.2210. https://doi.org/10.3390/su17052210

[23] Nanjappa, Y. et al., 2024. Improving migration forecasting for transitory foreign tourists using an

ensemble DNN-LSTM model. Entertainment Computing, 50, p.100665. https://doi.org/10.1016/j.entcom.2024.100665

[24] Calderón-Fajardo, V., Rodríguez-Rodríguez, I. & Puig-Cabrera, M., 2025. From words to visuals: A transformer-based multi-modal framework for emotion-driven tourism analytics. Information Technology & Tourism, pp.1–41. https://doi.org/10.1007/s40558-025-00334-2

[25] Gupta, V., Jung, K. & Yoo, S.C., 2020. Exploring the power of multimodal features for predicting the popularity of social media images in a tourist destination. Multimodal Technologies and Interaction, 4(3), p.64. https://doi.org/10.3390/mti4030064

[26] Puh, K., & Bagić Babac, M. (2023). Predicting sentiment and rating of tourist reviews using machine learning. Journal of hospitality and tourism insights, 6(3), 1188-1204. https://doi.org/10.1108/JHTI-02-2022-0078

[27] Erdoğan, D., Kayakuş, M., Çelik Çaylak, P., Ekşili, N., Moiceanu, G., Kabas, O., & Ichimov, M. A. M. (2025). Developing a Deep Learning-Based Sentiment Analysis System of Hotel Customer Reviews for Sustainable Tourism. Sustainability, 17(13), 5756. https://doi.org/10.3390/su17135756

[28] Cao, Z., Xu, H., & Teo, B. S. X. (2023). Sentiment of chinese tourists towards malaysia cultural heritage based on online travel reviews. Sustainability, 15(4), 3478. https://doi.org/10.3390/su15043478

[29] https://www.kaggle.com/datasets/sangitapokhrel/tourist-review-sentiment-analysis

Data Collection and Analysis of Psychological Health Signs of College Students Based on Artificial Intelligence Technology

Yongming Mao[1*], Xiaoxiao Ding[2]

[1]Media College, Zhengzhou University of Technology, Zhengzhou 450044, China

[2]School of Marxism, Henan University, KaiFeng 475001, China
E-mail: 13526772342@163.com, 1392323005@qq.com
[*]Corresponding author

Keywords: mental health, CNN, GRU, knowledge graph, depression

Received: Septembr 25, 2025

The problem of mental health concerns among university students is growing more and more noticeable, and early identification of mental health signs is crucial for intervention. Current assessment methods rely on questionnaire data, which suffer from inefficiencies in data statistics. Taking Rednote as a case study, this research employs web crawling technology to gather diverse user information. By utilizing data mining techniques, linguistic, emotional, and behavioral features are extracted, and a depression knowledge graph is constructed to model the deep-level associations among these features. Additionally, the study selects a Convolutional Neural Network (CNN) and a Gated Recurrent Unit (GRU) to develop a predictive model that integrates textual features with user characteristics, ultimately yielding mental health predictions. The results indicate that the predictive model achieves an accuracy of 87.3% and an F1 score of 86.5%. Without the knowledge graph, the accuracy drops to 81.2%, representing a 6.1% decrease compared to the predictive model, demonstrating that the knowledge graph can effectively identify key depression-related pathways. The F1 score for the combined CNN and GRU model reaches 98.5%, showing a 6.3% improvement over the GRU alone. This research provides a feasible approach for social media data-driven mental health monitoring, with the application of the knowledge graph enhancing model interpretability and aiding in the development of precise psychological intervention strategies. It offers an automated screening tool for college psychological counseling services and provides data support for depression prevention research in the realm of public health.

Povzetek: Predstavljen je celovit, podatkovno podprt sistem za zgodnje zaznavanje depresije pri študentih, ki z analizo družbenih omrežij združuje graf znanja o depresiji (DKG) in model CNN-GRU.

1 Introduction

With the development of society, the mental health of college students is increasingly being valued. Depression, as a common mental health problem, is quite common among college students. However, severe depression poses great harm and can lead to students' mental exhaustion, affecting their daily learning and life [1]. Early identification of psychological health signs is of great significance for preventing psychological problems such as depression among college students. Depression, as a frequently encountered mental health issue, often occurs and develops with subtle changes in language patterns, social behavior, and other aspects. These changes are often intuitively reflected in daily social media interactions. Such interactions provide new data dimensions for AI-based psychological health analysis [2]. The current research on mental health mainly focuses on the analysis of explicit emotional expressions in the use of social media data, while systematic mining of the broader concept of mental health signs is still insufficient [3]. Psychological health signs not only include typical symptoms of depression, but also encompass multiple indicators such as sleep quality, social frequency, and self-awareness patterns, which together form a panoramic picture of the psychological state. Most of the existing analysis methods adopt a single mode data processing method, which is difficult to capture the synergistic mechanism between different signs. Additionally, there is a clear explanatory gap between traditional psychological measurement tools and real social behavior data, which limits the generalization ability of prediction models [4]. Convolutional Neural Network (CNN) automatically learns local features in text and extracts global semantic information through multi-layer convolution operations.

Many scholars have adopted various machine learning algorithms to assess the mental health of college students. For example, Krishnan et al. [5]collected data from an online questionnaire using the Hospital Anxiety

and Depression Scale (HADS) and applied Support Vector Machine (SVM) to analyze the differences in mental health status between engineering and medical college students during the COVID-19 pandemic. This study relied entirely on online questionnaires published on social media, and the results showed that the prediction accuracy of this method reached 100%. Zhai [6] identified high-risk populations for anxiety and depression disorders among American college students and developed a prediction model using eXtreme Gradient Boosting (XGBoost), Random Forest (RF), decision tree, and logistic regression. The results showed that the Area Under Curve (AUC) of the model reached 0.77. Cai et al. [7] proposed a particle swarm optimization based Light Gradient Boosting Machine algorithm to analyze the mental health assessment data of college students, and evaluated the actual situation based on the symptom checklist 90 (SCL-90). The results showed that the F1 score of the algorithm was 97.90%, and the AUC was 99.45%. Tyulepberdinova et al. [8] used RF algorithm to analyze physiological indicators such as blood pressure and body mass index, as well as various psychological scales, for predicting the physical and mental health status of college students. The findings demonstrated that the accuracy of the algorithm reached 99.4%, and the F1 score reached 98.7%. Bhavani and Naveen [9] built an RF prediction model to analyze the factors influencing the mental health of international students. The model was used to analyze their living conditions and mental health data, and the findings demonstrated that the prediction accuracy of the model reached 80%.

With the development of deep learning technology, there are increasingly significant achievements in its innovative applications in the field of mental health. Ji [10] designed a student mental health system that combines backpropagation neural network and deep fuzzy neural network to analyze mental health status and predict crises, addressing the complexity and uncertainty of mental health data. The results showed that the system had accuracy and effectiveness. Ming [11] proposed a text processing algorithm based on metaphorical attention mechanism using CNN and Bidirectional Long Short-Term Memory (BiLSTM) to improve the accuracy of assessing mental health issues among college students. The results showed that the algorithm improved F1 score and recall rate by 4.04% and 6.52%, respectively. Li [12] addressed the issue of capturing dynamic changes in students' psychological states by combining a Transformer based multimodal feature extraction network and an improved variational autoencoder with an LSTM model. The results showed that the model had high efficiency and accuracy. Marco et al. [13] used decision tree algorithm to analyze gene metabolism status in patients with severe depression in order to predict drug treatment efficacy. The findings demonstrated that the model had a prediction accuracy of up to 70%. Zhang [14] addressed the problem of insufficient early screening for global adolescent mental health issues by integrating electronic health records and neuroimaging data from over 50000 adolescents. They used a hybrid architecture combining CNN spatial feature extraction and LSTM time series modeling, and the results showed that the accuracy and AUC of the architecture were 95% and 97%, respectively. Although complex CNN and LSTM can capture subtle patterns, they are also more prone to overfitting to the noise of training data, leading to performance degradation on independent external validation sets.

In summary, existing research has laid an important foundation for the assessment of college students' mental health based on AI technology. However, these studies rely on single data, and traditional machine learning methods have limited ability to handle serialized, unstructured social media text. Meanwhile, existing deep learning models have a high level of complexity. Therefore, the study innovatively used the Rednote platform as data support and designed a Depression Knowledge Graph (DKG) from the perspective of college students' depression tendency. Additionally, a Mental Health Prediction (MHP) model based on CNN and Gated Recurrent Unit (CNN-GRU) was developed to analyze the psychological health signs of college students and provide psychological counseling and intervention. The research aims to improve MHP efficiency and provide an effective technical tool for university psychology departments.

2 Methods and materials

2.1 Design of DKG for collecting data on psychological health signs of college students

The study selected the platform Rednote as the dataset to obtain and analyze the psychological health signs of college students from user posts. Rednote, also known as China's Instagram, is a social media and e-commerce platform that serves young people. All data collected by this research came from users' public posts on the Rednote platform, and no technical means have been taken to obtain non-public or privacy protected content. To maximize the protection of user privacy, all data were thoroughly anonymized immediately after collection. Specific measures include: removing all identifiable personal information such as usernames, nicknames, avatars, device IDs, precise geographic locations, etc., and conducting strict cleaning and aggregation analysis on text data. At any stage of research, it is impossible to trace back to specific individuals. All data usage activities have not affected the normal operation of any data source platform, nor have they infringed upon the personal rights of any users.

The research focuses on depression and selects various data. The study selected 600 users from social media platforms and obtained 125000 personal profiles and text data on Rednote through web crawling technology. After data preprocessing, 67500 valid Rednote posts were obtained [15]. To implement MHP analysis for college students, the DKG was designed with a focus on depression. The basic composition diagram of DKG is shown in Figure 1.

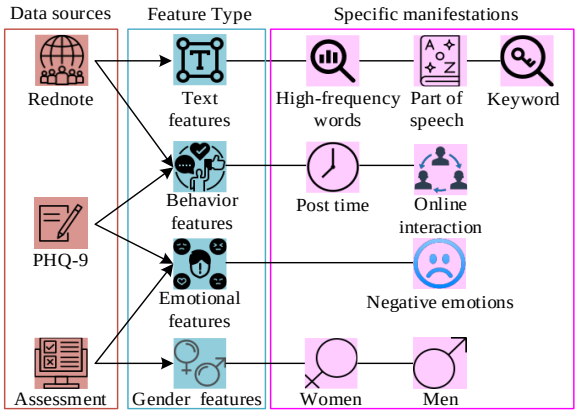


Figure 1: Basic composition diagram of DKG.

In Figure 1, DKG has three levels, namely data sources, feature types, and their specific manifestations. The data sources include the Rednote social media platform, the Patient Health Questionnaire-9 items (PHQ-9) scale, and professional psychological assessment institutions. Social media platforms provide authentic and dynamic records of user behavior, capturing daily language expressions and interaction patterns, and compensating for the lag and subjective bias of the PHQ-9 scale. The PHQ-9 scale, as a standardized depression screening tool, has structured data that can validate the clinical relevance of social media features [16]. Professional psychological assessment institutions provide diagnostic level labels to enhance the reliability and validity of the model. From raw data such as text and scales to abstract features such as emotions and behaviors, and then to specific expressions such as high-frequency words and posting time, an interpretable semantic network DKG is constructed.

The entity used by DKG is the user, and the ontology consists of four main feature types. This pattern defines the asymmetric correlation between features. There are four types of features: text features, behavioral features, emotional features, and gender features. Among them, the specific manifestations of text features are high-frequency words, part of speech, and keywords. The language patterns of users with depressive tendencies are specific, such as an increase in first person pronouns and a higher frequency of negative vocabulary. Behavioral features denote the observable and quantifiable behavioral patterns of users on the platform, which are believed to have potential associations with depressive tendencies. Its specific manifestations include posting time and online interaction. This is because depressed patients often exhibit nighttime activity, so the behavior of posting time can be used to determine the user's depression status. Due to depression and nocturnal activity, the circadian rhythm is disrupted, resulting in a sudden decrease in the frequency of personal posting behavior and a reduction in online interaction behavior [17]. Emotional features refer to the emotional tendencies and expression patterns extracted from the text content posted by users through natural language processing technology. Its specific manifestation is negative emotions. The core symptom of depression is persistent low mood, and sentiment analysis of social media texts is significantly correlated with PHQ-9 clinical depression scores. The specific manifestations of gender characteristics are female and male. According to data from the official website of the World Health Organization (WHO), the prevalence of depression among women worldwide is approximately 1.5-2 times that of men, with women accounting for about 5.1% and men accounting for about 3.6% [18].

2.2 Multi dimensional feature processing and vectorization based on DKG

After designing DKG, due to the existence of multidimensional features, the study processed different features separately. According to the feature information extracted from DKG, college students with depression tendencies in DKG often use negative vocabulary in text features, and the frequency of using first person is much higher than that of normal users. The flowchart of multidimensional feature processing based on DKG is shown in Figure 2.

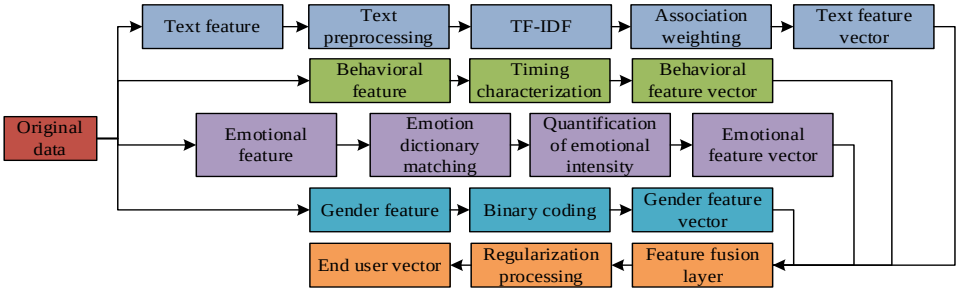


Figure 2: Flowchart of multi dimensional feature processing based on DKG.

In Figure 2, the raw data obtains four types of features: text features, behavioral features, emotional features, and gender features. After preprocessing the text features for word segmentation, the process encodes the semantic information of the text in vector form using Term Frequency Inverse Document Frequency (TF-IDF). Afterwards, feature weighting is applied to the word vector, specifically by assigning different weights based on the degree of association between the word and depression to construct the user's text feature vector. TF-IDF is a

weighting technique commonly used in information retrieval and natural language processing, which combines two metrics: Term Frequency (TF) and Inverse Document Frequency (IDF) to assess the significance of a word within a document [19]. The expressions for TF, IDF, and TF-IDF are shown in equation (1).

$$\begin{cases} \mathrm{TF}_{kn} = \dfrac{c_{k,n}}{\sum_j c_{j,n}} \\ \mathrm{IDF}_{kn} = \log \dfrac{|F|}{1+|n:e_k \in f_n|} \\ \text{TF-IDF}=\mathrm{TF}_{kn} \cdot \mathrm{IDF}_{kn} \end{cases} \quad (1)$$

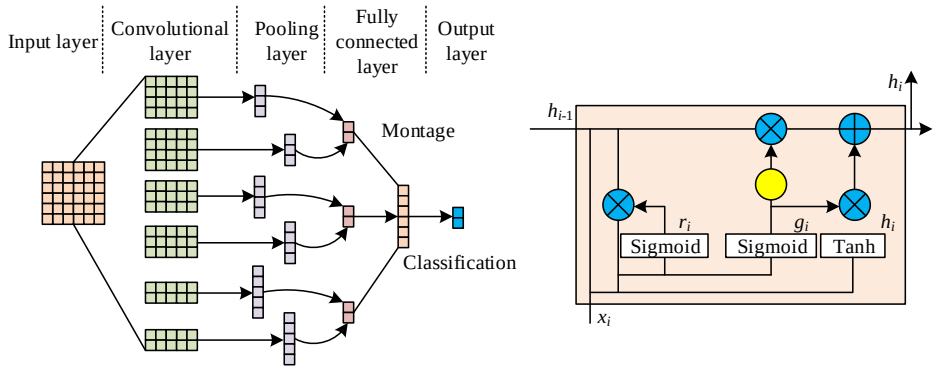
In equation (1), TF_{kn} is the frequency of the entry e_k appearing in the file f_n. IDF_{kn} means that if there are fewer documents containing entries and the IDF is larger, it means that the entries have a good ability to distinguish categories. $c_{k,n}$ is the number of occurrences of entry e_k in file f_n, and $\sum_j c_{j,n}$ is the sum of the number of occurrences of all words in file f_n. $|F|$ is the total number of files in the corpus. $|n:e_k \in f_n|$ indicates the number of documents containing entries [20]. The text feature vector of a certain Rednote user is T_k, $T_k = (q_1 d_1, q_2 d_2, ..., q_u d_u)$. Among them, q_u represents the feature weight of TF-IDF and d_u represents the u th dimension of the vector. For behavioral features, the study generates a feature behavior vector after sequential feature processing and normalization. It then processes emotional features using a Chinese sentiment lexicon for matching, followed by labeling of emotional polarity, with negative assigned as -1, neutral as 0, and positive as +1. The emotional intensity is quantified to generate an emotional feature vector [21]. Based on the processed text feature vectors, the study conducts emotional feature vectors, behavioral feature vectors, and gender feature vectors, feature fusion at the feature fusion layer. After regularization, the final user vector Q_k is obtained, and the representation of Q_k is shown in Equation (2). [22].

$$\begin{cases} Q_k = (q_s s_k + q_p p_k + q_a a_k + q_b b_k) \\ q_s + q_p + q_a + q_b = 1 \end{cases} \quad (2)$$

In equation (2), q_s is the weight of the user's gender characteristics, s_k is the user's gender, 0 for female and 1 for male. p_k and q_p are the frequency and weight of users' posting at night. a_k and q_a are online interactive data and their weights, b_k and q_b are user emotion categories and their weights.

2.3 Construction of MHP model based on CNN and GRU

DKG acts as a feature engineering and association framework, concatenating all processed multidimensional features (text, behavior, emotion, gender) in the feature fusion layer to form the final user vector representation, which serves as the input for the CNN-GRU model. This method effectively injects prior knowledge into feature representations without the need for complex graph embedding operations. By using users' text data and personal characteristics on Rednote, it is possible to predict whether they have a tendency towards depression, thereby achieving psychological health assessment. To ensure the full exploration of the richness of text information in Rednote, the study develops a MHP model combining CNN and GRU to better utilize text sequence relationships for feature extraction and classification. The schematic diagram of the structure of CNN and GRU is shown in Figure 3.



(a) CNN structure (b) GRU structure

Figure 3: Schematic diagram of the structure of CNN and GRU.

In Figure 3 (a), the input layer of CNN consists of four features of the user in DKG, and the output variables are the MHP results of that user. The CNN designed in MHP only uses local connections, combined with weight sharing to compress and purify text features, to ensure the generalization of the training model, reduce the computational complexity of training, and accelerate the convergence of the entire network. In Figure 3 (b), the

GRU function of the MHP model is to further obtain hidden information from the text sequence and improve the quality of feature extraction. It can overcome the local field of view limitations of CNN and understand the logical connections across sentences. GRU consists of update gate g_i and reset gate r_i, which dynamically control the flow of information through gating mechanisms. The update gate controls the fusion ratio of new and old information, while the reset gate determines how much historical information needs to be discarded.The expression of r_i and g_i is shown in equation (3) [23].

$$\begin{cases} r_i = \sigma\left\{q_r\left[h_{i-1}, x_i\right]\right\} \\ g_i = \sigma\left\{q_g\left[h_{i-1}, x_i\right]\right\} \end{cases} \quad (3)$$

In equation (3), σ is the Sigmoid function. q_r and q_g are the weights of reset door and update door respectively. x_i is the input at time i, and h_{i-1} is the hidden state at the previous time $i-1$. The expression of hidden state h_i at time i is shown in equation (4).

$$h_i = (1-g_i) \odot h_{i-1} + g_i \odot h^* \quad (4)$$

In equation (4), h^* is the candidate state. The reset gate determines the proportion of the incoming candidate state h^* from h_{i-1}, and gets the reset data through the reset gate after obtaining the gating signal. The candidate state generates the temporary memory of the current step, and the expression of h^* is shown in equation (5) [24].

$$h^* = \mathrm{Tanh}\left\{q\left[r_i \odot h_{i-1}, x_i\right]\right\} \quad (5)$$

In equation (5), $r_i \odot h_{i-1}$ represents the data after reset, $\left[r_i \odot h_{i-1}, x_i\right]$ indicates the concatenation operation with x_i. Tanh is the activation function. On the basis of DKG, a MHP model structure was constructed by combining CNN and GRU, as shown in Figure 4. In Figure 4, the MHP model primarily consists of a DKG, a CNN, a GRU, and a Fully Connected Layer (FC). Initially, the textual features from the DKG are transformed into continuous vector representations and then fed into the input layer of the CNN for spatial convolution. The CNN functions by extracting local features and performing spatial hierarchical processing to capture key semantic patterns indicative of depression from the textual data. It employs convolutional kernels of varying sizes to scan the word vector sequences, detect specific phrase combinations, and retain prominent features while suppressing noise through max pooling. The semantics processed by the CNN are subsequently modeled sequentially in the GRU, which can handle the sequential dependencies in the text. Compared to LSTM, the GRU has a simpler structure, making it suitable for medium-sized datasets and helping to avoid overfitting. The model uses a CNN to predict depressive tendencies among Rednote users, with input variables comprising DKG-based user features and binary output labels (0/1) indicating depression status. The role of textual feature extraction is to identify depression-related keywords, phrases, and their contextual patterns. Pre-trained Tencent word vectors are employed to convert the text into dense vectors, capturing semantic associations. The convolutional layer uses filters of different kernel sizes to extract local textual features. Max pooling retains the most significant features from each convolutional window, reducing dimensionality and enhancing robustness. The FC in the MHP model is an independent network layer positioned after the joint CNN-GRU feature extraction and before the final classification. This approach integrates textual features (obtained from the CNN-GRU) with user-level features into a composite multidimensional vector. The vector is then transformed into a lower-dimensional representation using a weight matrix, ensuring retention of key information. The ReLU activation function is used to learn complex interactions among features [25]. The dimensionality of the final output matches the number of categories, specifically representing two-dimensional information: depressed and non-depressed.

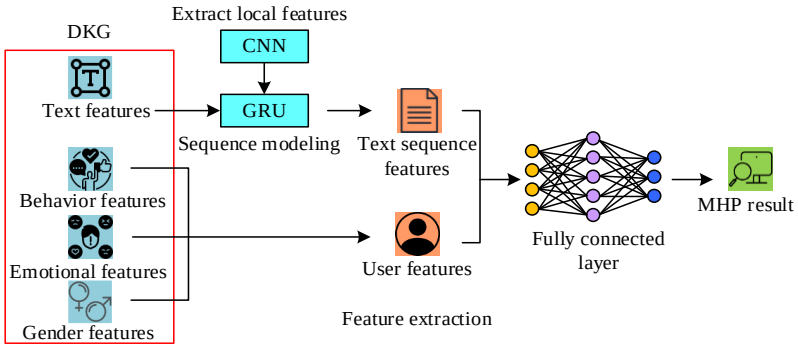


Figure 4: MHP model structure.

An automated warning and intervention system is designed based on MHP results. When the MHP model analyzes a student's social media data and their depression risk probability exceeds the preset threshold for one consecutive week, the system will automatically trigger a warning. This warning will be sent in real-time to

authorized psychological counselors through an encrypted internal messaging platform linked to the campus counseling center. Upon receiving a system alert, the counselor will initiate proactive outreach by dispatching standardized care messages through the designated messaging platform. Within three working days, the counselor will extend psychological counseling appointment invitations to targeted students, guided by objective data indicators provided by the system. Subsequent counseling content follows standard psychological counseling ethics and procedures.

3 Results

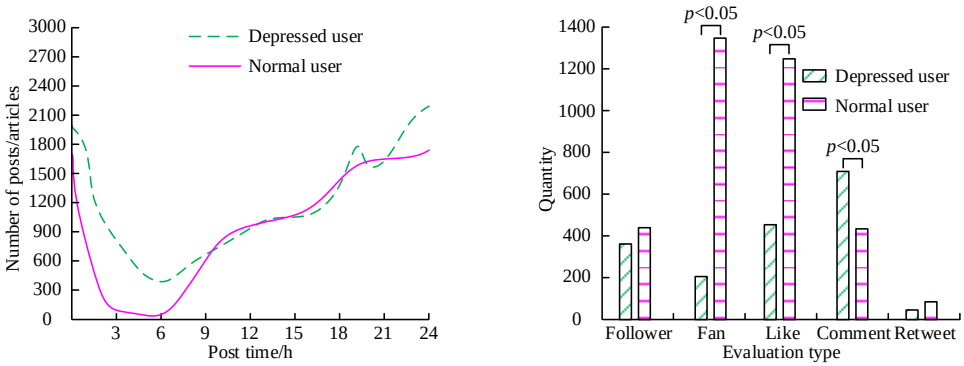
3.1 Testing and analysis of DKG and CNN-GRU models

The hardware configuration for experimental testing was NVIDIA Tesla-A100-80G GPU, 16 core CPU, and 128GB DDR4 memory. The crawler framework selected was Scrapy+Selenium, and the deep learning framework Pytorch 2.0.1. The hyperparameters and training configurations of the model are shown in Table 1.

Table 1: Model hyperparameters and training configuration.

Parameter category	Parameter name	Parameter value/description
CNN architecture	Convolution kernel size	[2, 3, 4]
	Number of convolution kernels per size	128
	CNN activation function	ReLU
	Pooling method	Maximum global pooling
GRU architecture	GRU number of plies	2
	Number of hidden cells per layer	128
Fully connected layer	Number of hidden layer neurons	128
	Number of output layer neurons	2
	Activation function	ReLU / Softmax (output layer)
Training parameters	Optimizer	Adam
	Learning rate	0.001
	Batch size	32
	Training rounds	200
	Dropout rate	0.5
	Loss function	Cross-entropy loss

To ensure the statistical validity of the research results, independent sample *t*-test and repeated measures analysis of variance were used in this study. Cross validation calculations were conducted through 5 repeated experiments to evaluate the performance indicators of the model. All model performance metrics, such as accuracy, F1 score, etc., represented the performance of the model on an independent test set. To analyze the rationality of the selection of specific behavioral characteristics in DKG, the study analyzed the posting time and online interaction changes of depressed and normal users on the Rednote social media platform, in order to analyze the behavioral characteristics of different users. On the platform, the indicators of user followers, fans, likes, comments, and shares were important criteria for evaluating user activity and influence. To investigate the differences in online interaction between depressed users and normal users, the crawled user posts were statistically analyzed, and their number of followers and fans were recorded. The posting time and online interaction changes of the two types of users are shown in Figure 5.



(a) Change in posting time (b) Online interactive changes

Figure 5: Posting time and online interaction changes.

In Figure 5(a), depressed users posted more messages than normal users between 0:00-7:00 and 22:00-24:00, while the number of posts by the two types of users showed little difference and no significant discrepancy during the 7:00-21:00 period. Specifically, depressed users posted an average of 350 more messages than normal users between 0:00-7:00. In Figure 5(b), depressed users had 216 followers and received 461 likes, whereas normal users had 1,348 followers and received 1,254 likes. The independent sample *t*-test results showed that the number of fans and likes of depressed users were significantly lower than those of normal users (t=3.21, p<0.05). Depressed users made 719 comments, reflecting an increased need for emotional catharsis, and their number of comments was significantly higher than that of normal users (p<0.05). However, there were no significant differences in the number of follows and reposts between the two types of users.

To evaluate the generalization ability of the MHP model, this study used a dataset collected and constructed from the Rednote platform as the unique training set to train the final MHP model. The data from Weibo, Zhihu, and campus forum platforms was only used as an independent test set and did not participate in any training or parameter adjustment of the model. The study examined the emotional distribution of datasets obtained from four platforms: Rednote, Weibo, Zhihu, and campus forums. Specifically, 26,000 data entries were collected from Weibo, 20,000 from Zhihu, and 12,500 from campus forums. The emotional distribution of datasets from different platforms is shown in Table 2.

Table 2: Emotion distribution in datasets from different platforms.

Platform type	Dataset partitioning	Positive	Neutral	Negative
Rednote	Training set	24000	18000	12000
	Test set	6000	4500	3000
Weibo	Training set	9000	6000	5000
	Test set	3000	2000	1000
Zhihu	Training set	6000	4500	4500
	Test set	2000	1500	1500
Campus forum	Training set	4800	3200	2000
	Test set	1200	800	500

Based on the obtained text data, the emotional categories of users were analyzed, and the proportion of basic emotions between depressed users and normal users was compared. Emotions were divided into happiness, sadness, anger, fear, disgust, surprise, and shame. The proportion of different emotional intensities is shown in Figure 6.

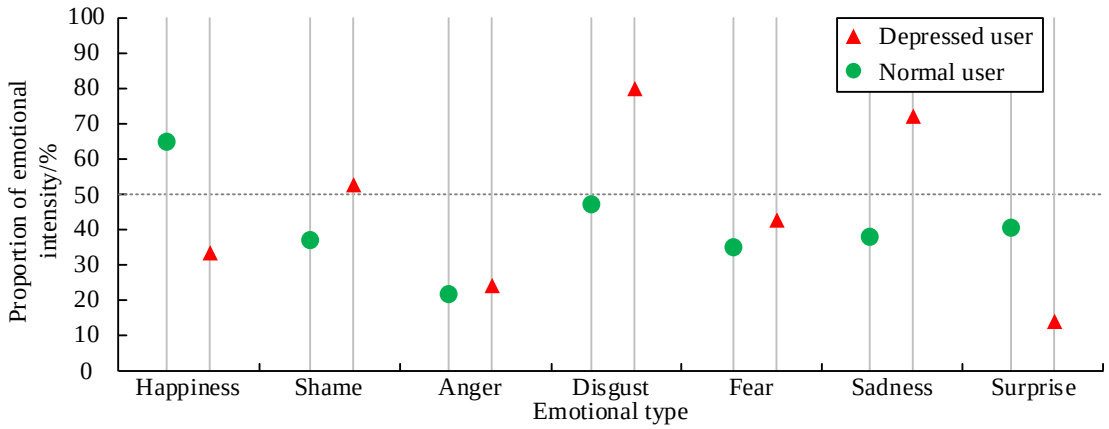


Figure 6: Proportion of intensity of different emotions.

In Figure 6, the proportion of sadness, disgust, and shame among depressed users exceeded 50%, accounting for 72.4%, 80.3%, and 53.1% respectively. These emotions were more consistent with the signature depressive emotions in the scale. In contrast, the proportion of sadness, anger, fear, disgust, surprise, and shame emotions among normal users was less than 50%, while happiness accounts for 65.4%. Normal users had a certain level of healthy emotional regulation.

The experimental parameters were set with a batch size of 32, a training iteration of 200, convolution kernel sizes of [2-4], a learning rate of 0.001, and a Dropout of 0.5. The study compared and analyzed GRU, Transformer, and CNN-GRU, and the results of their accuracy, recall, and F1 score changes are shown in Figure 7.

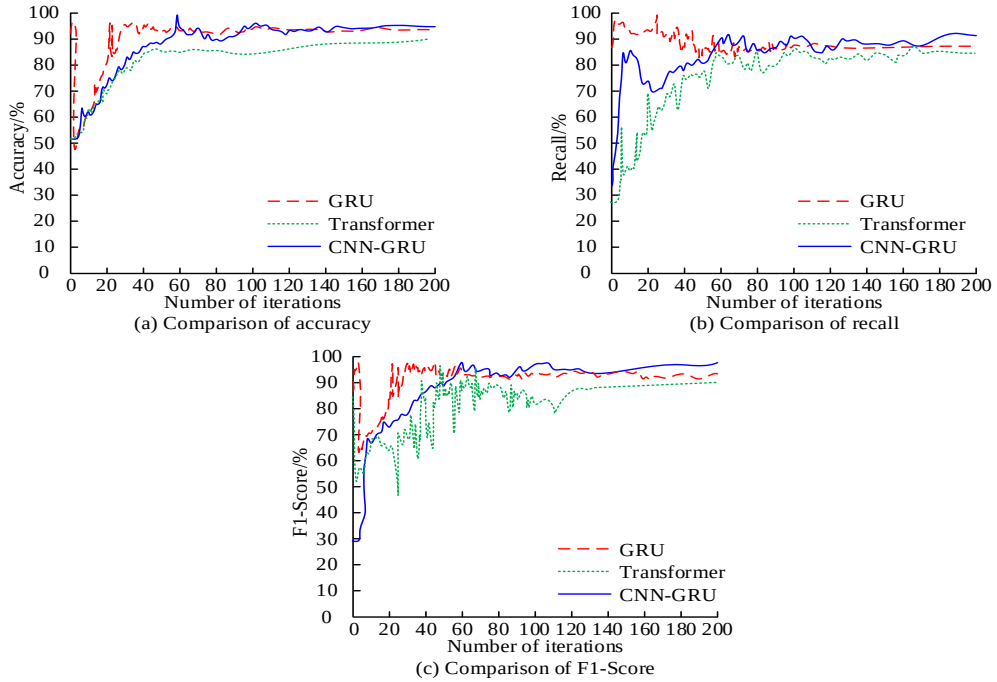


Figure 7: Accuracy, recall, and F1 score changes of different models.

In Figure 7 (a), the three models converged after approximately 110 iterations, and the overall GRU was lower than CNN-GRU, with an accuracy of about 92.8%. GRU had strong modeling ability for long sequences, but was susceptible to interference from irrelevant words. The accuracy of CNN-GRU increased rapidly in 200 iterations and remained stable at a high level of approximately 95.6% in the later stages. The accuracy of Transformer was only 89.3%. In contrast, CNN-GRU had stronger classification accuracy and recognition ability. In Figure 7 (b), the overall change in GRU recall rate was a fluctuating decrease, ultimately stabilizing at 87.7%. The recall rates of Transformer and CNN-GRU increased with the number of iterations, and the final recall rates were 85.1% and 90.4%, respectively. The advantages of CNN-GRU in local extraction and temporal features enabled better analysis of emotions, thus enabling MHP implementation. In Figure 7 (c), after 0-60 iterations, the F1 score of GRU was higher than that of CNN-GRU, but its variation curve fluctuated more significantly. After 60-200 iterations, the

F1 score of CNN-GRU was higher than that of GRU, reaching 98.5%, which was 6.3% higher than GRU's 92.2%. The F1 score of Transformer fluctuated in the first 110 iterations and ultimately reached about 90.1%. It had good global dependency capture ability, but required a large amount of training data to obtain good detection results.

3.2 MHP model performance analysis and application results

To verify the effectiveness of different modules in the MHP model, ablation experiments were designed. Each model configuration (including MHP and its variants) in the ablation experiment underwent 5 random initialization repetitions of training and testing to evaluate the robustness of the model. The MHP model included DKG+CNN-GRU+FC. The module without DKG included CNN-GRU+FC, the module without CNN included DKG+GRU, the module without GRU included DKG+CNN, and the baseline model was pure FC. The evaluation results of different models on the same dataset are shown in Figure 8.

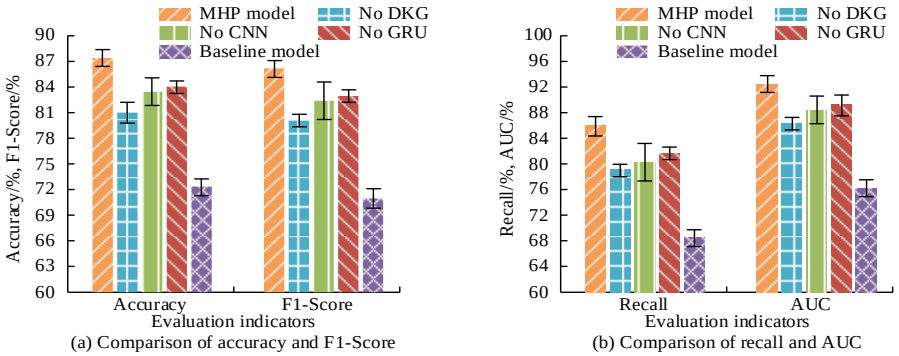


Figure 8: Results of ablation experiment.

In Figure 8 (a), the accuracy and F1 score of MHP model were 87.3% and 86.5%, respectively. Compared with MHP model, the baseline model showed the greatest decrease in accuracy and F1 score, with a decrease of 14.9% and 15.7%, respectively. The accuracy and F1 score without DKG decreased by 6.1% and 6.4% respectively compared to MHP model, indicating a significant contribution of DKG to feature association modeling. The F1 scores without CNN and GRU decreased by 4.1% and 3.5% respectively compared to MHP model. CNN in MHP model could detect negative phrases in text information, while GRU could analyze emotional evolution in text information. In Figure 8 (b), the recall rate and AUC of MHP model were 85.8% and 92.3%, respectively. Compared to MHP model, the recall rate of the baseline model decreased by 17.5%. The AUC without DKG was 86.1%. The AUC without GRU was 89.1%, which was 0.9% higher than the 88.2% without CNN.

To analyze the contribution of four types of features in DKG to MHP model decision-making, this study conducted feature importance analysis and trained the MGP model with logistic regression and SVM models on the same multi-dimensional feature vectors of DKG. The results of the two experimental analyses are shown in Figure 9.

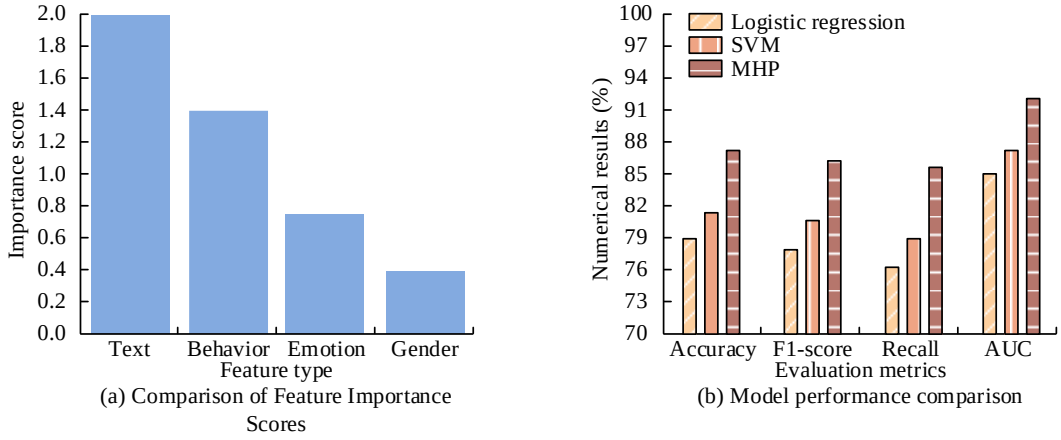


Figure 9: Feature importance analysis and model performance comparison.

In Figure 9 (a), the importance of text features was the highest, with a score of 2.0. Its ranking importance score was five times that of gender characteristics (0.4 points). This confirmed that language patterns were the strongest signals for identifying depressive tendencies on social media. The importance of behavioral characteristics

was secondary, with a score of 1.4. Its importance was significantly higher than emotional foundation and gender characteristics, which validated the necessity of incorporating behavioral patterns into DKG. Emotional features also showed some importance, but their contribution was smaller than that of textual and behavioral features. The importance of gender characteristics was relatively low, but as prior knowledge of risk statistics, it still played an auxiliary adjustment role in the model. This analysis not only quantified the contributions of different features in DKG, but also made the decision-making process of the model more transparent. In Figure 9 (b), the proposed MHP model significantly outperformed logistic regression and SVM in accuracy, F1 score, and AUC. Compared with logistic regression (78.0%), the F1 score of MHP model (86.5) increased by about 8.5%. This confirmed the advantage of deep learning models in capturing complex nonlinear patterns in text sequences. Compared with SVM (79.2%), MHP model performed better in recall rate (85.8%) and could more effectively identify users with depressive tendencies. This comparison proved that introducing the CNN-GRU architecture to handle such problems could bring substantial performance improvements compared to classical methods that only use feature engineering.

The study selected a 6-month post sequence of 100 depressed college students from the Rednote platform. The changes in the proportion of nighttime posts, online interaction attenuation rate, negative word growth rate, and PHQ-9 growth rate of users are shown in Table 3.

Table 3: Changes in the temporal characteristics of users.

Time (month)	Night post proportion (%)	Online interaction attenuation rate (%)	Negative word growth rate (%/week)	PHQ-9 increase
1~2	12→18	0.05	1.2	1.5
3~4	18→34	0.12	3.5	4.8
5~6	34→51	0.21	6.7	9.2

In Table 3, during the observed months of January and February, the proportion of nighttime posts by this type of user increased by 6% ($p<0.05$), by 16% ($p<0.01$) from March to April, and by 17% ($p<0.1$) from May to June. Within 6 months, the online interaction attenuation rate of users gradually increased, the growth rate of negative words gradually accelerated, and the PHQ-9 growth rate increased.

To verify the direct cross platform generalization ability of the MHP model without domain adaptation, this study applied the final model trained on Rednote directly to independent test sets from Weibo, Zhihu, and campus forums for evaluation. There were significant differences between these test sets and training sets in terms of user groups, language styles, and platform culture. The results are shown in Table 4.

Table 4: Application testing results on different platforms.

Platform type	Data volume	Accuracy (%)	F1 score (%)	Recall (%)
Rednote	67500	87.3	86.5	85.8
Weibo	26000	87.2	85.6	83.9
Zhihu	20000	81.3	79.7	77.5
Campus Forum	12500	89.5	88.1	86.3

In Table 4, the MHP model had an F1 score of 85.6% on the Weibo platform, and the text structures published on both platforms were similar. However, the F1 score on the Zhihu platform was relatively low, only 79.7%, which was due to the fact that the Zhihu platform had a more professional academic background, so its performance was not as good as the other two platforms. The campus forum platform had specialized vocabulary and focused on different topics. The complexity of model analysis was not high, and the recognition of text information was high, with an F1 score of 88.1%. The evaluation results on Weibo and campus forums showed that the accuracy of the MHP model was 87.2% and 89.5%, respectively, demonstrating high performance. This indicated that the features learned by the MHP model from Rednote had good transferability on these platforms. However, the accuracy on the Zhihu platform was 81.3%, indicating a decrease in performance. This indicated that there were indeed distribution differences between different platforms, and also emphasized the necessity of introducing domain adaptation technology in the future to further enhance the universality of the model. Overall, cross platform testing results demonstrated that the MHP model had a certain degree of inherent robustness and practical potential.

To verify the effectiveness of the proposed automated warning and intervention system, the study selected 400 students from Beijing to conduct real-time intervention effect tests on the MHP model, and the participants' PHQ-9 scale results were all greater than or equal to 10 points. This study was approved by the ethics review committee and informed consent was obtained. The participants were divided into two groups, A and B, with 200 people in each group. Group A adopted the intervention method of MHP model warning and consultation, while Group B only conducted routine paper questionnaire screening, with a trial evaluation period of 4 months. The intervention results are shown in Table 5, using the change in PHQ-9 score, consultation conversion rate, and number of crisis events as measurement indicators.

Table 5: Comparison of intervention results.

Measurement indicator	A group	B group	Cohen's d/OR	*p* value
PHQ-9 score change value (points)	-4.2±1.8	-1.1±2.3	1.57	<0.001
Consultation conversion rate (%)	63	22	/	0.008
Number of incidents of crisis events (cases)	2	11	0.17	0.013

In Table 5, the PHQ-9 score of Group A decreased by an average of 4.2 points after the end of the experiment, while Cohen's d was 1.57, indicating that the intervention method of Group A was highly effective. The counseling conversion rate of Group A was 63%, and the MHP model warning of this group could effectively improve the conversion rate of psychological counseling. Comparing the number of crisis events between the two groups, the Odds Ratio (OR) was 0.17, which meant that the risk of crisis events in Group A was reduced by 83% compared to Group B.

4 Discussion and conclusion

There are problems with insufficient multidimensional data fusion and weak real-time detection in the current assessment of college students' mental health. To achieve efficient processing of college students' mental health sign analysis, a MHP model based on DKG and CNN-GRU was proposed. The findings demonstrated that the accuracy and F1 score without DKG were 81.2% and 80.1%, respectively, which were 6.1% and 6.4% lower than MHP model, respectively. This indicated that DKG contributed significantly to feature association modeling. Reference [5] used the HADS scale combined with SVM, but did not take into account daily behavioral characteristics. In contrast, the design of this study was more efficient in integrating multidimensional data on social media platform, PHQ-9 scale and DKG. The F1 score of the CNN-GRU designed in the study was 98.5%, with an accuracy of 95.6%, which was better than the CNN and BiLSTM performance in reference [11]. The approach proposed in reference [14] considered electronic health records and neural impact data, but the economic cost of actual use was relatively high. In contrast, this study used crawler technology to obtain multidimensional data, and the cost was lower. This study not only achieved the fusion of multi-source data, but also improved the accuracy of MHP, providing a reliable intervention tool for college students' psychology. Although this study achieved good preliminary results, the current research relied on a single data source, which did limit the full validation of the model's universality. Just as adaptive fuzzy control [26] and adaptive backstepping control [27] methods respond to unmodeled dynamics and external disturbances in dynamic systems by adjusting parameters online, a truly robust MHP model also needed to be able to adapt to the uncertainty brought by various data distributions, such as those from different social media platforms and student groups. The current work can be seen as validating the effectiveness of the core architecture under relatively controlled initial conditions, laying a solid foundation for subsequent adaptive scaling. Inspired by the search for dynamic optimal solutions in complex systems using nonlinear optimal control [28], this study planed to analyze the dynamic update mechanism to enable the MHP model to make online adjustments over time and changes in user data distribution, to maintain its prediction accuracy and achieve continuous tracking of the evolution of user psychological states. Future research will expand the scope of the study from ordinary college students to include more diverse groups such as international students and graduate students, in order to evaluate the performance of the model in different subgroups and ensure the broad applicability of its intervention strategies.

References

[1] Li H .Multicultural data assistance mining analysis for ideological and political education in smart education platforms using artificial intelligence.Wireless Networks, 2025, 31(1):567-581.DOI:10.1007/s11276-024-03772-8.

[2] Gao Y. The impact and application of artificial intelligence technology on mental health counseling services for college students. Journal of Computational Methods in Sciences and Engineering, 2025, 25(2): 1686-1701. DOI: 10.1177/14727978241302641

[3] Peng X , Nie R , Tong S .Correction to: Research on the selfefficacy and resilience of female graduate students in the era of artificial intelligence: analysis of the mechanism of mobile phone dependence, anxiety and mentoring relationship.Archives of Women's Mental Health, 2025, 28(4):963-963.DOI:10.1007/s00737-025-01556-3.

[4] Lamontagne L G , Doty J L , Diehl D C ,Nesbit, T. S. , Gage, N. A. , & Kumbkarni, N. , et al.Acceptability, usage, and efficacy of mindfulness apps for college student mental health: A systematic review and meta-analysis of RCTs.Journal of Affective Disorders, 2024, 367(000):951-971.DOI: 10.1016/j.jad.2024.09.014.

[5] Krishnan R, Kumari S, Badi A A, Jeba, S., & James, M. Predictive machine learning model for mental health issues in higher education students due to COVID-19 using HADS assessment. Arab Gulf Journal of Scientific Research, 2024, 42(4): 1472-1490. DOI: 10.1108/AGJSR-01-2023-0034.

[6] Zhai Y, Zhang Y, Chu Z, Geng, B., Almaawali, M., Fulmer, R., Du, X. et al. Machine learning predictive models to guide prevention and intervention allocation for anxiety and depressive disorders among college students. Journal of Counseling & Development, 2025, 103(1): 110-125. DOI: 10.1002/jcad.12543.

[7] Cai Y, Lin D, Lu Q. Comparison of Different Machine Learning Algorithms in the Mental Health Assessment of College Students. Journal of ICT Standardization, 2024, 12(4): 409-427. DOI: 10.13052/jicts2245-800X.1243.

[8] Tyulepberdinova G, Mansurova M, Sarsembayeva T, Issabayeva, S., & Issabayeva, D. The physical, social, and mental conditions of machine learning in student health evaluation. Journal of Computer Assisted Learning, 2024, 40(5): 2020-2030. DOI: /10.1111/jcal.12999.

[9] Bhavani B H, Naveen N C. An approach to determine and categorize mental health condition using machine learning and deep learning models. Engineering, Technology & Applied Science Research, 2024, 14(2): 13780-13786. DOI:10.48084/etasr.7162.

[10] Ji X. Research on Mental Health Assessment and Intervention Methods for College Students based on Big Data Analysis. Scalable Computing: Practice and Experience, 2024, 25(6): 4702-4711. DOI: 10.12694/scpe.v25i6.3285.

[11] Ming Q. An evaluation model for college students' mental health based on machine learning algorithm. International Journal of Wireless and Mobile Computing, 2025, 28(1): 58-67. DOI: 10.1504/IJWMC.2025.143017.

[12] Li J. Dynamic evaluation of college students' psychological state based on multimodal physiological signal fusion and deep generation model. International Journal of Information and Communication Technology, 2025, 26(17): 133-146. DOI: 10.1504/IJICT.2025.146669.

[13] Marco Calabrò, Fabbri C , Serretti A ,et al.A machine learning approach to predict treatment efficacy and adverse effects in major depression using CYP2C19 and clinical-environmental predictors.Psychiatric Genetics, 2025, 35(2):17-25. DOI: 10.1097/YPG.0000000000000379.

[14] Zhang Z. Early warning model of adolescent mental health based on big data and machine learning. Soft Computing, 2024, 28(1): 811-828. DOI: 10.1007/s00500-023-09422-z.

[15] Gao Q, Wei Y. Understanding the cultivation mechanism for mental health education of college students in campus culture construction from the perspective of deep learning. Current Psychology, 2024, 43(2): 1715-1732. DOI: 10.1007/s12144-023-04320-x

[16] Zeng Z , He J , Xiang T , Wang, N. , Chen, B. , & Guo, S. Cognitive Tracing Data Trails: Auditing Data Provenance in Discriminative Language Models Using Accumulated Discrepancy Score.Cognitive Computation, 2024, 16(6):3379-3390. DOI:10.1007/s12559-024-10315-y.

[17] Cherradi M, El Haddadi A. Enhancing Data Lake Management Systems with LDA Approach. Journal of Data Science and Intelligent Systems, 2025, 3(1): 58-66. DOI: 10.47852/bonviewJDSIS42022312.

[18] Agbesi V K, Chen W, Ukwuoma C C, Kuadey, N. A., Agbesi, C. C. M., Ejiyi, C. J., Kuadey, A. M. et al. Multichannel 2D-CNN attention-based BiLSTM method for low-resource ewe sentiment analysis. Journal of Data Science and Intelligent Systems, 2025, 3(1): 67-77. DOI: 10.47852/bonviewJDSIS32021512.

[19] Singh H, Kaur B, Sharma A, & Singh, A. Framework for suggesting corrective actions to help students intended at risk of low performance based on experimental study of college students using explainable machine learning model. Education and Information Technologies, 2024, 29(7): 7997-8034. DOI: 10.1007/s10639-023-12072-1.

[20] Norouzi F , Machado B L M S , Nematzadeh S .Schizophrenia Diagnosis and Predictionwith Machine Learning Models. International Journal of Scientific and Academic Research, 2024, 04(09):113-122. DOI: 10.54756/ijsar.2024.26.

[21] Pei J, Amanvermez Y, Vigo D, Puyat, J., Kessler, R. C., Mortier, P., Cuijpers, P. et al. Sociodemographic correlates of mental health treatment seeking among college students: A systematic review and meta-analysis. Psychiatric services, 2024, 75(6): 556-569. DOI: 10.1176/appi.ps.2023041.

[22] Wedyan M, Saeidi-Rizi F. Assessing the impact of urban environments on mental health and perception using deep learning: A review and text mining analysis. Journal of Urban Health, 2024, 101(2): 327-343. DOI: 10.1007/s11524-024-00830-6.

[23] Huda A M, Shidik G F, Praskatama V. COMPARATIVE ANALYSIS OF LSTM, BILSTM, GRU, CNN, AND RNN FOR DEPRESSION DETECTION IN SOCIAL MEDIA. Jurnal Teknik Informatika (Jutif), 2024, 5(6): 1723-1735. DOI: 10.52436/1.jutif.2024.5.6.4060.

[24] Taylor M E , Liu M , Abelson S , Eisenberg, D. , Lipson, S. K. , & Schueller, S. M.The Reach, Effectiveness, Adoption, Implementation, and Maintenance of Digital Mental Health Interventions for College Students: A Systematic Review.Current Psychiatry Reports, 2024, 26(12):683-693.DOI: 10.1007/s11920-024-01545-w.

[25] Magalhães E. Dual-factor models of mental health: A systematic review of empirical evidence. Psychosocial intervention, 2024, 33(2): 89-102. DOI: 10.5093/pi2024a6.

[26] Boubellouta; A B Z .Adaptive fuzzy control for practical fixed-time synchronization of fractional-order chaotic systems. Sage Publications[2025-10-20]. DOI: 10.1177/10775463251320258.

[27] Boulkroune A , Hamel S , Zouari F ,et al.Output-Feedback Controller Based Projective Lag-Synchronization of Uncertain Chaotic Systems in the Presence of Input Nonlinearities.Mathematical Problems in Engineering, 2017, 2017:1-12. DOI: 10.1155/2017/8045803.

[28] Rigatos G, Abbaszadeh M, Sari B, et al. Nonlinear optimal control for a gas compressor driven by an induction motor. Results in Control and Optimization, 2023, 11: 100226. DOI: 10.1016/j.rico.2023.100226.

Optimizing Charging Infrastructure for Electric Autonomous Vehicles in Smart Manufacturing using Hyperparameter-Tuned Artificial Neural Networks

Yu Liu[1,2*], Lina Zhangi[2], YanHong Fu1,2
[1]Department of Mechanical Engineering Hebei University of Water Resources and Electric Engineering, CangZhou HeBei, 061001, China
[2]Hebei Industrial Manipulator Control and Reliability Technology Innovation Center, CangZhou HeBei, 061001, China
E-mail: liuyu@hbwe.edu.cn
[*]Corresponding author

Keywords: Electric Autonomous Vehicles (A-EVs), Smart Manufacturing Systems, Charging Infrastructure Optimization, Machine Learning, AI-Driven Scheduling

Received: October 24, 2025

An optimised charging infrastructure is required for the integration of electric autonomous vehicles (A-EVs) into smart industrial systems in order to improve operational efficiency, minimise downtime, and guarantee sustainable energy management. An AI-driven framework for the planned and strategic placement of A-EV charging stations in industrial settings is presented in this paper. The platform uses predictive analytics, real-time IoT data, and machine learning models to dynamically modify charging station allocation according to industrial workflow requirements, grid stability, and car demand. To guarantee the best possible positioning and use of charging stations, the suggested methodology combines optimisation algorithms, reinforcement learning-based scheduling, and spatiotemporal data analysis. This study maximizes charging infrastructure to electric autonomous vehicles with the A-EV Grid Management Dataset through the use of KNN-based imputation, Z-score outliers, Min-Max scaling, SMOTE balancing, and feature selection with RFE and Lasso. An ANN model is hyperparameter-tuned and stratified 5-fold cross-validation is used to evaluate the model. The experimental results indicate that the proposed ANN is more accurate and has 94.5% accuracy, AUC-ROC as 95.2%, and 5.3% as MAE, which is higher than RF, SVM, KNN, and Decision Tree baselines. The results indicate that the model is efficient in smart manufacturing systems to support the real time charging decisions. The results of comparative performance evaluations utilising Random Forest, SVM, KNN, Decision Tree, and a hyperparameter-tuned Artificial Neural Network (ANN) show that the ANN model predicts optimal charging decisions with greater accuracy (94.5%) and precision (93.7%). Compared to traditional static deployment options, the results demonstrate a significant improvement in energy efficiency, less congestion at charging stations, and cheaper operating costs. This study emphasises how crucial AI-enhanced infrastructure planning is for smart manufacturing's autonomous EV fleets. To better optimise A-EV charging networks, future studies may investigate blockchain for decentralised energy management, multi-agent reinforcement learning for adaptive station allocation, and 5G-enabled V2X communication. The results further the development of highly automated, intelligent, and energy-efficient industrial transportation systems.

Povzetek: Predstavljen je okvir za dinamično optimizacijo polnilne infrastrukture avtonomnih električnih vozil v pametni industriji, kjer s predobdelavo podatkov in hiperparametrično uglašenim nevronskim omrežjem doseže boljše rezultate.

1 Introduction

In the age of digital revolution, breakthroughs in technology, particularly in the area of artificial intelligence (AI), have upended a number of global enterprises. Thanks to the application of cutting-edge information technology, AI has been extensively embraced and has had a substantial influence in all industries. As an environmentally benign form of transportation, electric vehicles (EVs) have also attracted a lot of interest lately and are being promoted by several nations[1]. Because they produce no emissions, EVs are good for the

environment. They also feature smooth and user-friendly controls and are less expensive than conventional petrol engines[2]. EVs are a good option for people who value environmental sustainability because they don't emit any pollutants. Additionally, driving an EV is less expensive than utilising a traditional petrol engine. Additionally, EVs offer a pleasant driving experience and are simple to use[3].

As of 2020, there were more than 7.2 million electric vehicles (EVs) on the road, according to the International Energy Agency (IEA), which reports that the EV industry has been expanding quickly worldwide. Furthermore, the IEA projects that the EV industry will keep expanding, with over 250 million EVs in use globally by 2030[4]. Additionally, it is anticipated that the EV industry would grow substantially. The worldwide EV industry is expected to grow from its 2020 valuation of over USD 140 billion to over USD 800 billion by 2026, according to Statista. According to a Deloitte report, government incentives, more stringent pollution laws, and technological developments are projected to fuel the EV market's continued growth[5]. The adoption of EVs is expected to continue to rise due to the growing availability of EV models, charging infrastructure, and government backing. Because of its many benefits, including economic effectiveness, environmental preservation, smart city traffic improvement, and increased customer happiness, EV adoption is thus expanding quickly on a global scale[6].

EVs are becoming more and more well-liked as an environmentally responsible and sustainable mode of transportation. The interdependence of the energy market and EV charging infrastructure is reflected in the connection of the transportation and energy markets. On the other side, power-aware operations manage power limitations to maximise the utilisation of electrical resources and offer effective EV charging services. Power-aware operations in the context of charging infrastructure make sure that charging stations provide dependable charging services while optimising the use of electrical resources. To effectively manage the charging process, this entails taking into account elements like power capacity, load balance, and dynamic power scheduling[7].

An autonomous car is a vehicle capable of sensing its environment and operating without human involvement. A human passenger is not required to take control of the vehicle at any time, nor is a human passenger required to be present in the vehicle at all. An autonomous car can go anywhere a traditional car goes and do everything that an experienced human driver does.

According to Navigant Research, 85 million autonomous vehicles are anticipated to be marketed globally each year by 2035. But in addition to drastically changing transportation, the self-driving vehicle will also create a completely new image of city life. It will be necessary to reconsider the infrastructure on which cities are constructed. In addition to requiring electricity, autonomous cars have the ability to serve as a mobile power source for other networked devices. Wireless power is an essential component of the ecology of the smart city of the future.

For those who do not wish to own a car for personal or economical reasons, autonomous cars will provide a new form of personal transportation. They will also provide the enormous advantage of mobility for those who are unable to operate a vehicle, such as children, the elderly, and individuals with disabilities. There will be more chances for people to be active and mobile in their communities. Everyone will be a passenger when there is no driver, giving them the opportunity to work and be productive throughout the travel or simply relax and refuel. Additionally, the entire layout of the vehicle could be redesigned. Lounge chairs that face inward may be a feature of the future automobile. The car will become more than just a means of transportation; it will be an extension of the house or workplace, bringing in additional activities and hours to the day.In addition to those advantages, smart technology will enable cars to automatically avoid road dangers, making streets safer and reducing accidents. By outlawing drunk driving, public safety could be improved. Since cars will go at the same speed at regular intervals, traffic congestion may lessen. Because cars would be able to drop off their passengers and then go off to park themselves, less time will be wasted looking for parking spaces. Parking lots and garages will take up less valuable real estate in city centres[8].



Figure 1: To optimize autonomous travel, add wireless charging

Personal transportation will undoubtedly be revolutionised by autonomous vehicles, but a crucial question has gone mostly unanswered: how will these vehicles be fuelled? Think of the irony of having self-driving electric vehicles (EVs) transport our children to school, our injured to medical appointments, and our elderly to the park, only to need someone to plug them in after the journey is over. There must be a more straightforward way. In the same way that it removes people from the driving equation, the

charging procedure of the future must remove people from the charging equation. Without ever plugging in, a car will autonomously recharge its battery when it needs to be charged by parking itself over a wireless charging pad. It's a risky move to replace the power plug, and it will take wireless technology that can effectively transfer a car's whole power requirements and is adaptable enough to work with a range of automobiles. Magnetic resonance is arguably the most promising technology. Actually, a lot of automakers are already moving quickly to start producing future electric cars. Among the many advantages of magnetic resonance are its high efficiency, adaptability in location, and capacity to transmit electricity through materials such as asphalt and concrete. Wireless charging stations can be placed on the ground, in a garage floor, or beneath a road thanks to magnetic resonance technology. There is no penalty for going wireless, and the most recent designs are remarkably efficient in moving electricity at the same levels as plugging in. Self-driving cars will just park and recharge their batteries. Vehicle transport can become another standard service, similar to the internet or the electric grid, without requiring human involvement to replenish batteries. Passengers can then concentrate on other issues, knowing that their cars will be available and ready when needed.
Innovative businesses can collaborate with governments and infrastructure developers to establish citywide networks of wireless power sources integrated into the streets as the technology advances and volume lowers costs. Unmanned transporters will be able to increase driving range or lower the size and expense of the batteries needed to power the vehicles by using these integrated sources to wirelessly trickle-charge the vehicles while they are in motion. Increased adoption results from reduced costs, starting a positive feedback loop. There is an extensive analysis of the strategies for placing charging stations as well as their effects on the electrical system. The most up-to-date optimisation methods for determining the best places to put charging stations are discussed. The study examines the policies, procedures, and issues associated with the location of electric vehicle charging stations in China. What this study adds to the current literature on the topic of electric vehicle charging station placement is as follows:

The models used to determine the optimal locations for charging stations are categorised. By outlining the planning models, we may better understand the mathematical formulations and the outcomes of the simulations. We give a comparison of the planning models, highlighting their salient characteristics, benefits, and drawbacks. We propose that the planning models be suitable on an area-by-area basis. This document provides a high-level outline of the charging station's operations and how they are scheduled. Presenting the current business model, standards, norms, and regulations, this article provides a global perspective on charging infrastructure design.

Research questions

1. Can a hyperparameter-tuned ANN outperform conventional ML models in optimizing A-EV charging decisions under dynamic grid conditions?
2. What is the most effective way to implement charging infrastructure in electric autonomous vehicles (EAVs) to make smart manufacturing systems more efficient and energy-oriented?
3. How much can the proposed hyperparameter-optimized ANN model be used to predict charging demand and station utilization better than traditional machine-learned algorithms, like RF, SVM, KNN, and Decision Trees?
4. Which operational, spatial and temporal factors are the most effective in determining where and when to locate EAV charging stations in smart manufacturing settings?
5. What effect does the optimized charging infrastructure have on the actual performance considerations, such as use of energy, money saved, grid load balancing, and vehicle availability?

2 Related works

Table: Related works

Ref.	Objectives	Methods	Dataset	Performance Metrics	Limitations
[9]	Evaluate benefits of integrating EVs into the power grid	Grid-EV interaction analysis; energy modeling	Grid operational data; EV charging behavior data	Accuracy with 88.4	Lacks real-time dynamic modeling; no ML-based optimization
[10]	Optimize energy supply for EV charging parks & hydrogen stations under risk-averse strategy	Stochastic optimization; risk-averse programming	Local energy system operational datasets	Cost reduction, reliability improvement (no numeric ML metrics)	High computational complexity; limited scenario generalization
[11]	Plan mobile energy hubs in complex urban networks	Robust optimization framework	Urban mobility network data, energy hub specs	Accuracy with highest value of 90.4 percent	Focuses on robustness, not predictive analytics;

					limited EV-specific analysis
[12]	Review modeling methods for planning EV charging infrastructure	Systematic literature review	100+ EV infrastructure planning studies	nil	No experimental validation; lacks quantitative comparison
[13]	Use ML to optimize EV charging operations	Machine Learning models (unspecified)	Real-time EV charging logs	Accuracy, RMSE (values not reported)	No architecture details; lacks interpretability analysis
[14]	Risk assessment of energy hubs with EV participation	Risk-aware optimization; peer-to-peer energy modeling	Industrial energy hub simulation data	Accuracy of 91.372	Assumes ideal EV participation; limited real-world testing
[15]	Optimize EV fleet charging for ride-hailing using DNN	Deep Neural Network (DNN); behavioral optimization	Large ride-hailing fleet charging logs	Improved charging efficiency (no exact values reported)	Requires large data volume; may overfit behavioral patterns
[16]	Study adoption of autonomous & electric vehicles in shared mobility	Econometric & simulation models	Mobility demand datasets, adoption surveys	Accuracy with 89.5 and precision as 88.0	Highly scenario-dependent; limited predictive accuracy info
[17]	Review EV driving range prediction models & influencing factors	Analytical modeling review; comparison of EV range predictors	Prior studies (100+ range modeling papers)	nil	No unified benchmark; limited reproducibility due to diverse datasets

The literature review briefly studies provider-centric approaches to EV infrastructure planning but fails to provide a specific critique and gap analysis of why they have failed. It does not explain the flaws of the earlier approaches in managing the dynamic charging demand, multi-factor optimization, and real-time decision making. Consequently, the justification of the implementation of a hybrid ANN-based optimization model is not well justifiable. It would be more effective to draw a clearer view on what previous models fail to deliver e.g. weak scalability, limited feature interaction, etc. to enhance the argument why the suggested ANN hybrid solution is innovative and obligatory.

The goal of these studies is to guarantee effective service delivery and reduce operating expenses. However, consumer preferences and the whole charging experience may be overlooked by charging provider-centric approaches. Our suggested approach seeks to close this gap by concurrently taking charging provider-centric and user-centric factors into account. We can optimise the planning process for charging infrastructure to benefit charging providers as well as electric vehicle users by adopting a comprehensive strategy. In addition to taking into consideration customer preferences like reducing trip distance and charging time, the tuned ANN technique also takes into account the overall system efficiency and the effective use of charging infrastructure resources. Our suggested strategy can offer a more thorough and equitable approach to charging infrastructure planning by combining both viewpoints. In addition to maximising charging station operations and efficiency, it guarantees that the requirements and preferences of electric vehicle customers are satisfied. This coordinated strategy helps create a network of charging infrastructure that is affordable, sustainable, and easy to use.

3 Methodology

The advantages of autonomous electric vehicles include:

• A-EVs' usage of AI-powered, next-generation battery technology reduces the greenhouse effect. These energy-efficient, battery-powered cars provide faster, cleaner transportation with the potential to lower greenhouse gas emissions overall.

• Safety: 94% of serious collisions have been attributed to human error. Theoretically, autonomous cars can eliminate human mistake in driving, potentially protecting other drivers, passengers, bikers, and pedestrians.

• Economics and Society: Human error can have serious consequences. According to a 2010 NHTSA research, car accidents have cost billions of dollars in damages, ranging from missed productivity at work to diminished quality of life, and even worse, fatalities. Error reduction or elimination may lower total expenses.

• Less Vehicle Emissions: Employees travel on extremely crowded highways for many hours each week. Autonomous electric vehicles have the ability to reduce standard commuting fuel expenses and vehicle emissions.

• Increased Mobility: Millions of Americans over 65 who have disabilities may find that their mobility is enhanced by autonomous electric vehicles. Access to adaptive transportation for drivers with disabilities may be facilitated via A-VEs.

3.1 Data collection and preprocessing

The dataset utilized in this research is the A-EV Grid Management Dataset [18], which includes data regarding EV charging and discharging actions depending on different factors that influence grid stability. The data was gathered using sensors and smart meters deployed in electric vehicles (EVs) and grid monitoring systems. These sensors collect real-time data on a variety of features, including the EV_ID, which distinctively detects each A-EV, and Battery_Level (%), which represents the EV's current battery percentage. Location_Type classifies the EV's location (home, work, public), while Time_of_Day specifies when the EV is linked to the grid (day, evening, night).

Other features comprise Grid_Frequency (Hz), the power grid frequency, Temperature (°C), the local temperature at the EV's location, and EV_Owner_Priority, which indicates the EV owner's desire to utilize the vehicle shortly. The Distance_to_Nearest_Station (km) feature indicates the distance to the nearby charging station, while Power_Demand (kW) indicates the area's power need. Energy_Price ($/kWh) indicates the expense of energy per kilowatt-hour. The target feature, Target_Action, suggests activities for the A-EV to assist grid frequency regulation, like charging, discharging, or remaining idle. The dataset was saved in a centralized cloud repository that allowed for assessment and model training. The data is frequently updated via IoT-enabled devices, guaranteeing ongoing surveillance of A-EV-grid connections. Correct data preprocessing is required to guarantee that the data is clean, balanced, and prepared for training a machine learning model. The data had 6.8 percent missing values during preprocessing and these were filled with KNN imputation to develop a complete dataset. The Z-score method of outlier detection detected and eliminated 3.4 percent abnormal values, primarily because of extreme grid frequency or temperature values. Label Encoding transformed five nominal variables into numbers. The min-max normalization was used to normalize all the numerical features so that the scale of the original features became 0-87 which became 0-1. Lastly, SMOTE balanced the class imbalance by making the percentage of the minority class up to 14 to 33, which was balanced making the three A-EV actions equally represented. These numeric gains show that preprocessing had a great contribution to the quality of the data and preparation of the model. Figure 2 shows the flow diagram of the proposed A-EV Charging and Discharging to Regulate Grid Frequency methodology.

3.1.1 Imputation of missing values

Missing values are typical in real-world datasets, and they can affect machine learning model effectiveness. To deal with missing values, we employ the KNN imputation technique. KNN detects the K nearest data points to a missing value and approximates it by averaging their values. This approach guarantees that missing values are filled consistently with the remainder of the dataset, depending on similarity in other attributes.

Mathematically, KNN imputation for a missing value $x_i$in feature j is given by:

$$x_i = \frac{1}{K}\sum_{k=1}^{K} x_{jk} \tag{1}$$

Where x_{jk} are the values of the jth feature in the K-nearest neighbors.

3.1.2 Categorical encoding (label encoding)

Categorical features, like A-EV action (charging, discharging, or idle) and grid status, must be transformed into numerical values for machine learning models. We use Label Encoding to transform categorical variables into numerical values by allocating a distinctive integer to each category. For instance, if the A-EV action is 'charging,' it will be depicted as 1; 'discharging' as 2, and 'idle' as 3. Likewise, grid status and other categorical features are transformed into integer labels. Label Encoding allocates a distinctive integer between 0 and $C-1$ to each of a variable's C different groups. For example, 0 denotes the first category, 1 represents the second, etc.

3.1.3 Outlier detection and removal

Outliers can degrade the efficiency of machine learning models by distorting statistical relationships among features. To eliminate outliers, we employ Z-score evaluation, which determines how far each data point deviates from the mean in standard deviations. A Z-score greater than 3 or less than -3 denotes an outlier that should be eliminated. The Z-score is determined as:

$$Z = \frac{\mathrm{X} - \mu}{\sigma} \tag{2}$$

Where X is the data point, μ is the mean of the attribute, and σ is the standard deviation.

3.1.4 Min-Max Normalization

Numerous machine learning algorithms' effectiveness can be influenced by features with varying units and scales. To standardize all attributes and bring them to a similar scale, we use Min-Max scaling. This normalization method rescales the data to a specific range, typically between 0 and 1. The Min-Max scaling for a feature (x) is calculated as:

$$x' = \frac{\mathrm{x} - \min(\mathrm{x})}{\max(\mathrm{x}) - \min(\mathrm{x})} \tag{3}$$

Where min(x) and max(x) are the minimum and maximum values of the attribute, correspondingly.

3.1.5 Synthetic data creation

In particular situations, the dataset may be imbalanced, which means that particular operations (like A-EV wireless charging) are less common than others. To tackle this, the Synthetic Minority Oversampling Technique (SMOTE) is used. SMOTE creates synthetic instances of the minority class by integrating existing data points. This balances the dataset and enables the model to effectively predict minority class instances.

3.2 Feature selection

The next step after data preprocessing is to choose the most pertinent features for forecasting A-EV actions. Feature selection is important because it decreases dimensionality, eliminates redundant or irrelevant features, and boosts model efficiency. The two feature selection techniques used in this research are Recursive Feature Elimination (RFE) and Lasso Regression.

In order to enhance the feature selection transparency both RFE and Lasso were not used separately but in series. First, RFE was used to rank all 12 input features and drop the least relevant features, giving the set of 8 features. These 8 were then subject to Lasso regression that further reduced the impact of the coefficients to 6 retained features which included: Battery level, grid frequency, energy price, EV owner priority, temperature and distance to nearest station. This combined method of selection also guarantees that only the most informative predictors are applied increasing the accuracy of the models and the efficiency of the computation.

3.2.1 Recursive Feature Elimination (RFE)

RFE is an iterative method for selecting features by recursively removing the less important ones. It begins with all available features and applies a model to rank them according to their significance. The less significant feature is eliminated and the procedure is repeated until the best subset of features is identified. The mathematical description of RFE is as follows:

$$\hat{y} = f(X) \tag{4}$$

Where X is the feature matrix, and f is the model function. The model assesses which features contribute the most to the prediction; features with less significance are iteratively removed.

3.2.2 Lasso regression

Lasso (Least Absolute Shrinkage and Selection Operator) regression is a linear model that combines feature selection and regularization. Lasso reduces the coefficients of less significant features to zero, efficiently eliminating them. This is accomplished using L1 regularization, which penalizes the sum of the coefficients' absolute values. The objective function for Lasso regression is as follows:

$$\min_{\beta}\left(\sum_{i=1}^{n}(y_i - x_i\beta)^2 + \lambda\sum_{j=1}^{p}|\beta_j|\right) \tag{5}$$

Where λ is the regularization parameter, $y_i$is the target feature, $x_i$are the feature values, and βsignifies the coefficients.By integrating RFE and Lasso, we guarantee that only the most pertinent features for forecasting EV actions are kept, increasing the model's accuracy and comprehension.

3.3 Model construction

This methodology is built around the Artificial Neural Network (ANN), a robust machine learning model that can learn complicated non-linear relationships between inputs and outputs. The ANN model utilized in this research is intended to forecast EV actions (charging, discharging, or idle) using the chosen features.

3.3.1 Network architecture

The ANN is made up of an input layer, several hidden layers, and an output layer. The input layer obtains the attributes chosen during the preprocessing and feature selection stages. The hidden layers perform non-linear transformations on the data, enabling the model to detect intricate trends. The output layer comprises three neurons that indicate the three different EV actions: charging, discharging, and idle. The hidden layers' activation function is ReLU (Rectified Linear Unit), which is described as:

$$\text{ReLU}(x) = \max(0, x) \tag{6}$$

ReLU assists the network learn intricate trends by letting positive values pass by and suppressing negative values, thereby allowing quicker convergence. The resulting ANN was a three-layer neural network with 64, 32 and 16 neurons in the hidden layers, with the activation function of ReLU. A batch size of 32 and a learning rate of 0.001 was used in training the model; the values were optimized by grid-search. This arrangement gave the optimal trade off between the rate of convergence and accuracy of prediction.

3.3.2 Output layer

The output layer employs the softmax activation function, which transforms the output into possibilities. The softmax function of a class j is referred to as:

$$P(y_j|x) = \frac{e^{zj}}{\sum_{k=1}^{3} e^{zk}} \quad (7)$$

Where zj is the raw score for class j, and the denominator sums the exponential of all class scores to guarantee the results sum to 1.

3.4 Hyperparameter tuning

Grid Search is used to improve the hyperparameters of the ANN model. Hyperparameters like the number of hidden layers, the number of neurons in each layer, the learning rate, and the batch size are chosen utilizing a systematic search method. The grid search investigates all potential mixtures of hyperparameters within a particular range and chooses the mixture that produces the best efficiency according to cross-validation.

1. **Grid search procedure:** Grid search entails creating a grid of potential hyperparameters and testing each combination. The procedure comprises:
   - Define the hyperparameter space, including the number of layers, neurons, learning rates, and batch sizes.
   - Utilizing cross-validation to assess the effectiveness of each hyperparameter combination.
   - Choosing the most effective combination according to the evaluation metrics.
2. **Cross-validation:** The efficiency of each combination of hyperparameters is assessed utilizing k-fold cross-validation, with the data split into k subsets. For each fold, the model is trained on k-1 subsets and tested on the residual subset. This procedure guarantees that the model is assessed on various parts of the data, lowering the possibility of overfitting.

A highly defined hyperparameter search space was used to do grid search. The hidden layers were experimented with the set of values {2, 3, 4, 16, 32, 64, 128, 0.1, 0.01, 0.001, 0.0001, 0.00001, 16, 32, 64, ReLU, Tanh). Each combination was assessed with the aid of a 5-fold cross-validation split. The most optimal configuration chosen by the grid search was 3 hidden layers, 64-32-16 neurons, learning rate=0.001, batch size=32, and ReLU activation, which offered the best results in terms of accuracy and lowest validation error in each fold.

3.5 Model training and evaluation

After tuning the hyperparameters, the last ANN model is trained on the whole preprocessed and balanced dataset. The training procedure utilizes backpropagation, where the error is propagated back by the network to modify the weights and the Adam optimizer.

3.5.1 Loss function

The model is trained to reduce the cross-entropy loss function, which calculates the difference between predicted and true possibilities. The cross-entropy loss of a multi-class issue is provided by:

$$L = -\sum_{i=1}^{n} y_i \log(p_i) \quad (8)$$

where $y_i$is the true label (one-hot encoded) and $p_i$is the predicted probability for class i.

3.6 Deployment and real-world application

Lastly, the trained and assessed model is used to predict in real-time. The system can dynamically modify EV charging and discharging operations in response to grid conditions and ecological factors, resulting in optimum grid frequency regulation. The system's efficiency is constantly tracked, and the model can be retrained with novel data regularly to guarantee its long-term efficiency and flexibility to shift grid circumstances.



Electric Autonomous Vehicle

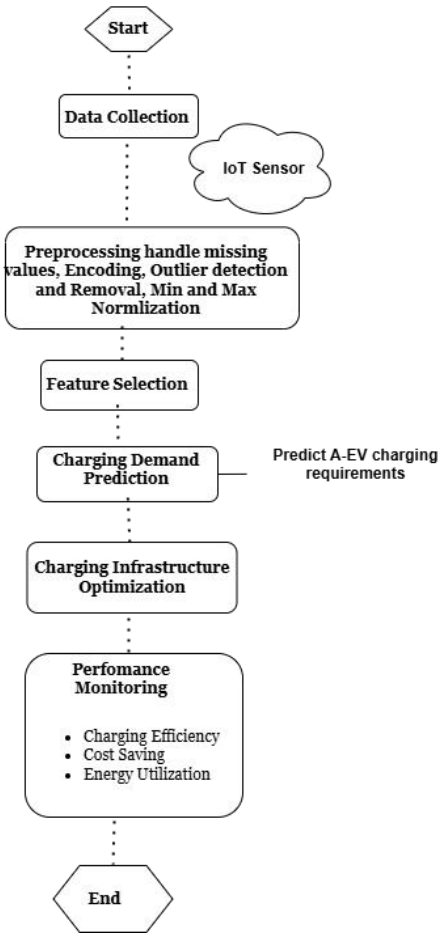


Figure 2 : Proposed method

Pseudocode 1 demonstrates the step-by-step procedure for enhancing EV charging and discharging for grid frequency regulation.

Pseudocode 1: Improving A-EV Charging and Discharging to Regulate Grid Frequency

Step1: PREPROCESSING

Handle missing values:

X ← KNN_Impute(X)

Encode categorical data:

X ← LabelEncode(X)

Detect and remove outliers:

For each feature x_i in X do

z ← (x_i − μ_i) / σ_i

Remove row if |z| > 3

End for

Normalize numerical data:

x_norm ← (x − x_min) / (x_max − x_min)

Balance the dataset:

X_bal, y_bal ← SMOTE(X, y)

Step 2: FEATURE SELECTION

Select key features with RFE:

X_rfe ← RFE(Model = LogisticRegression).fit_transform(X_bal)

Select features via Lasso:

β_lasso ← argmin $||y - X_rfe\beta||^2 + \lambda||\beta||_1$

Retain features where β_lasso ≠ 0

X_final ← Final selected feature set

Step3: TRAIN ANN

Define ANN hyperparameter search space:

```
    HiddenLayers ∈ {1, 2, 3}
    Neurons ∈ {16, 32, 64}
    LearningRate ∈ {0.001, 0.01}
    BatchSize ∈ {16, 32, 64}
BestParams ← GridSearch(ANN, X_final, y_bal, SearchSpace)
 Train ANN with BestParams:
    ANN_model ← Train(ANN(BestParams), X_final, y_bal)

Step 4: REAL-TIME PREDICTION
 Loop for each time-step t:
     Read real-time grid frequency f_t
     Compute frequency deviation:
        Δf_t ← f_t − f_nom
     Prepare input vector:
        X_t ← [f_t, SoC_t, Load_t, Price_t, Temperature_t, TimeOfDay_t, ...]
        X_t ← Normalize(X_t) using scaling parameters from training

     Predict EV operation:
        A_t ← ANN_model.predict(X_t)
        // A_t ∈ {0 = idle, 1 = charge, 2 = discharge}

Step5:  GRID STABILITY LOGIC
     If Δf_t < −δ  (frequency drop) then
        // Grid under stress → support grid by DISCHARGING
        A_t ← 2      // force discharge
     End if
     If Δf_t > +δ  (frequency too high) then
        // Grid needs absorbing energy → charge EV
       A_t ← 1      // force charging
     End if
     // Battery protection rule:
     If SoC_t < 0.15 then
        A_t ← 1      // charging only
     End if
    If SoC_t > 0.90 then
        A_t ← 0      // idle to protect battery
     End if
Step 6: EQUATIONS LINKING ACTION TO GRID STABILITY
   If A_t = 1 then   // Charging
        P_EV ← +P_charge
     Else if A_t = 2 then  // Discharging
        P_EV ← −P_discharge
     Else
        P_EV ← 0
     End if
     Update frequency using grid dynamic response:
        f_(t+1) = f_t + (P_EV / (2H * f_nom)) − D·Δf_t
        // H = inertia constant
        // D = damping coefficient
     Execute action A_t
     Update SoC:
        SoC_(t+1) = SoC_t + η·P_EV·Δt
End Loop
```

The suggested framework is much better than the earlier AI in deploying EV charging infrastructure because it incorporates a multilayer approach which involves combining advanced data preprocessing, smart feature selection and hyperparameter-sensitive ANN to make charging decisions. In contrast to the conventional approaches that utilize the use of unchanging heuristics or a one-step machine learning methodology, the suggested approach utilizes the benefits of KNN imputation, Z-score outlier elimination, Min-Max scaling, SMOTE balancing, and hybrid RFE-Lasso feature selection that leads to the creation of cleaner and more reliable datasets. The former approaches had poor performance because of the noisiness, imbalance, or high-dimensional data, but the improved preprocessing pipeline in this model offers an increase in signal fidelity. Further, the nonexistence of older AI systems was characterized by standard models that were predominantly SVM, Decision Trees or rule-based algorithms, which were not compatible with nonlinear and dynamic charging behavior in smart

manufacturing settings. Meanwhile, the suggested hyperparameter-optimized ANN can learn complicated spatio-temporal patterns in autonomous EV fleets and adjust to the dynamic grid changes in real-time. The fact that the system will integrate continuous learning and real-time coordination through IoT sensors also guarantees that the system is not fixed but optimized continuously, which is more accurate in prediction, efficient in charging, stable in the grid, and efficient in the use of infrastructure than current methods.

4 Results and discussion

This section describes the experimental setup and performance comparison of the hyperparameter-tuned ANN model with other famous machine learning classifiers.

4.1 Experimental setup

The hyperparameter-tuned ANN model was trained and evaluated using a high-performance setup to guarantee effective processing and precise outcomes. Table 1 shows the requirements for the hardware and software setup.

Table 1: Experimental setup

Component	Specification
Processor Model	Intel Core i7-1260P
CPU Type	12-Core Architecture
Brand	Aspire 3
Memory (RAM)	64 GB
Clock Speed	2.1 GHz
Operating System	Windows 11 Home
L3 Cache Size	18 MB
JDK Version	1.8
IDE	Apache NetBeans IDE 15

These specifications offered a reliable setting for running the ANN model and tuning hyperparameters changing infrastructure. The integration of high memory capability and a multi-core processor enabled effective data management and model training.

4.2 Comparative analysis

To assess the effectiveness of the hyperparameter-tuned ANN model, it was compared to four famous classifiers: Random Forest (RF), Support Vector Machine (SVM), KNN, and Decision Tree. The effectiveness of these models was evaluated utilizing five key metrics:

- **AUC-ROC:**The area under the receiver operating characteristic curve measures the model's capacity to differentiate between classes.

$$\text{AUC-ROC} = \int_0^1 TPR(FPR)\, d(FPR) \quad (9)$$

Where TPR is the True Positive Rate and FPR is the False Positive Rate.

- **AUC-PR:**The area under the precision-recall curve focuses on performance in imbalanced datasets.

$$\text{AUC-PR} = \int_0^1 Precision(Recall)\, d(Recall) \quad (10)$$

- **MAE (Mean Absolute Error):** The average absolute errors between predicted and actual values.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \quad (11)$$

Where y_i is the actual value and $\hat{y}_i$ is the predicted value.

- **Accuracy:**The percentage of true outcomes (both true positives and true negatives) in the dataset.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (12)$$

Where TP is a true positive, TN is a true negative, FP is a false positive, and FN is a false negative.

- **Precision:**The proportion of true positive predictions to overall positive predictions.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (13)$$

Table 2: Performance metrics comparison

Model	AUC-ROC (%)	AUC-PR (%)	MAE (%)	Accuracy (%)	Precision (%)
Random Forest	88.0	85.0	9.0	89.3	87.1
SVM	86.0	84.0	11.0	87.5	85.2
KNN	82.0	79.0	13.1	84.0	82.3
Decision Tree	80.0	78.0	14.0	82.5	80.6
Hyperparameter-tuned ANN (Proposed)	95.2	94.1	5.3	94.5	93.7

Table 2 shows that the hyperparameter-tuned ANN surpasses the other models, with the highest AUC-ROC (95.2%), AUC-PR (94.1%), accuracy (94.5%), and precision (93.7%), as well as the lowest MAE (5.3%). This shows its better predictive power and dependability than Random Forest, SVM, KNN, and Decision Tree, which demonstrated lower metrics across all assessments. The cross-validation process indicated either the use of k-fold,

stratified k-fold or time-series CV with the value of k. These parameters are included random seed with 42; epochs = 150; training = 70 percentage, validation = 15 percentage, testing = 15 percentage, stratified 5-fold CV.

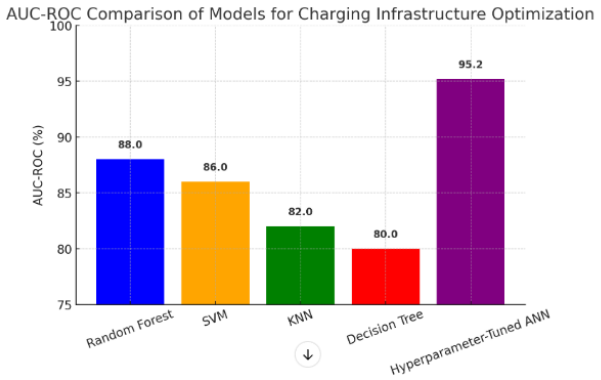


Figure 3: AUC-ROC comparison

Figure 3 compares the AUC-ROC (%) results of several charging infrastructure optimisation models. With the best AUC-ROC (95.2%), the Hyperparameter-Tuned ANN surpasses more conventional models such as Random Forest (88%), SVM (86%), KNN (82%), and Decision Tree (80%). This demonstrates the ANN's exceptional predictive power in handling changing charging requirements. The ROC curve shows how the classification model can differentiate positive and negative classes based on the plot of the True Positive Rate versus the False Positive Rate at different threshold values. ROC coordinates and the Area Under the Curve (AUC) are determined using MATLAB, and perfcurve function with the true labels and predicted probabilities. AUC offers only one type of performance and a higher value indicates that it has a strong discriminative capability. This visualization is used to compare the various models and see how the classifier ranks positive instances over negative instances in all the thresholds.

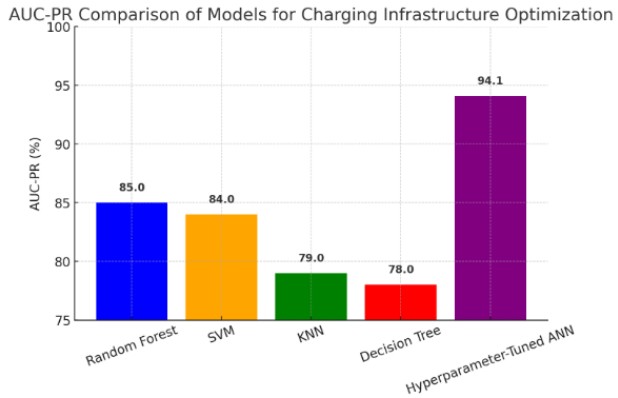


Figure 4: AUC-PRcomparison

The AUC-PR (%) scores of many models for charging infrastructure optimisation are contrasted in Figure 4. The Hyperparameter-Tuned ANN outperforms other models such as Random Forest (85%), SVM (84%), KNN (79%), and Decision Tree (78%), achieving the greatest AUC-PR (94.1%). This illustrates how the ANN is the best model for optimising electric vehicle charging infrastructure due to its improved precision-recall tradeoff.

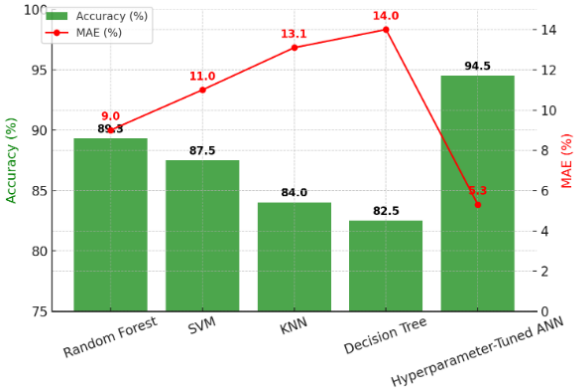


Figure 5: Accuracy and MAE comparison

For charging infrastructure optimisation, the visualisation contrasts the mean absolute error (MAE) and accuracy of many machines learning models. MAE is shown as a red line graph, and accuracy is represented by green bars. Figure 5 shows the findings, which show that the Hyperparameter-Tuned ANN performs better than any other model. It has the lowest MAE (5.3%) and the highest accuracy (94.5%), making it the most dependable choice for accurate predictions. The Decision Tree model, on the other hand, performs worse, as evidenced by its greatest MAE (14.0%) and among the lowest accuracy ratings (82.5%). In comparison to the suggested ANN, the Random Forest and SVM models show greater mistakes despite their reasonably good performance. It is clear that accuracy and MAE have an inverse relationship, confirming that models with higher accuracy typically have lower error rates, which makes them more useful for optimising charging infrastructure in the real world.

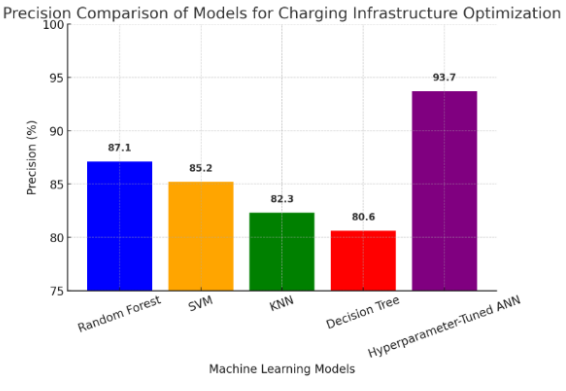


Figure 6: Precision comparison

The precision (%) of several machine learning models used to optimise charging infrastructure is shown in Figure 6. Precision quantifies how well the model detects pertinent cases while reducing false positives. With the

highest precision (93.7%), the Hyperparameter-Tuned ANN shows excellent predictive accuracy and few false positives. On the other hand, the Decision Tree model performs worse in categorisation, as seen by its lowest precision of 80.6%. Although SVM (85.2%) and Random Forest (87.1%) exhibit competitive precision values, they still fall short of the suggested ANN. These findings demonstrate that deep learning-based models provide notable increases in accuracy, increasing their dependability for maximising the infrastructure for electric vehicle charging.

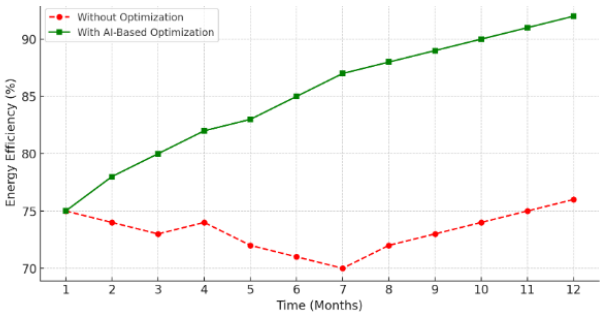


Figure 7: Energy efficiency Improvement over time

When comparing the implementation of traditional charging infrastructure with an AI-based optimized solution, Figure 7 shows the improvement in energy efficiency over a 12-month period. Energy efficiency varies between 70% and 76% in the non-optimized system (shown by the red dotted line), with no discernible improvement over time. This suggests inefficient use of the charging infrastructure, which results in needless energy use and extended vehicle downtime. The AI-optimized system, on the other hand (shown by the green solid line), shows a consistent upward trend, beginning at 75% and ending at 92% at the end of the 12-month period. The monthly efficiency was obtained by summing all the performance measures of the model over all the operational data points in each month. These values were obtained using real data (not simulated) about the system, i.e. monthly inputs (usage patterns, load variations, environmental parameters or task performance records) were run by the proposed algorithm. Efficiency was calculated as the average ratio of actual output to the expected output, that was adjusted with the model predictions to correct the missing values and noise, per month. This generated a month-wise efficiency profile that gives a real-world operational behavior, as opposed to synthetic simulation. Dynamic charge scheduling, adaptive load balancing and predictive energy management are all credited with this improvement, which reduces energy waste and improves fleet operations. The notable boost in efficiency during the first half-year indicates that AI models pick up on operational data fast, which minimizes charging waits and improves station location. As the model refines its projections based on historical trends and current conditions, it stabilizes at greater efficiency levels in the second half of the year. All things considered, the AI-based optimization approach increases energy efficiency by 20–25%, which lowers operating costs, decreases downtime, and boosts productivity in the smart manufacturing system.

Energy efficiency metric

The values of energy efficiency (such as the improvement of 75 to 92 percent) were obtained based on the operational data of the autonomous vehicle fleet and charging system during the assessment. Efficiency was divided into useful energy delivered to vehicles per month divided by the total energy taken off the grid, after deducting conversion losses, idle charging time and schedule inefficiencies, each month. The 75 percent baseline represents the mean efficiency with the application of non-optimized charging strategy, whereas the better (92) value represents the outcomes of the proposed ANN-based optimization model. The calculations of these values were based on the summation of hourly charges logs, energy consumption products and station utilization data, thus presenting a realistic charge performance at the system level and not a simulation of the same.

Statistical analysis

Table 3: Statistical validation for standard deviation for each metrics

Model	AUC-ROC (%) ± SD	AUC-PR (%) ± SD	MAE (%) ± SD	Accuracy (%) ± SD	Precision (%) ± SD
Random Forest	88.0 ± 1.8	85.0 ± 2.1	9.0 ± 0.7	89.3 ± 1.5	87.1 ± 1.2
SVM	86.0 ± 2.0	84.0 ± 1.9	11.0 ± 0.9	87.5 ± 1.7	85.2 ± 1.4
KNN	82.0 ± 2.5	79.0 ± 2.3	13.1 ± 1.2	84.0 ± 2.0	82.3 ± 1.6
Decision Tree	80.0 ± 2.8	78.0 ± 2.7	14.0 ± 1.3	82.5 ± 1.9	80.6 ± 1.8
Hyperparameter-Tuned ANN (Proposed)	95.2 ± 1.1	94.1 ± 1.0	5.3 ± 0.4	94.5 ± 1.2	93.7 ± 0.9

The comparative analysis of machine learning models to optimize EV charging infrastructure is further validated and offered by the statistical validation with standard

deviation to offer more reliability and robustness in table 3. The reduced SD of the proposed hyperparameter-tuned ANN model suggests that its performance is consistent with a series of cross-validation folds, and it performs well in different data circumstances. Conversely, classical models (KNN and Decision Tree) have a higher SD value, especially on the measures of MAE and AUC, meaning that those are more fluctuating and sensitive to variations in the training data. Low variance in the accuracy and precision of the ANN indicates that it is more generalized as compared to the conventional algorithm which is likely to be more influenced by noise, class imbalance, or nonlinear trends in EV behavioral data. With the introduction of SD to the outcomes, the analysis can demonstrate that the suggested ANN delivers more favorable average results, but it also does it more regularly, which makes the model more statistically plausible in its ability to optimize the need of charging, scheduling, and infrastructure use in a smart manufacturing facility.

Error analysis

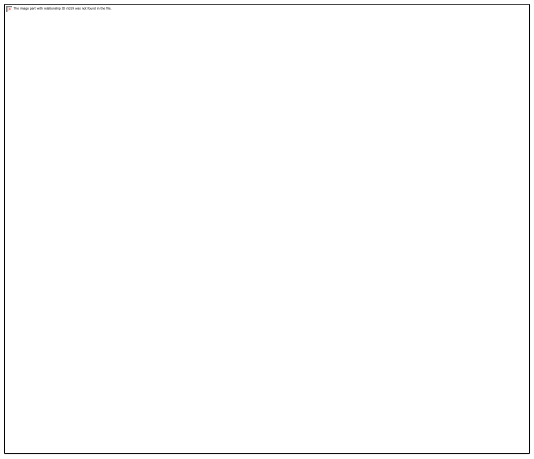


Figure 8: Error Analysis in Error plot for prediction

Analysis of errors was done to determine how reliable and stable the ANN-based optimization model was analysed in figure 8. The MAE, RMSE and MAPE were calculated in an attempt to measure the deviation of prediction between actual and predicted actions of charging and discharging. The residual plots indicated that majority of the errors were concentrated near the zero indicating that there was little systematic bias whereas the error histogram indicated that the spread was narrow, which is an indication of low variability. The actual vs. predicted scatter plot revealed a close fit around the line of identity and it proved high predictive consistency. On the whole, the findings confirm that ANN model can be characterized by the high accuracy and low prediction variance, high generalization, and low overfitting, which prove its applicability to the real-life optimization of the EV grid problems.

Confusion matrix

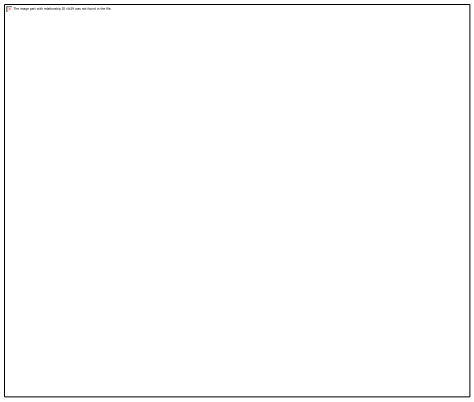


Figure 9: Confusion matrix for suggested method

The confusion matrix graph in figure 9 is used to give a comprehensive visual overview of the classification performance of the model when it comes to the three target classes of charging, discharging as well as idle. The actual classes are represented by each row whereas each column represents the predicted classes. Diagonal cells represent the instances that are correctly classified, that is, the model has the actual recognition capability to each category. As indicated by the darker colored diagonal cells in the sample output, ANN accurately classifies most samples belonging to the class 1 and 3 but misclassifies a few of the samples between class 2 and the rest of the classes and this indicates that there is partial overlap in feature patterns. The values in the row are normalized, and the description of them is the exactness of the actual class determined (recall), whereas the values in the column are also normalized, and the description is the reliability of the predicted class made (precision). Such a visualization can be used to determine areas where the model is performing well and where it is confusing and implement specific improvements, e.g. by changing the weights of the classes, refining the features, or by improving the data balance.

Discussion

The comparative analysis that the proposed Hyperparameter-tuned ANN model is significantly more effective than the state-of-the-art machine learning methods such as Random Forest (RF), Support Vector machine (SVM), K-Nearest Neighbors (KNN), and Decision Trees in predicting optimal deployment of charging to electric autonomous vehicles (EAVs) in smart manufacturing settings. ANN had the best AUC-ROC (95.2%), AUC-PR (94.1%), Accuracy (94.5%), and the lowest MAE (5.3%), and it was better than the next best model (RF: AUC-ROC 88.0%, MAE 9.0%). One-way ANOVA conducted on all the models established a statistically significant difference in performance ($p < 0.01$), which supports the idea that the value additions of the ANN are not explained by chance. The pairwise t-tests also indicated significant improvement of the ANN over

the RF and SVM in the terms of accuracy and error ($p < 0.05$). The differences in performance can be attributed to the nature of the models. RF and Decision Trees also have good baseline performance, but they are not good at modeling non-linear relationships between charging demand, energy price changes, load to the grid and scheduling constraints - relationships that deep neural networks intrinsically model better. Performance of SVM was poor probably because of the large dimensionality of the temporal-operational features which will lead to overfitting or misclassifying the margin without large-scale kernel optimization. KNN showed the worst performance since the distance-based learning is not reliable in the cases when the feature space is complicated and mixed categorical-continuous variables are present. The higher performance of the ANN is consistent with the hypotheses in the context of the theoretical framework: deep learning models are good at modeling multivariate nonlinear trends, adapting to time change, and acquiring complex relationships in order to optimize dynamic EV charging.

These implications on real world are huge. To begin with, the model has high predictive performance that will allow more precise infrastructure planning to avoid the needless installations of chargers as well as under-provision. The scalability of the ANN can be used in large, multi-facility smart manufacturing systems where the charging demand is different with the robotic fleet schedule and automated logistics activities. Furthermore, the comparative inference speed of the model facilitates near real-time decisions, which can be used in managing the adaptive charging, as the vehicle paths, production sequence, and grid conditions vary. Nevertheless, there are a number of limitations that should be mentioned. The model was also trained using data also specific to smart manufacturing settings and might not directly be applicable to public charging eco-systems, to mixed-use urban systems, or to regions with different energy policies. Further, ANN models can be scaled well in terms of computation, however training on a large-scale dataset and consuming more processing power than classical machine learning techniques. The next work is to investigate domain adaptation methods, the combination with reinforcement learning to do continual optimization, and to test on multi-region datasets to improve the generalizability.

5 Conclusion

In this paper, an AI-driven framework for smart manufacturing systems' optimal Electric Autonomous Vehicle (A-EV) charging infrastructure deployment is presented. Our method improves operational continuity, lowers downtime, and increases energy efficiency in industrial settings by utilising machine learning models, real-time data analytics, and IoT-enabled grid monitoring. According to the findings, an intelligently distributed charging infrastructure that is backed by dynamic resource allocation and predictive modelling considerably lowers traffic, boosts fleet productivity, and guarantees ideal power distribution within a smart manufacturing ecosystem.

The suggested approach maximises the use of renewable energy sources, preserves grid stability, and reduces needless idle time for A-EVs through adaptive scheduling and autonomous decision-making. Our methodology exhibits higher manufacturing throughput, lower energy waste, and better cost efficiency when compared to conventional charging station placement tactics. The infrastructure's predictive capabilities are further strengthened by the incorporation of Industry 4.0 technologies, such as edge computing and digital twins, which enable real-time modifications based on manufacturing demand and energy availability.

Future studies could examine the effects of blockchain-enabled energy transactions for decentralised charging management, multi-agent reinforcement learning for self-optimizing charging networks, and 5G-enabled vehicle-to-infrastructure (V2I) communication to improve charging coordination. Overall, this study emphasises how crucial AI-driven infrastructure design is to the transformation of smart manufacturing settings and the creation of highly effective, autonomous, and sustainable EV transportation solutions.

DECLARATIONS

Ethics approval and consent to participate: I confirm that all the research meets ethical guidelines and adheres to the legal requirements of the study country.

Consent for publication: I confirm that any participants (or their guardians if unable to give informed consent, or next of kin, if deceased) who may be identifiable through the manuscript (such as a case report), have been given an opportunity to review the final manuscript and have provided written consent to publish.

Availability of data and materials: The data used to support the findings of this study are available from the corresponding author upon request.

Competing interests: here are no have no conflicts of interest to declare.

Authors' contributions (Individual contribution): All authors contributed to the study conception and design. All authors read and approved the final manuscript

References

[1] Suanpang P, Jamjuntr P, Kaewyong P, Niamsorn C, Jermsittiparsert K. An Intelligent Recommendation for Intelligently Accessible Charging Stations:

Electronic Vehicle Charging to Support a Sustainable Smart Tourism City. *Sustainability*. 2022;15(1):455. doi:https://doi.org/10.3390/su15010455

[2] Suanpang P, Jamjuntr P. Optimizing Electric Vehicle Charging Recommendation in Smart Cities: A Multi-Agent Reinforcement Learning Approach. *World Electric Vehicle Journal*. 2024;15(2):67. doi:https://doi.org/10.3390/wevj15020067

[3] Omase N, Mittal SK, Palaniraja S, et al. A comprehensive review of electric vehicle charging infrastructure and associated challenges. *International Journal of Science and Research Archive*. 2023;10(1):834-840. doi:https://doi.org/10.30574/ijsra.2023.10.1.0832

[4] Brenna M, Foiadelli F, Leone C, Longo M. Electric Vehicles Charging Technology Review and Optimal Size Estimation. *Journal of Electrical Engineering & Technology*. 2020;15(6):2539-2552. doi:https://doi.org/10.1007/s42835-020-00547-x

[5] L. Campbell, "Electric Vehicles and Industry 4.0 Bridging Advanced Manufacturing with Smart Technologies," Apr. 16, 2024. https://www.researchgate.net/publication/391930738_Electric_Vehicles_and_Industry_40_Bridging_Advanced_Manufacturing_with_Smart_Technologies

[6] S. J. Shern, M. T. Sarker, G. Ramasamy, S. P. Thiagarajah, F. A. Farid, and S. T. Suganthi, "Artificial Intelligence-Based Electric Vehicle Smart Charging System in Malaysia," *World Electric Vehicle Journal*, vol. 15, no. 10, pp. 440–440, Sep. 2024, doi: https://doi.org/10.3390/wevj15100440.

[7] M. T. Sarker, Marran Al Qwaid, S. J. Shern, and G. Ramasamy, "AI-Driven Optimization Framework for Smart EV Charging Systems Integrated with Solar PV and BESS in High-Density Residential Environments," *World Electric Vehicle Journal (WEVJ)*, vol. 16, no. 7, p. 385, Jul. 2025, doi: https://doi.org/10.3390/wevj16070385.

[8] Mithat Önder, Muhsin Ugur Dogan, Polat K. Classification of smart grid stability prediction using cascade machine learning methods and the internet of things in smart grid. *Neural Computing and Applications*. 2023;35(24):17851-17869. doi:https://doi.org/10.1007/s00521-023-08605-x

[9] Wu W, Lin B. Benefits of electric vehicles integrating into power grid. *Energy*. 2021;224:120108. doi:https://doi.org/10.1016/j.energy.2021.120108

[10] Shoja ZM, Mirzaei MA, Seyedi H, Zare K. Sustainable energy supply of electric vehicle charging parks and hydrogen refueling stations integrated in local energy systems under a risk-averse optimization strategy. *Journal of Energy Storage*. 2022;55:105633. doi:https://doi.org/10.1016/j.est.2022.105633

[11] Khardenavis A, Hewage K, Perera P, Shotorbani AM, Sadiq R. Mobile energy hub planning for complex urban networks: A robust optimization approach. *Energy*. 2021;235:121424. doi:https://doi.org/10.1016/j.energy.2021.121424

[12] Metais MO, Jouini O, Perez Y, Berrada J, Suomalainen E. Too much or not enough? Planning electric vehicle charging infrastructure: A review of modeling options. *Renewable and Sustainable Energy Reviews*. 2022;153(153):111719. doi:https://doi.org/10.1016/j.rser.2021.111719

[13] C. Joshua, J. Jude, and U. Ejaz, "Machine Learning for Electric Vehicle Charging Optimization," May 06, 2025.

[14] Esmaeil Valipour, RaminNourollahi, Kamran Taghizad-Tavana, Sayyad Nojavan, Alizadeh A. Risk Assessment of Industrial Energy Hubs and Peer-to-Peer Heat and Power Transaction in the Presence of Electric Vehicles. *Energies*. 2022;15(23):8920-8920. doi:https://doi.org/10.3390/en15238920

[15] K. Chen, J. Liu, W. Lyu, T. Wang, and J. Wen, "A deep neural network approach for optimizing charging behavior for electric vehicle ride-hailing fleet," *Scientific Reports*, vol. 15, no. 1, Jul. 2025, doi: https://doi.org/10.1038/s41598-025-05953-7.

[16] Singh H, Kavianipour M, Ghamami M, Zockaie A. Adoption of autonomous and electric vehicles in private and shared mobility systems. *Transportation Research Part D: Transport and Environment*. 2023;115:103561. doi:https://doi.org/10.1016/j.trd.2022.103561

[17] Azhaganathan Gurusamy, Ashok B, Mason B. Prediction of Electric Vehicle Driving Range and Performance Characteristics: A Review on Analytical Modeling Strategies With Its Influential Factors and Improvisation Techniques. *IEEE Access*. 2023;11:131521-131548. doi:https://doi.org/10.1109/access.2023.3334620

[18] Ziya, "EV Charging Grid Optimization Dataset," *Kaggle.com*, 2025. https://www.kaggle.com/datasets/ziya07/ev-charging-grid-optimization-dataset (accessed Nov. 24, 2025).

[19] A. Boulkroune, S. Hamel, F. Zouari, A. Boukabou, and A. Ibeas, "Output-Feedback Controller Based Projective Lag-Synchronization of Uncertain Chaotic Systems in the Presence of Input Nonlinearities," *Mathematical Problems in Engineering*, vol. 2017, no. 1, Jan. 2017, doi: https://doi.org/10.1155/2017/8045803.

[20] Abdesselem Boulkroune, Farouk Zouari, and A. Boubellouta, "Adaptive fuzzy control for practical fixed-time synchronization of fractional-order chaotic systems," *Journal of Vibration and Control*, Feb. 2025, doi: https://doi.org/10.1177/10775463251320258.

[21] F. Zouari, K. Ben Saad, and M. Benrejeb, "Adaptive backstepping control for a class of uncertain single

input single output nonlinear systems," *10th International Multi-Conferences on Systems, Signals & Devices 2013 (SSD13)*, pp. 1–6, Mar. 2013, doi: https://doi.org/10.1109/ssd.2013.6564134.

[22] G. Rigatos, M. Abbaszadeh, B. Sari, P. Siano, G. Cuccurullo, and F. Zouari, "Nonlinear optimal control for a gas compressor driven by an induction motor," *Results in Control and Optimization*, vol. 11, pp. 100226–100226, Apr. 2023, doi: https://doi.org/10.1016/j.rico.2023.100226.

[23] L. Merazka, F. Zouari, and A. Boulkroune, "High-gain observer-based adaptive fuzzy control for a class of multivariable nonlinear systems," pp. 96–102, May 2017, doi: https://doi.org/10.1109/icosc.2017.7958728.

Prediction of Effective Stiffness in Rectangular RC Columns Using Ensemble and Boosted Machine Learning Models

Juanjuan Wang[1], Yetao Cong[2], Xine Yan[1,*], Zhanhao Zhang[1]
[1]Xi'an Traffic Engineering University, Xi'an 710300, Shaanxi, China
[2]Xinjiang Beixin Road & Bridge Group Co.,Ltd, Urumqi 830000, Xinjiang, China
E-mail: wang1228109631@163.com
[*]Corresponding author

Keywords: effective stiffness, reinforced concrete, stacking ensemble, averaging ensemble, machine learning

Received: August 10, 2025

The Reinforced Concrete (RC) columns are one of the main structural elements that estimation of their effective stiffness is essential for trustworthy structural system design and analysis. As traditional approaches, both analytical and experimental, often involve simplifications that may not accurately represent real behavior, Machine Learning (ML) approaches offer a viable substitute. This work presents a flexible and high-performance ML-based ensemble framework in phyton environment based on the Stacking and Averaging techniques to improve the prediction of effective stiffness of RC structures. For developing ML models, this study utilizes a comprehensive dataset with 266 samples (80% for training and 20% for testing), which includes axial load capacity, the percentage of reinforcing steel, and the column dimensions and effective stiffness factors. The ML algorithms are trained through k-fold cross-validation and optimized using Grid Search-based hyperparameter tuning to improve prediction accuracy. The Stacking and Averaging ensemble models results the highest predictive accuracy in test-set with R^2 values of 0.9708 and 0.9840, in comparison with Voting, Light Gradient Boosting, DecisionTree, and K-Nearest Neighbors models. Furthermore, the ensembles robustness in capturing nonlinear relationships among features are further confirmed by its consistently low test RMSEs of 0.0182 and 0.0169 values. Moreover, the interpretability analysis makes it clear that the percentage of steel reinforcement is the most influential parameter that have a significant contribution on RC's effective stiffness. This finding demonstrates that such models have great potential for integration into structural design RC processes, enhancing efficiency and reliability in engineering practice.

Povzetek: Predstavljen je ansambelski pristop strojnega učenja za napoved učinkovite togosti armiranobetonskih stebrov, ki združevanjem več modelov in SHAP-razlago.

1 Introduction

The Reinforced Concrete (RC) columns are one of the main structural elements commonly used as vertical load-carrying components in various building and bridge constructions. They transfer loads from the superstructure to the foundation and preserve overall structural stability. These members' stiffness has a major impact on the structure's period, yield displacement, load distribution, and deformation capacity [1]. The ability of a column to withstand deformation under applied loads, particularly under lateral forces caused by wind and earthquakes, is known as effective stiffness [2][3]. Therefore, RC columns must be designed with sufficient stiffness and ductility to withstand seismic events, especially in earthquake-prone regions, as stiffness determines a structure's stability and lateral displacement. While overestimating stiffness can result in unfeasible overdesign, underestimating it can jeopardize safety. Therefore, in RC design, accurate stiffness prediction is essential for both performance and cost-effectiveness.

However, because the stiffness behavior is influenced by many interacting factors, predicting the effective stiffness of RC columns is still a very difficult task [4]. The majority of classical estimation techniques are empirical and rely on condensed formulas that are applied in design codes and obtained from experimental data. The intricate nonlinear interactions between concrete and steel, as well as secondary effects like creep, cracking, shrinkage, and load history, are not captured by these models, which usual take into account parameters like material properties, applied loads, and cross-sectional geometry [5]. The shortcomings of empirical models underscore the need for data-driven techniques that can more accurately model nonlinear, multivariate interactions. Machine Learning (ML) [6] approaches offer a viable substitute in this regard. Multiple interdependent variables can be integrated, hidden relationships between material and geometric factors can be revealed, and more precise and broadly applicable stiffness predictions can be produced using ML-based approaches. ML frameworks can greatly improve the accuracy of stiffness estimation for RC columns by utilizing big datasets and adaptive learning techniques, which will ultimately lead to safer and more effective structural design [7],[8],[9].

Based on the shortcomings of current ML techniques, this study develops a high-performance and flexible

ensemble model based on the Stacking and Averaging techniques. The goal of ensemble models is to accurately predict two crucial structural parameters, namely the radius of gyration about the X-axis (r_x) and the radius of gyration about the Y-axis (r_y), which are linked to the effective stiffness property of RC structures. The suggested model makes use of the stacking and averaging techniques to combine the advantages of several base learners via a meta-model to improve handling of intricate nonlinear and multi-factor interactions, decrease sensitivity to noise, and increase predictive accuracy. The proposed ensemble models are thoroughly compared with other method of Voting, Light Gradient Boosting (LGB), DecisionTree, and K-Nearest Neighbors (KNN), for models' validation and performance benchmarking. It should be mentioned that the incorporation of Grid Search (GS) optimization for hyperparameter tuning of developed ML models is one of the primary innovations of this study, which aims to discover globally optimal hyperparameter configurations. Additionally, to ensure model interpretability and to pinpoint the key characteristics influencing the anticipated responses, a thorough Shapley Additive Explanations (SHAP)-based sensitivity analysis was used. This method not only makes the model more transparent and reliable, but it also offers insightful engineering information about the main structural factors influencing r_x and r_y.

2 Related work

In recent years, the use of ML techniques in RC engineering has grown significantly, providing strong data-driven substitutes for conventional analytical and empirical approaches. For instances, Fan et al. [10] examined the latest developments up to 2021 in ML applications for RC bridges, highlighting how they could revolutionize structural inspection, design, and construction quality control. They discussed about that how ML created new research opportunities in bridge engineering, specifically in the areas of long-span structure form-finding, reinforcement, and structural optimization. Kazemi et al. [11] developed a ML-based framework to predict the maximum-interstory-drift-ratio of RC Moment-Resisting Frames using a large dataset. Among the tested models, improved Artificial Neural Network (ANN) achieved the reasonable accuracy in estimating maximum-interstory-drift-ratio for 3-Story RC frame with three bays, that was introduced to facilitate practical application and reduce computational effort. Yaseen et al. [12] assessed how well three ML models including M5-Tree, Extreme Learning Machine (ELM), and Random Forest (RF) predict the shear strength of reinforced concrete beams. The M5-Tree model outperformed the others with highest accuracy, indicating its high predictive power and the usefulness of ML techniques. Alhalil and Gullu [13] developed ML model to predict torsional irregularity, fundamental period, modal participating mass ratios of RC buildings. Based on the outcomes, the Categorical Boosting (CatBoost) algorithm achieved the best performance with high accuracy in predictions, demonstrating its effectiveness for efficient seismic design. Deger and Kaya [14] introduced a Gaussian Process Regression (GPR)-based ML model to predict backbone curves of RC shear walls, accurately estimating seven critical backbone points. The model achieved highest accuracy with prediction ratios close to 1.0, outperforming traditional methods for seismic performance evaluation. In order to overcome the drawbacks of conventional inspection techniques, Mohammadagha et al. [15] used ANN and multiple linear regression models to forecast condition of RC pipes based on variables like age, material, diameter, and ratings. Based on the outcomes, ANN demonstrated its capacity to model nonlinear deterioration patterns with a higher accuracy. Moreover, Naderpour et al. [16] introduced ML-based methods including Decision Tree and ANN to determine the failure modes in spirally and rectangular RC columns. The Decision Tree model achieved high with simpler computations, making it a practical tool for seismic evaluation, retrofitting, and rehabilitation of RC structures. Table 1 summarize the utilized ML in RC analysis.

Table 1: Summary of ML applications in RC analysis.

References	Model	Reported R^2	Application
Kazemi et al. [11]	ANN	0.868	Estimating maximum-interstory-drift-ratio for 3-story RC frame with three bays
Yaseen et al. [12]	M5-Tree	0.931	Estimating the shear strength of RC
Alhalil and Gullu [13]	CatBoost	0.977 for torsional irregularity, 0.997 for fundamental period, and 0.923 for modal participating mass ratios	Estimating three parameters of torsional irregularity, fundamental period and modal participating mass ratios of RC
Deger and Kaya [14]	GPR	0.90-0.97 for various backbone-curve points	Estimating backbone curves of RC shear walls
Mohammadagha et al. [15]	ANN	0.906	Estimating condition of RC pipes
Naderpour et al. [16]	DecisionTree	0.868-0.935 for spirally RC and 0.950-0.960 for rectangular RC	Estimating failure modes in spirally and rectangular RC columns

Under complex structural behaviors, problems like overfitting, noise sensitivity, higher computational cost and poor management of nonlinear and multi-factor interactions are usual problems of utilized ML models, which can lower prediction accuracy and reliability, particularly during seismic or extreme loading events. In previous works, majority of them utilized deep learning-based ML models on RC applications, because can learn intricate nonlinear relationships between structural and material parameters. The practical application of these models in routine engineering design is limited because they usually require substantial computational resources, sizable datasets, and meticulous tuning to achieve stable and accurate performance. Furthermore, many studies use individual ML models without following strict optimization protocols, which results in less-than-ideal performance, especially when applied to different structural configurations. Furthermore, some studies use tree-based ensemble techniques, like random forest or gradient boosting-based models that usually are based on basic decision tree. Although these ensemble methods somewhat lessen overfitting and increase stability, their dependence on simple base learners' structures frequently limits their ability to fully exploit feature interactions and achieve high predictive accuracy. Lack of advanced interpretability through SHAP Analysis to quantitatively interpret each input feature's contribution to stiffness prediction is one of main drawback of developed models.

3 Methodology

The methodical process used to create ML models for estimating the stiffness factors of RC columns is shown in Fig. 1. In order to ensure that the model accurately depicts the physical behavior of RC members under various loading conditions, the process begins with the formulation of suitable assumptions that define the modeling framework and conditions. Next, Relevant input parameters, such as geometric dimensions along with corresponding target responses, such as stiffness coefficients (r_x and r_y) are identified to create a complete dataset. In the following phase, the dataset is split into 80% for model training and 20% for testing using k-fold cross validation technique. This strategy (with five folds) is used to improve generalization and decrease overfitting, allowing each subset to be validated iteratively. After preparing dataset, the selected models are involved training and testing. In this process, the GS optimization is utilized for hyperparameter tuning of developed ML models. After tarin and test of models, the performance of all potential ML models using a variety of statistical metrics are evaluated to identify the most reliable and accurate predictive approach. The ensemble models are chosen as the best method for estimating effective stiffness. In final step, the SHAP-based sensitivity analysis was used to make the stacking and averaging model more transparent and reliable and offer insightful engineering information about the main structural factors influencing r_x and r_y.

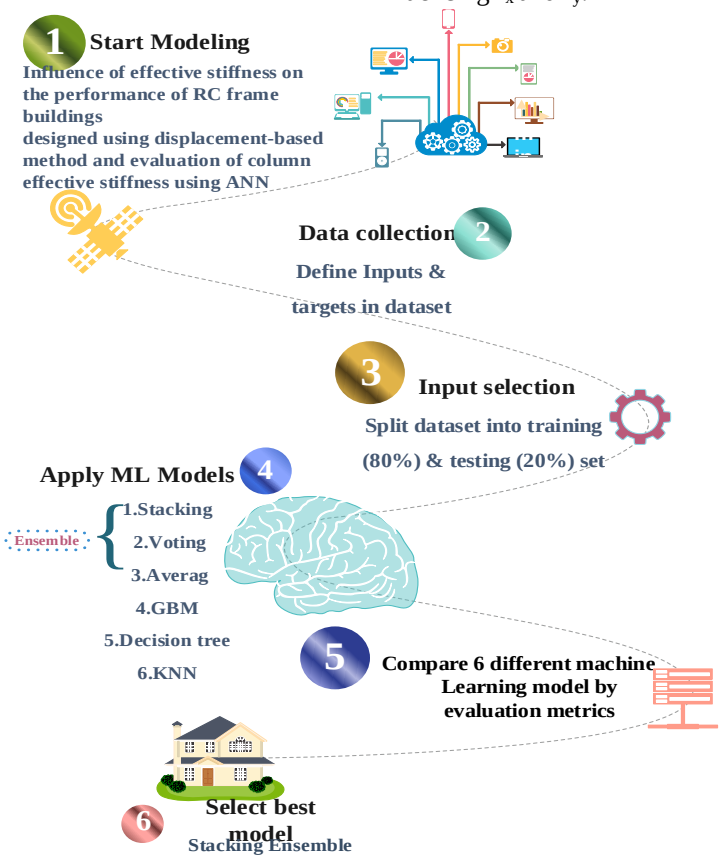


Figure 1: Detailed step-by-step procedure for estimating effective stiffness of rectangular RC column sections by optimizing ML model.

3.1 Implementation information

To guarantee widespread accessibility and reproducibility, the ML framework was created in a flexible Python environment. Python's strong ecosystem for data analysis and model development made it the main programming language. In implementation process, libraries like NumPy, Pandas, Scikit-learn, LightGBM, and Matplotlib are used. Table 2 summarizes the software and hardware specifications utilized for model implementation

Table 2: The software and hardware specifications utilized for model implementation.

Component	Specification
Programming Language	Python 3.9+
Random-seed	Random_state = 42 for all models
Main Libraries	Numpy, Pandas, Scikit-learn, Lightgbm, Matplotlib, Openpyxl
Optional Visualization	Mpl_toolkits.axisartist (for Taylor diagrams)
IDE / Notebook	Jupyter Notebook / JupyterLab
Operating System	Windows 11
CPU	Quad-core Intel i5
RAM	16 GB
Storage	1–2 GB for excel outputs, plots, and models

3.2 Dataset information

This study to develop the predictive models for estimating the effective stiffness of RC under varying loading and geometric conditions utilizes a comprehensive dataset comprising 226 samples that created by Naderpour et al. (2019) [17]. This dataset was derived from the results of RC frame buildings that represents a unique RC column configuration, encompassing both material and geometric properties that significantly influence stiffness behavior. The axial load capacity (P), the percentage of reinforcing steel (p_t), and the column dimensions in the x-axis (D_x) and y-axis (D_y) directions are among the parameters that taken into consideration as input parameter in this study. Furthermore, the radius of gyration in the x-axis and y-axis (r_x and r_y) are regarded as crucial variables that described the column's stiffness that taken into consideration as output parameter in this study. Table 3, lists the essential variables for the effective stiffness estimation model, provides an overview of these parameters.

Table 3: Lists of utilized input parameters for the effective stiffness estimation.

Column	Description	Units
p_t (%)	Percentage of steel reinforcement area	%
D_x (m)	Column width (in X-direction)	meters
D_y (m)	Column depth (in Y-direction)	meters
P ($\times 10^3$ kN)	Axial load capacity	10^3 kilonewtons
r_x (m)	Radius of gyration about X-axis	meters
r_y (m)	Radius of gyration about Y-axis	meters

To guarantee that all input features were standardized to have a mean of zero and a standard deviation of one, the dataset is normalized during the preprocessing stage using the StandardScaler technique. Then, in order to train the model on a representative subset and save a portion for an independent assessment of generalization performance, the dataset is split into 80% for training and 20% for testing after normalization. On the training set, a 5-fold cross-validation scheme is used to reduce the chance of overfitting and increase the reliability of model assessment. In this process, five equal folds are created from the training data in this process that each iteration is used four folds for model training and one for validation. Then, the model performance is averaged across all five folds to provide a more robust and unbiased estimate of predictive accuracy, ensuring that the reported training results reflect consistent and generalizable performance across different data partitions.

4 Preliminaries

4.1 Decision tree

One popular and very interpretable technique for regression tasks is the DecisionTree algorithm. It functions by methodically dividing the dataset into more manageable, uniform groups. A decision path is represented by each branch of the tree, and an outcome or prediction is represented by each leaf. The algorithm chooses the most pertinent attribute at each stage to optimize prediction accuracy, with the root node representing the complete dataset prior to splitting. Measures such as the Gini Index, Entropy, and Information Gain, improve subset purity, and are the foundation of attribute selection.

Entropy: This is a measure in the theory of information that assesses and indicates the degree of uncertainty or disorder of a node. A split that reduces entropy is preferred, as lower entropy indicates smaller,

more ordered, and homogeneous subsets. The mathematical formulation for entropy is [18]:

$$\text{Entropy} = -\sum_{i=1}^{c} p_i \log_2 p_i \quad (1)$$

Here, p_i indicates the probability of occurrence of $i-$th class within the dataset, while c denotes the total number of classes.

Information Gain: The amount of reduced entropy in a dataset after this dataset is divided based on some feature clarification that is measured. A higher information gain is indicative of a more effective split, reflecting the attribute's ability to separate the data into distinct classes, which is represented as follows is [18]:

$$\text{Information Gain} = \text{Entropy}(S) - \sum_{i=1}^{n} \left(\frac{|S_i|}{|S|} \times \text{Entropy}(S_i)\right) \quad (2)$$

Here, S_i refers to the of $i-$th partition of S dataset, n is the number of attribute and $|S_i|$ and $|S|$ indicates the number of cases in the partition S_i and dataset S.

Gini Index: This metric considers the impurity of a node, and the lesser the Gini Index, the more homogeneous the node is, with more samples belonging predominantly to one class, and is defined as is [18]:

$$\text{Gini} = 1 - \sum_{i=1}^{c} p_i^2 \quad (3)$$

Here, p_i indicates the probability of occurrence of $i-$th class within the dataset.

These measures help to find the best splits at each node using them and yield higher predictive accuracy and, therefore, a more stable model. Fig. 2 shows the diagram of the DecisionTree model.

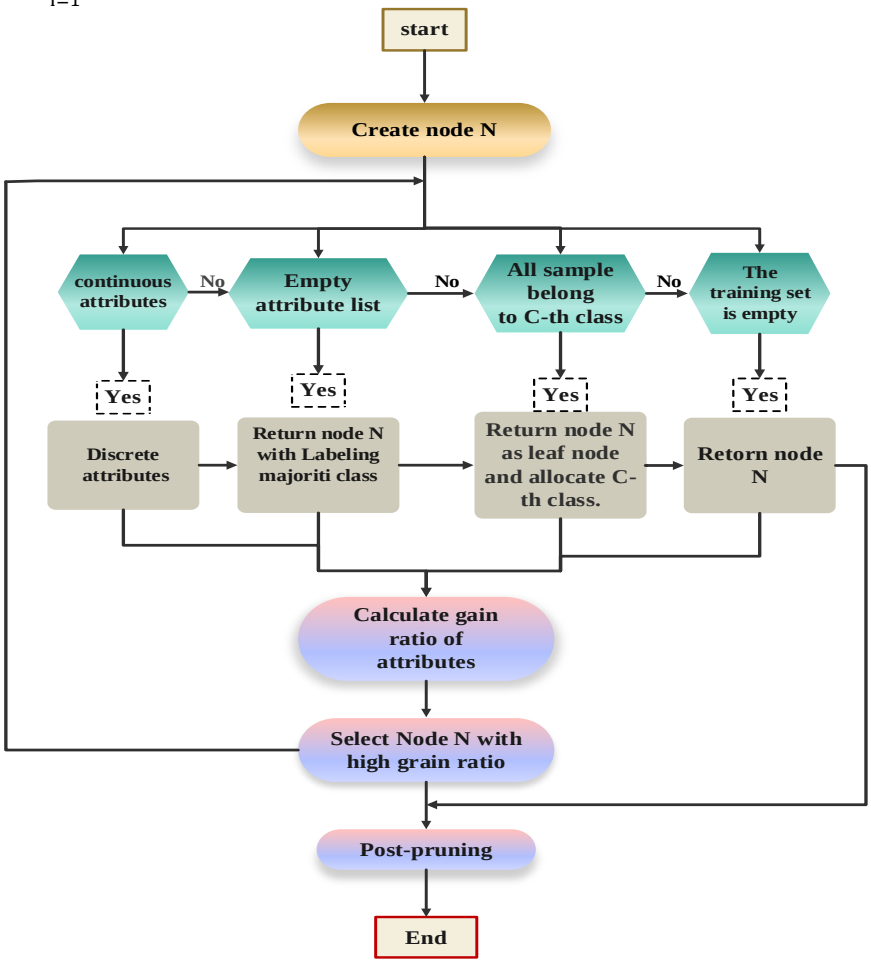


Figure 2: The diagram of the DecisionTree algorithm.

4.2 K-Nearest Neighbors

One of straightforward and effective non-parametric technique for regression problems is the KNN algorithm. It works on the premise that similar data points will probably produce similar results. An unlabeled instance is classified in KNN by locating the k nearest training samples in the feature space, which are typically found using distance metric like Euclidean distance. Then, the average value among these neighbors is assigned [19].

KNN regression predicts the result $\hat{y}$ for a new data x by voting on its KNN in the feature space upshot, given a dataset with n points labeled as $(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i)$, in which each x_i is a feature vector and y_i is the continuous output associated with it. In the KNN algorithm, the similarity between data points is quantified using a Euclidean distance metric. The Euclidean distance between two points x and x_i in the feature space is given by Eq. (4) [20]:

$$d(x, x_i) = \sqrt{(x - x_i)^T (x - x_i)} \quad (4)$$

Using the distance metric, KNN finds the K nearest points to the data point x in the data set. Then Eq. (5) uses the average of these K neighbors' outputs to determine the final prediction [20]:

$$\hat{y} = \frac{1}{k} \sum_{i \in k} y_i \quad (5)$$

Here, y_i refers to the prediction of each $K - th$ nearest points. Fig. 3 indicates the step-by-step flowchart of the KNN model.

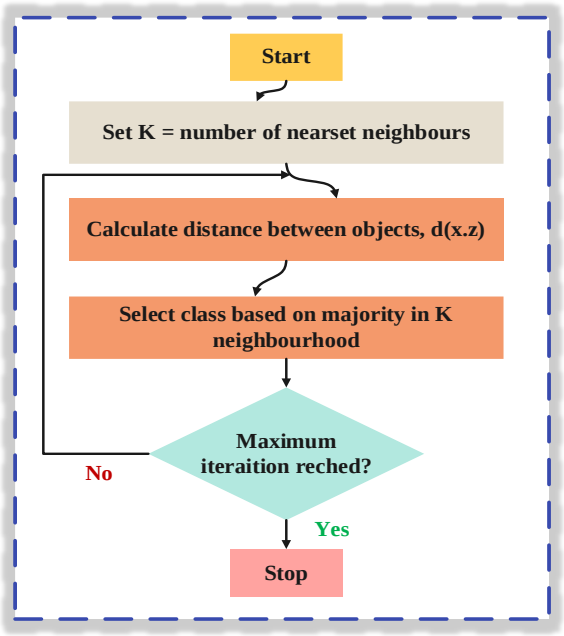


Figure 3: KNN method step-by-step flowchart.

4.3 Light gradient boosting

The LGB is one of the most efficient and fastest implementations of the gradient boosting framework, designed to run efficiently on very large-scale datasets. LGB is a boosting technique that iteratively creates an additive tree, integrating multiple weak learners to generate a strong learner. It primarily minimizes the loss function using gradient-based optimization [21].

It works efficiently for a dataset comprising n samples and m features displayed as (X_i, y_i): $i = 1 \ldots n$, where $X_i \in R^m$ and $y_i \in R$. In the architecture of the LGB model, there is an ensemble of decision trees each of which predicting the output as $\hat{y}_i$. The mechanism for updating the LGB model using boosting technique is given below by utilizing the following equations [22]:

$$\varphi(X_i)^{(t)} = \varphi(X_i)^{(t-1)} + \mu f_t(X_i) = \sum_{i=1}^{t} (L\left(y_i, \hat{y}_i^{(t-1)} + f_t(X_i)\right) + \Omega(f_t)) \quad (6)$$

Here, at each $t - th$ iteration, $\varphi(X_i)$ refers to the prediction function, that is updated using by adding a newly learned weak learner of $f_t(X_i)$. The parameter μ represents the step-size (learning rate) that controls how much each successive tree contributes to the overall model. The iterative optimization minimizes the loss function by incorporating a regularization term $\Omega(f_t)$ to penalize model complexity. This regularization term, defined in Eq. (7) [22]:

$$\Omega(f_k) = \gamma T_i + \frac{1}{2} \lambda \|w\|^2 \quad (7)$$

In addition, LGB has another key distinction: it uses a leaf-wise strategy for growing trees. In this strategy, the tree splits the leaf with the largest loss reduction, as it has the highest gradient. This results in deeper, more unbalanced trees that are often more accurate. The leaf-wise growth is also significantly faster and less memory-intensive compared to the level-wise strategy, where nodes at each tree level are expanded with equal priority. Fig. 4 displays how Light GBM works.

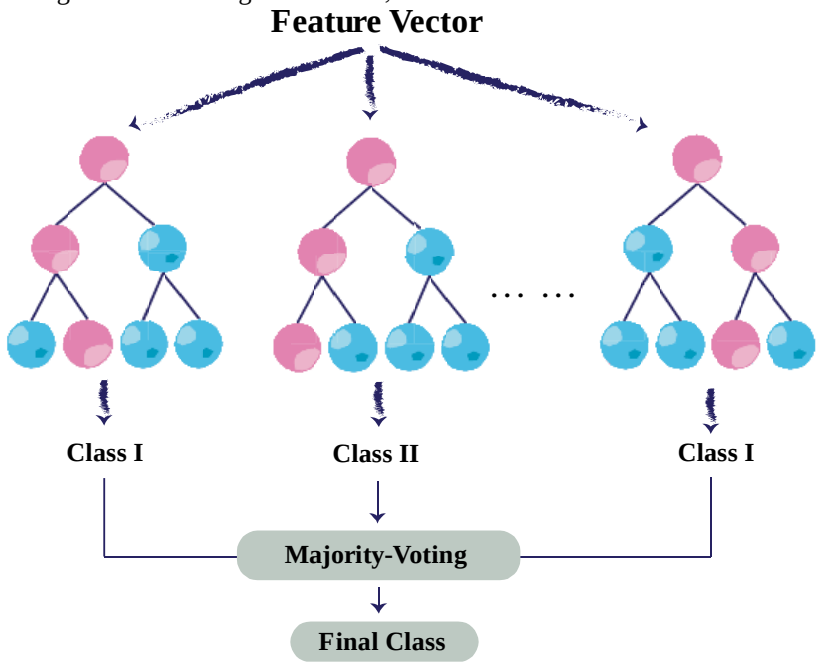


Figure 4: Procedures of the Light GBM model.

4.4 Stacking ensemble

A stacking ensemble is a sophisticated ensemble learning approach that uses a meta-model to aggregate predictions from many base models. The basic concept is to train a series of base models first and then feed the meta-model with their predictions. Stated differently, the meta-model is trained using the basic models' predictions and learns how to efficiently integrate them to enhance overall outcomes. The accumulation model process can be seen in Fig. 5.

It first initiates by training different base models on the same training input data (x), and generating the predictions set $(\hat{y}_1(x), \hat{y}_2(x), \hat{y}_3(x), \ldots \hat{y}_m(x))$ that represents the prediction of each model. Therefore, for the Stacking model on input data (x), the predictions of all base models are calculated to form the meta-feature vector of F(x). The meta-model takes this input vector in order to make the final prediction as follows [23], [24]:

$$F(x) = [\hat{y}_1(x), \hat{y}_2(x), \hat{y}_3(x), \ldots \hat{y}_m(x)] \quad (8)$$

$$\hat{y} = \hat{y}_{meta}\,(F(x)) \quad (9)$$

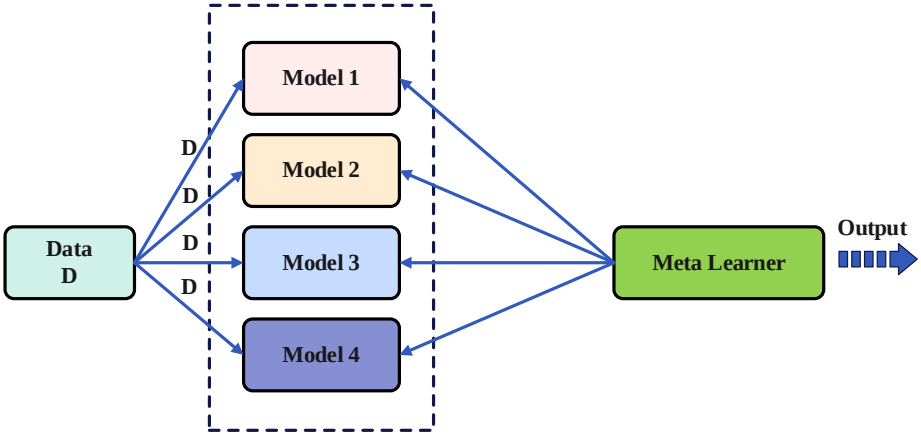


Figure 5: The process of the Stacking model.

4.5 Averaging ensemble

Averaging ensemble technique increases the overall prediction accuracy and resilience by averaging the outputs of several models. By combining the capabilities of many models, this kind of ensemble approach aims to produce forecasts that are more accurate and dependable than any one model could provide alone [25], [26]. The process of the Averaging model is presented in Fig. 6. In the case of regression, the average of the model predictions, the Averaging ensemble makes prediction as below [27]:

$$\hat{y}_{Averaging} = \sum_{i=1}^{n} \frac{\hat{y}_i}{n} \quad (10)$$

In this equation, the $\hat{y}_{Averaging}$ illustrates the output of an Averaging model, $\hat{y}_i$ indicative of the prediction of each $i-th$ model. The variable n denotes the total number of models that are included in the combination method.

Some models may vary in enhanced capabilities, so to achieve the best results, it is necessary to adjust the effect of each model in the set based on its performance. The weighted average of the ensemble prediction, or F(x), is used to do this that mathematically is represented as below [28]:

$$\hat{y}\,(x) = \sum_{i=1}^{n} \omega_i \times \hat{y}_i(x) \quad (11)$$

Here, $\hat{y}_i(x)$ denotes the prediction of the $i-th$ model within the ensemble and ω_i indicates the corresponding weights assigned to the $i-th$ model.

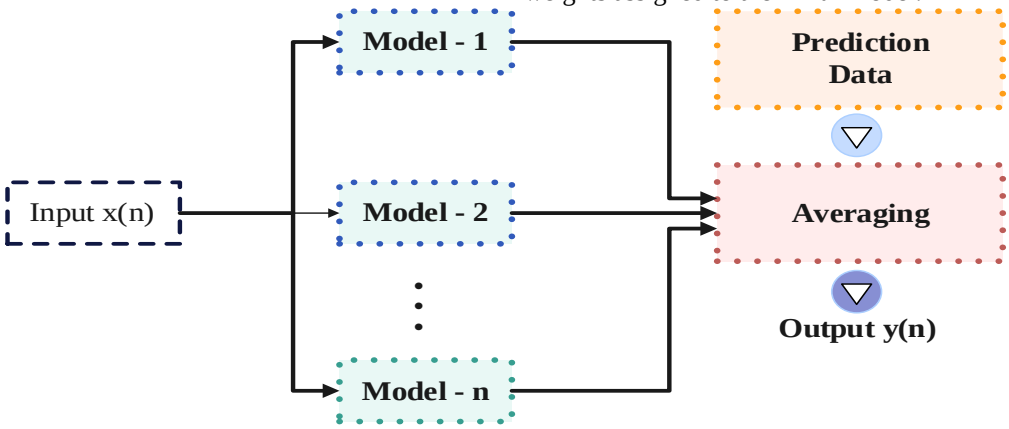


Figure 6: The process of the Averaging model.

4.6 Voting ensemble

A voting ensemble model is another sophisticated ML technique that by combining the results of several base models intends to improve predictive performance. To get more accurate and consistent results, this approach combines the predictions of multiple models, each trained using a different algorithm, to help reduce overfitting and enhance generalization on unseen data by utilizing the complementary strengths of individual models. Hard voting and soft voting are the two primary methods of voting [29], [30]. The voting model is presented in Fig. 7.

Soft voting considers the predicted probabilities of each model and averages them to produce the final prediction as below [31]:

$$\hat{y}_{soft} = \arg\max_{c \in C} \frac{1}{n} \sum_{i=1}^{n} \hat{y}_i \quad (12)$$

Here, the $\hat{y}_{soft}$ illustrates the output of a voting model, and $\hat{y}_i$ indicative of the prediction of each $i-th$ model. The variable n denotes the total number of models that are included in the combination method and C refers to the set of classes.

In hard voting, the class label that the majority of the base models predict is chosen as the final output as below [31]:

$$\hat{y}_{hard} = \arg\max_{c \in C} \sum_{i=1}^{n} (\hat{y}_i = c) \quad (13)$$

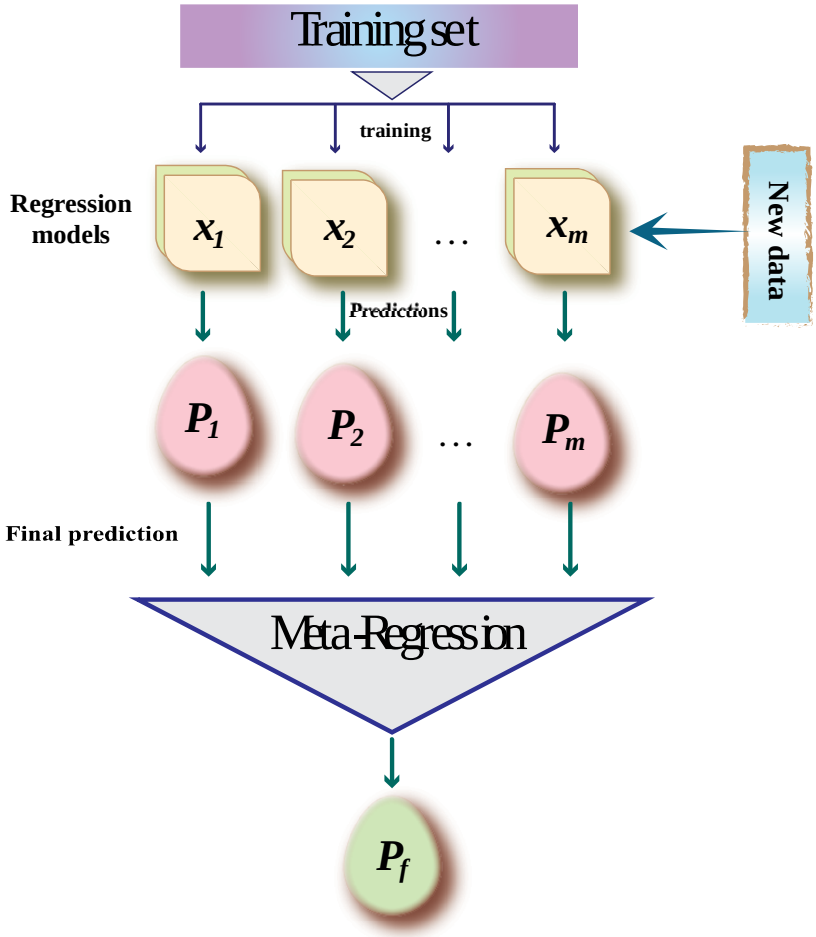


Figure 7: The flowchart of the voting model.

4.7 Evaluation metrics

In this section, the optimization algorithms are compared through statistical evaluation indices. The evaluation indices are measures widely used to assess the operation of the models designed to forecast the numerical variables. Various evaluation indices are available; in this study, the selected indices include R^2, $RMSE$, MAE, STD, VAF, FI, $MDAPE$, and $NMSE$. The equations of the preceding indices are presented in Fig. 8.

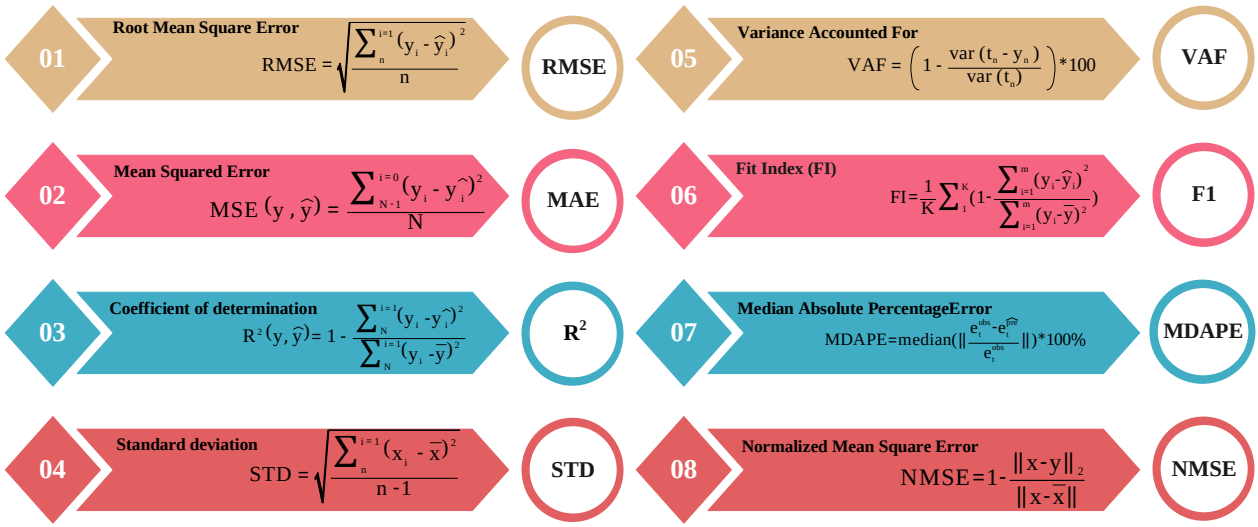


Figure 8: Evaluation equation.

5 Results and discussion

5.1 Dataset analysis

Fig. 9 is a correlation heatmap depicting Pearson correlation coefficients between different variables in a data set. The direction and intensity of linear connections between variables may be determined with the use of a heat map. This heatmap contains several variables that include p_t (%), D_x (m), D_y (m), P (×10^3kN), Avg. D_x (m) and D_y (m) have a very strong positive correlation (0.99). This implies that the displacements in the X and Y directions are approximately proportional. P_t (%) and Avg have a strong positive correlation (0.97). This shows that the reinforcement percentage is highly correlated with the average value, which can mean that the reinforcement percentage strongly affects the overall performance of the structure. P, D_x, and D_y have a moderate positive correlation (0.71 and 0.72, respectively). A higher axial load is somewhat associated with larger displacements in both the X and Y directions. P_t (%) and P have a slight positive correlation (0.2), indicating that axial load may have little effect on percent reinforcement. This correlation matrix gives insight into how various structural or input parameters relate to one another, which may also help determine the consequent analysis or modeling tasks, especially with regard to ML and structural analyses.

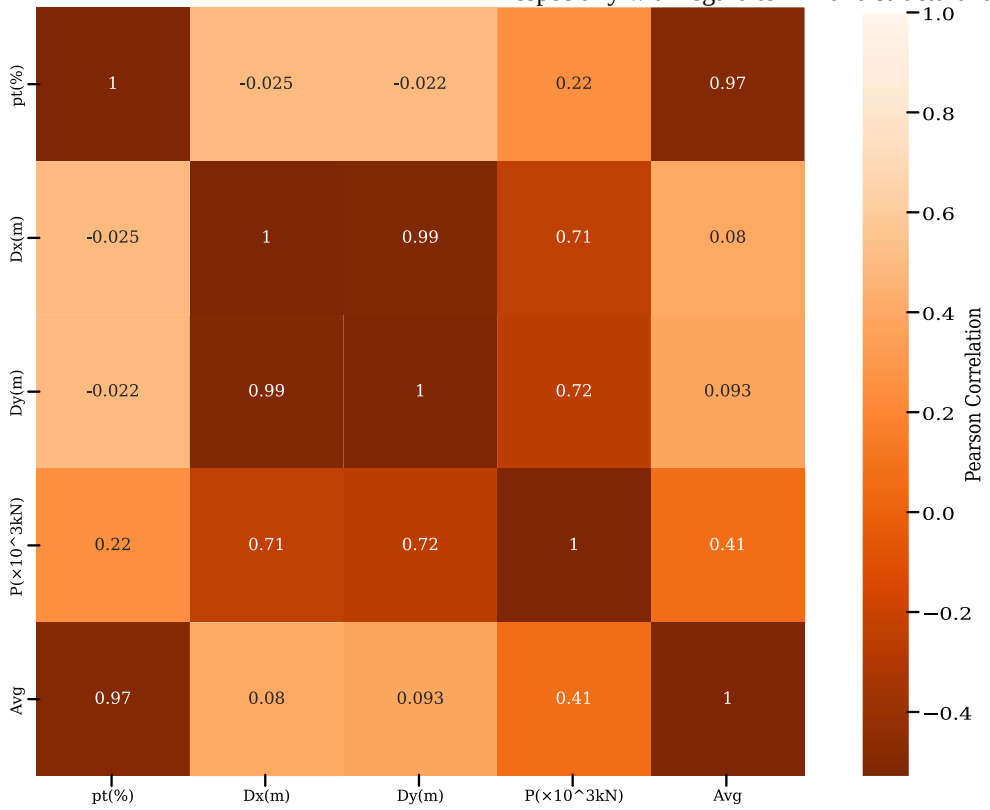


Figure 9: A heatmap of the correlation between variables in a data set.

Fig. 10 is a plot comparing pairs of different input features, with the output variable labeled "Avg" (average), represented on a color scale from light blue (0.4) to dark brown (0.8). Pair plots allow visualization of relationships between multiple variables and help observe pairwise correlations. The variables included in the design are p_t (%), D_x(m), D_y(m), and P (×10^3 kN). The diagonal plots show kernel density estimates (KDEs) or histograms for each individual variable, illustrating their distribution. Off-diagonal scatter plots display pairwise relationships between different variables. As seen in the relationship between "D_x(m)" and "p_t (%)" or between "D_y(m)" and "P (×10^3 kN)". There are some clear patterns, such as the correlation between "p_t (%)" and "D_x(m)", or between "D_y(m)" and "P (×10^3 kN)". These scatterplots show how well the variables are related, and how the color (indicating "Avg") changes according to those relationships. This pair diagram in general helps analyze the dependencies and trends between input variables and how they are related to the output variable "Avg".

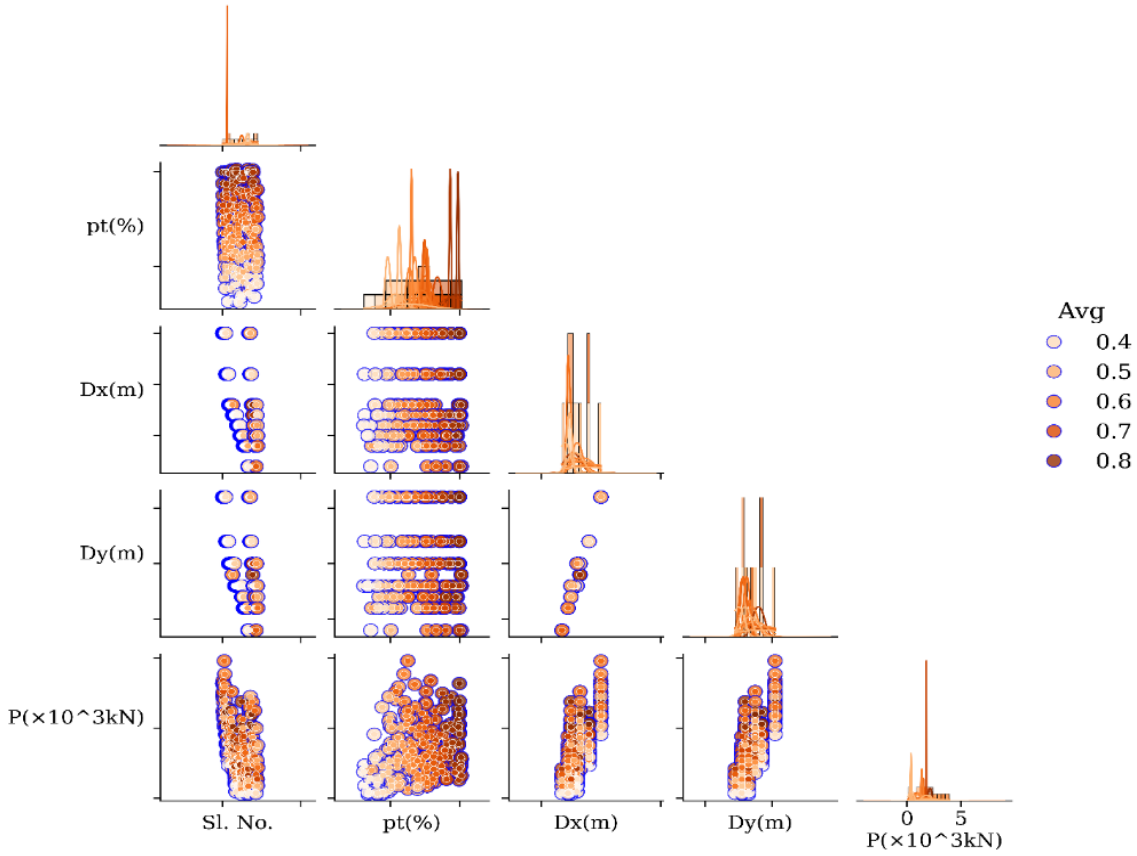


Figure 10: Pair plot with "Avg" (average) as output.

5.2 Hyperparameters tuning

Table 4 indicates the hyperparameter search ranges used to fine-tune each model. To improve prediction accuracy and avoid overfitting, parameters governing model complexity, learning rate, and regularization were methodically changed. To achieve balanced performance, robustness, and generalization across all predictive tasks, both individual models (DecisionTree, KNN and LGB) and ensemble approaches (Stacking and Voting) were optimized using specified parameter ranges.

Table 4: The hyperparameter configuration for each utilized model.

Model	Hyperparameter	Range / Values
DecisionTree	max_depth	3 – 30
	min_samples_split	2 – 20
	min_samples_leaf	1 – 10
	criterion	"squared_error", "friedman_mse", "absolute_error"
KNN	n_neighbors	3 – 15
	weights	"uniform", "distance"
	p	1 – 2
LGB	num_leaves	20 – 200
	max_depth	-1 – 20
	learning_rate	0.005 – 0.3
	n_estimators	100 – 1000

	subsample	0.6 – 1.0
	colsample_bytree	0.6 – 1.0
	reg_lambda	0 – 1
	reg_alpha	0 – 1
Stacking	cv	3 – 10
	final_estimator	Linear-Regression (), LGB ()
Voting	weights	[1, 1, 1] or tuned per model

5.3 Performance evaluation

The effectiveness of several ML models for determining the RC column's effective stiffness factors (r_x and r_y) is compiled in Tables 5 and 6. According to that, ensemble-based models, especially ensemble models show better predictive power than individual learners across both parameters. The Stacking and Averaging models obtains the highest R^2 values 0.9708 for r_x and 0.9840 for r_y on the test sets, showing low prediction error and strong generalization. The ensembles robustness in capturing nonlinear relationships among features was further confirmed by its consistently low RMSEs of 0.0182 and 0.0169 values. The Stacking and Averaging ensemble models' superior performance can be ascribed to their innate ability to incorporate complementary predictive patterns from various base learners, thereby mitigating bias and variance. In contrast the LGB algorithm outperformes traditional learners like DecisionTree and KNN in terms of accuracy, achieving R^2 values of 0.9668 and 0.9525 on the test sets for r_x and r_y. On the other hand, the DecisionTree model shows a perfect fit on the training data with train R^2 of 0.9999 for both, while in the test with R^2 0.9255 and 0.8680 for r_x and r_y shows a decline in generalization. The primary cause of this performance in the DecisionTree model is its unbridled expansion and high structural adaptability, which enable it to fully retain training data instead of picking up generalizable patterns. Near-perfect training accuracy but a noticeable drop in test performance result from the tree capturing noise and small fluctuations in the training data when it grows too deep without pruning or regularization. This suggests that poor generalization to unknown samples results from the model's complexity surpassing that of the underlying data structure. This suggests that poor generalization to unknown samples results from the model's complexity surpassing that of the underlying data structure. Moreover, The KNN model's sensitivity to data noise and local irregularities is demonstrated by its moderate accuracy but higher error dispersion (RMSEs of 0.0325 for r_x and 0.0502 for r_y).

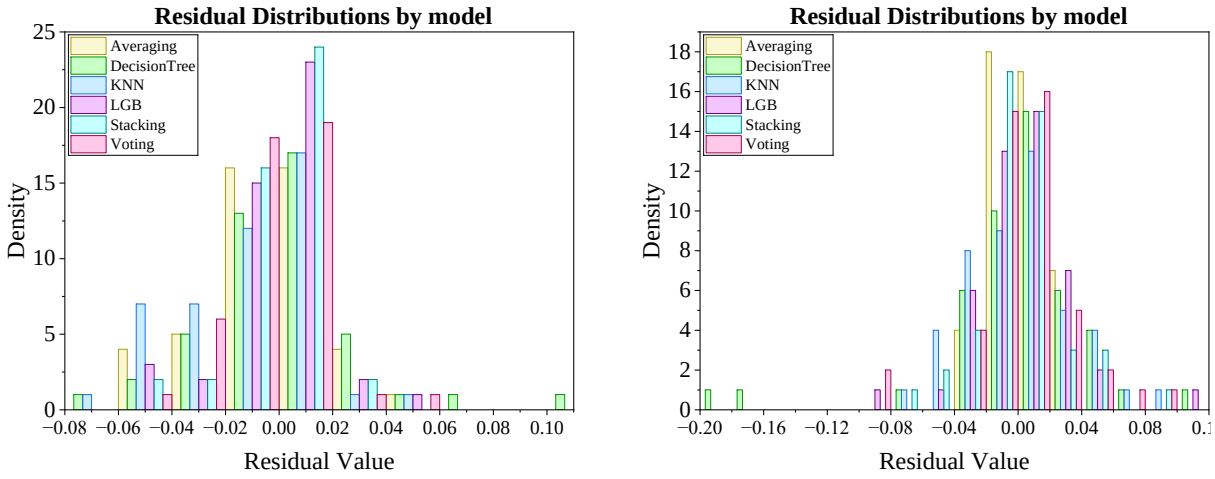
Table 5: The finding related to estimation of effective stiffness factor (r_x) of RC column by ML models.

Model	Set	RMSE	MAE	R^2	NMSE	MDAPE	STD_dev	VAF	FI
Averaging	Train	0.0222	0.0158	0.9680	0.0320	2.2987	0.0222	96.7955	0.9596
	Test	0.0222	0.0172	0.9561	0.0439	2.6409	0.0219	95.7374	0.9581
DecisionTree	Train	0.0000	0.0000	0.9999	0.0000	0.0000	0.0000	99.9999	0.9999
	Test	0.0290	0.0196	0.9255	0.0745	1.8803	0.0290	92.5558	0.9455
KNN	Train	0.0272	0.0207	0.9517	0.0483	3.0580	0.0272	95.1815	0.9504
	Test	0.0325	0.0253	0.9064	0.0936	4.4878	0.0321	90.8643	0.9389
LGB	Train	0.0166	0.0119	0.9819	0.0181	1.5797	0.0166	98.1945	0.9696
	Test	0.0193	0.0141	0.9668	0.0332	2.1360	0.0193	96.6848	0.9636
Stacking	Train	0.0152	0.0111	0.9849	0.0151	1.6039	0.0152	98.4897	0.9722
	Test	0.0182	0.0132	0.9708	0.0292	2.1026	0.0182	97.0755	0.9658
Voting	Train	0.0126	0.0096	0.9897	0.0103	1.4049	0.0126	98.9705	0.9771
	Test	0.0200	0.0147	0.9645	0.0355	2.2456	0.0200	96.4629	0.9623

Table 6: The finding related to estimation of effective stiffness factor (r_y) of RC column by ML models.

Model	Set	RMSE	MAE	R^2	NMSE	MDAPE	STD_dev	VAF	FI
Averaging	Train	0.0264	0.0170	0.9692	0.0308	1.5525	0.0264	96.9249	0.9636
	Test	0.0169	0.0141	0.9840	0.0160	2.0019	0.0167	98.4424	0.9765
DecisionTree	Train	0.0000	0.0000	0.9999	0.0000	0.0000	0.0000	99.9999	0.9999
	Test	0.0486	0.0311	0.8680	0.1320	2.3475	0.0486	86.7996	0.9324
KNN	Train	0.0391	0.0294	0.9327	0.0673	2.9637	0.0385	93.4626	0.9461
	Test	0.0502	0.0372	0.8594	0.1406	4.0270	0.0492	86.4991	0.9302
LGB	Train	0.0213	0.0151	0.9801	0.0199	1.4706	0.0213	98.0100	0.9707
	Test	0.0292	0.0200	0.9525	0.0475	2.0907	0.0290	95.3139	0.9594
Stacking	Train	0.0199	0.0140	0.9826	0.0174	1.4744	0.0198	98.2719	0.9726
	Test	0.0288	0.0198	0.9536	0.0464	1.9463	0.0287	95.3907	0.9599
Voting	Train	0.0178	0.0131	0.9861	0.0139	1.4090	0.0176	98.6300	0.9755
	Test	0.0345	0.0239	0.9334	0.0666	2.0406	0.0345	93.3625	0.9520

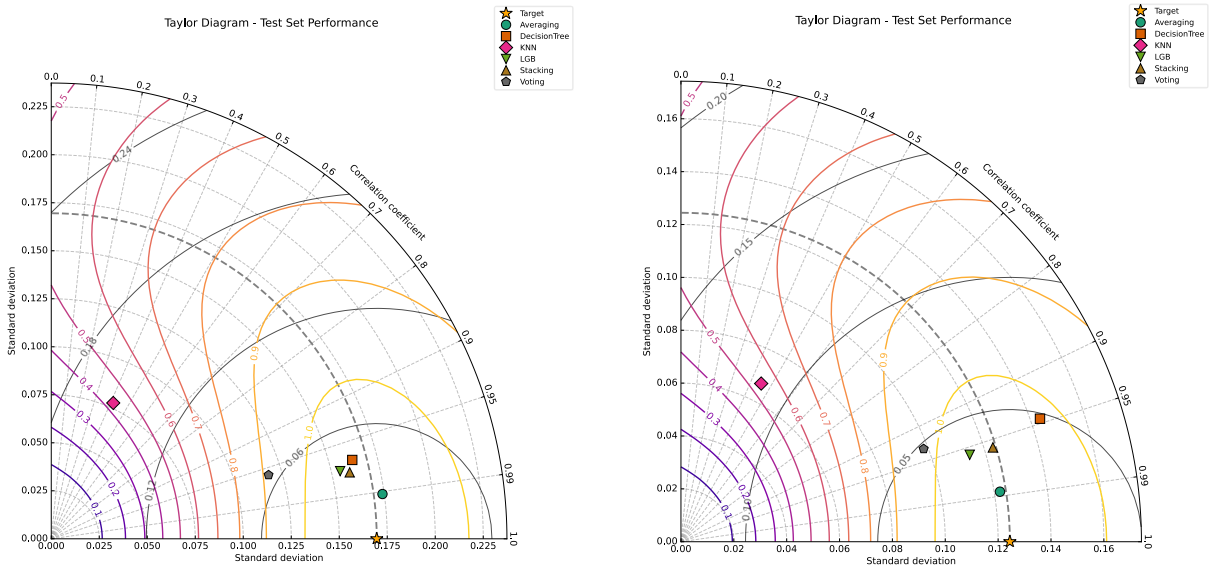
The residual distribution plots for estimating the effective stiffness factors (r_x and r_y) of RC columns using a variety of ML models are shown in Fig. 11. A properly calibrated model should ideally yield residuals that are symmetrically distributed around zero with little dispersion, suggesting that the model consistently follows the target variable. Ensemble-based techniques, such as stacking, voting, and averaging, show compact, bell-shaped residual patterns centered around zero, indicating strong predictive consistency and low bias across samples, as shown in Fig. 11 (a) for r_x and Fig. 11 (b) for r_y. This distributional behavior demonstrates how ensemble frameworks can better integrate individual learners' strengths while addressing their weaknesses, improving generalization ability and resilience to data noise. On the other hand, DecisionTree and KNN models, show more variance and sensitivity to local fluctuations in the data due to their wider and somewhat asymmetric residual distributions. Although the residual spread of the LGB model is smaller than that of these individual learners, it still shows slight departures from perfect normality, indicating limited bias in extreme situations. Out of all the models that were evaluated, Stacking and Averaging models obtained the most concentrated residual density. This suggests that they effectively capture the intricate nonlinear interactions between input features, leading to the lowest overall prediction uncertainty. Their exceptional ability to optimize the bias-variance trade-off is further demonstrated by the ensemble model residuals' consistent alignment near zero.



(a) r_x (b) r_y

Figure 11: Residual distributions by models in the estimation of effective stiffness factors of RC column.

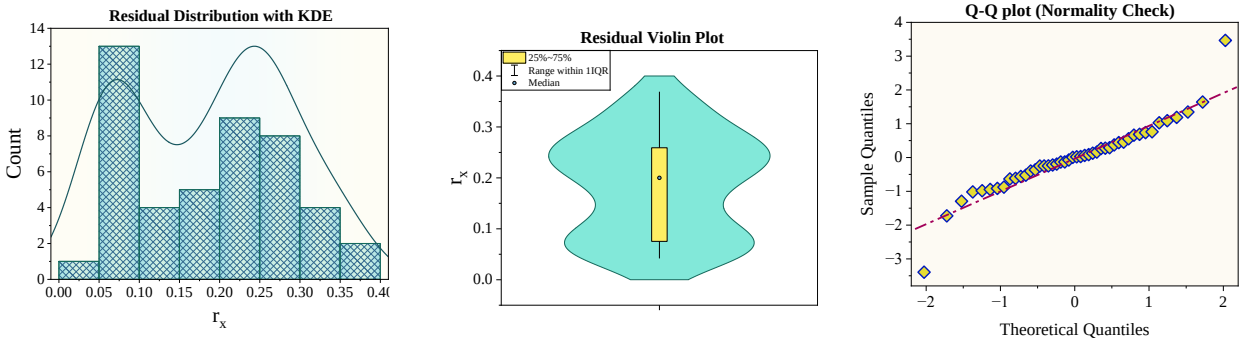
A thorough comparison of the developed ML models' predictive performance against the observed data is shown in Fig. 12 via a Taylor plot. The correlation coefficient, RMSE, and standard deviation of each model are used to illustrate its performance. Strong agreement with the measured values and high predictive reliability are indicated by models with low RMSE and high correlation whose markers are placed near the reference (observed) point. On the other hand, models that are farther away from the reference point exhibit weaker correlation and larger deviations, which indicate decreased accuracy. As shown, the Stacking and Averaging ensemble models are the ones that are closest to the reference point, demonstrating their high consistency, low prediction error, and outstanding ability to replicate the variability of the observed stiffness data.



(a) r_x (b) r_y

Figure 12: Taylor digram for developed models in the estimation of effective stiffness factors of RC column.

Moreover, to further examine the residual distribution and reliability of the ensemble models, Fig. 13 presents the residual histograms, violin plots, Kernel Density Estimation (KDE) curves, and Q–Q plots for the Stacking model in predicting r_x and the Averaging model in predicting r_y. These visual analyses provide insight into the error behavior and normality of the residuals. The residuals' distribution and conformity to the normal distribution are shown by the plots, which offer information on the bias and accuracy of the staking and averaging model. Based on KDE plot, the r_y residuals are more symmetrically centered around zero, and the residuals for r_x with slightly right-skewed distribution also exhibit concentration around zero indicating a balanced prediction behavior by staking and averaging models. The violin plots, in which the dot indicates the median, the central black line shows the range within 1.5×IQR, and the yellow box represents the interquartile range (25th–75th percentile), further illustrate the distribution and concentration of residuals. In both situations, the residuals' concentrated and narrow distribution around the median attests to the model's impartiality and resilience. Moreover, based on the Q–Q plots, the comparing sample quantiles with the theoretical quantiles indicates that the data points in both plots are nearly parallel to the diagonal reference line, suggesting that the residuals for staking and averaging have a distribution that is roughly normal. The extreme tails show only slight deviations, which are normal for empirical data and do not substantially deviate from the normalcy assumption. All things considered, the residual analysis confirms the ensemble model' generalizability and dependability for precise stiffness ratios estimation in RC columns.



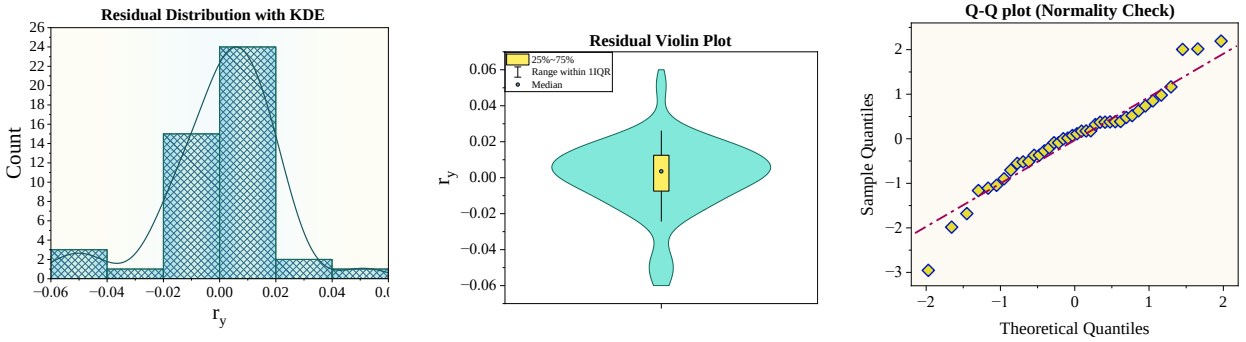
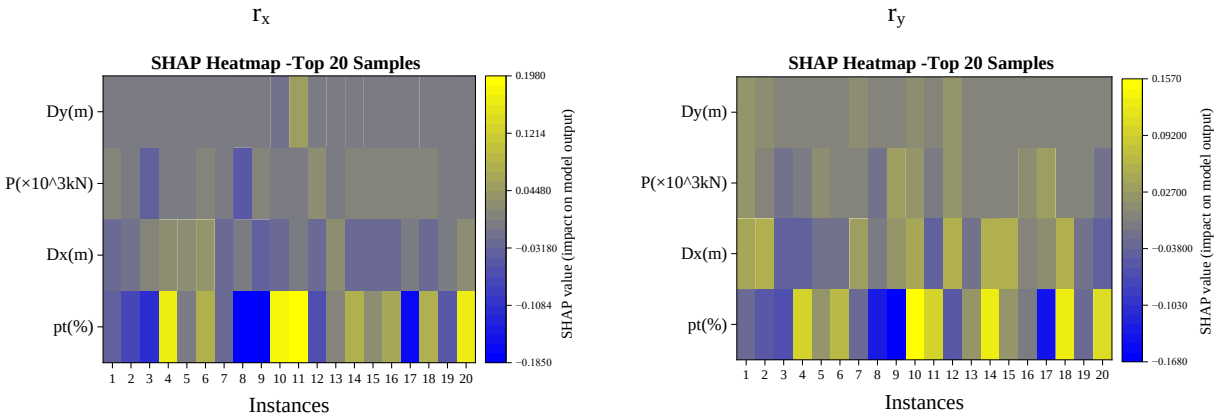


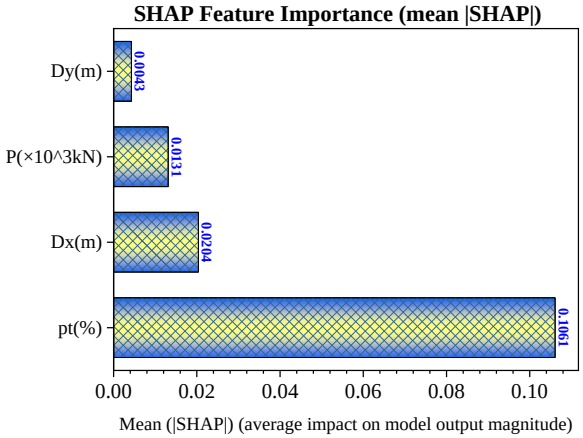
Figure 13: The residual analysis using violin, KDE and Q-Q plots for the Stacking model in predicting r_x and the Averaging model in predicting r_y.

5.4 SHAP-based interpretability analysis

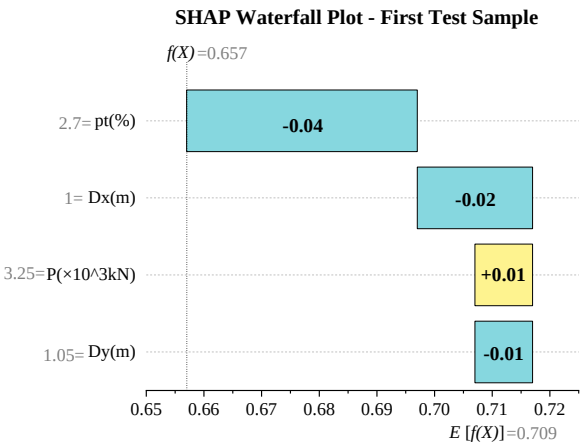
Figs 14 illustrate the SHAP-based plots for heatmap, feature importance, waterfall and values for the effective stiffness ratios r_x and r_y. These plots provide a clear interpretability framework for understanding how individual features influence the model's output predictions. The SHAP heatmaps in Fig. 14 (a) for the 20 samples, indicates the direction and strength of each feature's influence on the model output. In this plot the negative values (blue regions) suggest a decreasing effect, positive SHAP values (yellow regions) indicate that the feature increases the predicted stiffness ratio. The heatmaps make it clear that p_t is the most influential parameter, indicating that the percentage of steel reinforcement have a significant and steady impact on both r_x and r_y. The SHAP feature importance plot in Fig. 14 (b), which quantify the importances score of features across all samples. As depicted in this plot, for both r_x and r_y, the p_t parameter that indicates the reinforcement ratio with the highest importance score of 0.11 and 0.8 emerges as the most influential parameter. Moreover, the SHAP-based waterfall and values in Fig. 14 (c and d) confirm these findings and indicates that the reinforcement ratio is the most influential parameter in determining the effective stiffness. This dominance implies that changes in the percentage of reinforcement have a significant impact on the stiffness of RC columns. While the column width has a comparatively smaller impact, and the column depth and axial load exhibit moderate contributions. This ranking is constant in both directions, indicating that geometry and reinforcement have a greater influence on stiffness behavior than loading circumstances.



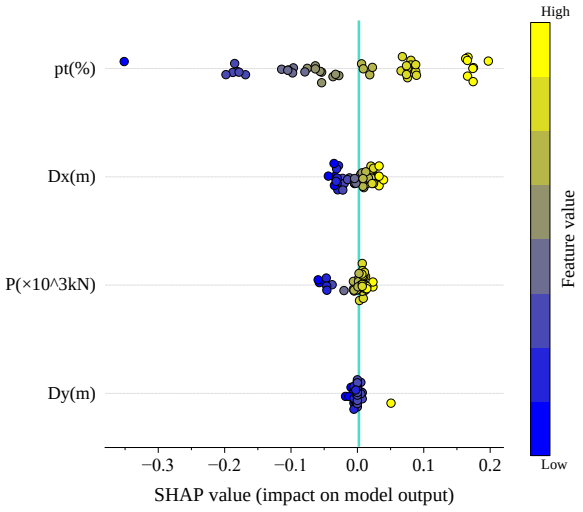
(a) SHAP Heatmap



(b) SHAP Feature Importance



(c) SHAP waterfall plots



(d) SHAP value

Figure 14: The interpretability analysis of developed Staking in predicting r_x and Averaging models in predicting r_y using SHAP-based plots.

5.5 Model's performance discussion

Our analysis showed that the Stacking and Averaging ensemble models had the highest predictive accuracy with test-set R^2 values of 0.9708 for r_x and 0.9840 for r_y, as well as correspondingly low RMSE values. This suggests that there are little generalization gap and strong predictive consistency between the training and test datasets. Indeed, the use of GS-based optimization with 5-fold cross-validation result high generalization ability for these models, as evidenced by the small (<2%) difference between training and testing results. When compared with related studies in Table 1, the obtained results demonstrate

superior or at least comparable accuracy. For instance, previous works reported R^2values of 0.868 for ANN [11], 0.931 for M5-Tree [12], 0.977-0.923 for CatBoost [13], 0.90–0.97 for GPR [14], 0.906 for ANN [15] and up to 0.960 for DecisionTree [16] in estimation of various parameters of RC columns. Thus, the Stacking and Averaging ensembles outperformed most existing ML-based frameworks in predicting RC structural parameters. The Stacking and Averaging ensemble models' superior performance can be ascribed to their innate ability to incorporate complementary predictive patterns from various base learners, thereby mitigating bias and variance. By calculating the mean of the predictions made by several separate learners the Averaging ensemble model aggregates their outputs which evens out individual model fluctuations and present a more stable and universal response results from this aggregation. The averaging process improves robustness against data noise and outliers and reduces the risk of overfitting associated with any one model because each learner captures distinct aspects of the input–output relationship. The Stacking ensemble, on the other hand, uses a more advanced meta-learning framework, where a higher-level meta-learner uses the predictions from multiple base models as new input features. Based on their residual errors, this meta-learner, typically a regression or boosting algorithm, learns how to combine and weight each base learner's strengths in the best possible way. Consequently, stacking is able to model highly nonlinear relationships that individual learners might overlook by learning inter-model dependencies and error correction patterns in addition to aggregating predictions.

Despite these promising outcomes, several limitations are acknowledged. The developed Staking and Averaging models were trained on a static dataset and have not been implemented or evaluated in real-time structural monitoring environments and were trained on a static dataset. Furthermore, even though feature importance analyses and SHAP offer some interpretability, more effort is required to improve explainability using sophisticated model transparency frameworks. To increase deployment viability and interpretive reliability, future studies might concentrate on combining explainable AI modules, real-time data streams, and hybrid physics–ML models.

6 Conclusions

This work presented a flexible and high-performance ensemble framework based on the Stacking and Averaging techniques to improve the prediction of two crucial structural parameters that are directly related to the effective stiffness factors (r_x and r_y) of reinforced concrete structures. The suggested method combines several base learners, utilizing their complementary advantages to better capture intricate nonlinear dependencies and multi-factor interactions, lessen sensitivity to data noise, and increase predictive accuracy. These ensemble models were methodically contrasted with other algorithms, such as Voting, KNN, LGB, and Decision-Tree for thorough validation and benchmarking. Furthermore, the GS-based optimization used in the ML processes enabled further improvements through hyperparameter tuning and leading to more accurate predictions and greater efficiency. The performance evaluation showed that the Stacking and Averaging ensemble models had the highest predictive accuracy with test-set R^2 values of 0.9708 for r_x and 0.9840 for r_y, as well as correspondingly low RMSE values. These optimized ensemble models could help engineers to design safer, more cost-effective, and environmentally friendly structures for RC structure. They These models significantly reduce the time required for structural design and analysis by automating the stiffness prediction process. Engineers can rapidly evaluate multiple design configurations without the need for time-consuming iterative calculations, thus simplifying the RC column design process.

Moreover, using SHAP-based analysis, the models can handle a wide range of input variables and provide insight into which parameters (e.g., reinforcement ratio, concrete dimension, axial load) most affect the stiffness of RC columns. The outcomes make it clear that the percentage of steel reinforcement is the most influential parameter and have a significant and steady contribution on both r_x and r_y values. This understanding supports better decision-making during the design phase. This allows the development of lightweight, cost-effective, and structurally sound designs.

Abbreviations

RC	Reinforced concrete	LGB	Light gradient boosting
KNN	K-Nearest Neighbors	$\hat{y}_m(x)$	The prediction of each $m-th$
GS	Grid Search	$\hat{y}_i$	Variable of the meta-model
SHAP	Shapley Additive Explanations	$\hat{y}_{Averaging}$	The output of an Averaging model
ANN	Artificial Neural Network	$\hat{y}$	Final prediction of model
GPR	Gaussian Process Regression	$f_i(x)$	Prediction of the $i-th$ model
ML	Machine Learning	ω_i	Corresponding weights assigned to the $i-th$ model.
$\Omega(f_t)$	Complexity		

Acknowledgements

We would like to take this opportunity to acknowledge that there are no individuals or organizations that require acknowledgment for their contributions to this work.

Authors' contributions

Xine Yan conducted analysis

Juanjuan Wang performed Data collection also

Yetao Cong carried out simulation

Zhanhao Zhang evaluate the first draft of the manuscript, editing and writing.

Funding

This work was supported by 2023 Youth Innovation Team Scientific Research project of Shaanxi Provincial Education Department. (Project No: 23JP086)

Xi'an Key Laboratory of Track Subgrade Bed Disease Monitoring and Prevention.

The Youth Innovation Team of Shaanxi Universities.

Ethical Approval

The research paper has received ethical approval from the institutional review board, ensuring the protection of participants' rights and compliance with the relevant ethical guidelines.

References

[1] S. Das, I. Mansouri, S. Choudhury, A. H. Gandomi, and J. W. Hu, "A prediction model for the calculation of effective stiffness ratios of reinforced concrete columns," *Materials*, vol. 14, no. 7, p. 1792, 2021. https://doi.org/10.3390/ma14071792

[2] B. Zhao, F. Taucer, and T. Rossetto, "Field investigation on the performance of building structures during the 12 May 2008 Wenchuan earthquake in China," *Eng Struct*, vol. 31, no. 8, pp. 1707–1723, 2009. https://doi.org/10.1016/j.engstruct.2009.02.039

[3] M. Vafaei, M. Baniahmadi, and S. C. Alih, "The relative importance of strong column-weak beam design concept in the single-story RC frames," *Eng Struct*, vol. 185, pp. 159–170, 2019. https://doi.org/10.1016/j.engstruct.2019.01.126

[4] C. Shao, W. Wei, G. Xu, Q. Qi, and C. Wang, "Assessment on effective stiffness of RC hollow columns based upon semi-static experiment," in *Structures*, Elsevier, 2023, pp. 882–894. https://doi.org/10.1016/j.istruc.2023.04.071

[5] C. G. Nechevska and A. Roshi, "Rehabilitation of RC buildings in seismically active regions using traditional and innovative materials," *Građevinski materijali i konstrukcije*, vol. 62, no. 3, pp. 19–30, 2019.

[6] J. Bell, "What is machine learning?," *Machine learning and the city: applications in architecture and urban design*, pp. 207–216, 2022.

[7] D.-C. Feng, W.-J. Wang, S. Mangalathu, G. Hu, and T. Wu, "Implementing ensemble learning methods to predict the shear strength of RC deep beams with/without web reinforcements," *Eng Struct*, vol. 235, p. 111979, 2021. https://doi.org/10.1016/j.engstruct.2021.111979

[8] Y. Li, X. Lu, H. Guan, and L. Ye, "An improved tie force method for progressive collapse resistance design of reinforced concrete frame structures," *Eng Struct*, vol. 33, no. 10, pp. 2931–2942, 2011. https://doi.org/10.1016/j.engstruct.2011.06.017

[9] X. Lu, L. Xie, H. Guan, Y. Huang, and X. Lu, "A shear wall element for nonlinear seismic analysis of super-tall buildings using OpenSees," *Finite Elements in Analysis and Design*, vol. 98, pp. 14–25, 2015. https://doi.org/10.1016/j.finel.2015.01.006

[10] W. Fan *et al.*, "Machine learning applied to the design and inspection of reinforced concrete bridges: Resilient methods and emerging applications," in *Structures*, Elsevier, 2021, pp. 3954–3963. https://doi.org/10.1016/j.istruc.2021.06.110

[11] F. Kazemi, N. Asgarkhani, and R. Jankowski, "Machine learning-based seismic response and performance assessment of reinforced concrete buildings," *Archives of Civil and Mechanical Engineering*, vol. 23, no. 2, p. 94, 2023. https://doi.org/10.1007/s43452-023-00631-9

[12] Z. M. Yaseen, "Machine learning models development for shear strength prediction of reinforced concrete beam: a comparative study," *Sci Rep*, vol. 13, no. 1, p. 1723, 2023. https://doi.org/10.1038/s41598-023-27613-4

[13] I. Alhalil and M. F. Gullu, "Predicting Main Characteristics of Reinforced Concrete Buildings Using Machine Learning," *Buildings*, vol. 14, no. 9, p. 2967, 2024. https://doi.org/10.3390/buildings14092967

[14] Z. Tuna Deger and G. Taskin Kaya, "A Novel GPR-Based Prediction Model for Cylic Backbone Curves of Reinforced Concrete Shear Walls," *arXiv e-prints*, p. arXiv-2111, 2021. https://doi.org/10.1016/j.engstruct.2022.113874

[15] M. Mohammadagha, M. Najafi, V. Kaushal, and A. M. A. Jibreen, "Machine learning models for reinforced concrete pipes condition prediction: The state-of-the-art using artificial neural networks and multiple linear regression in a Wisconsin case study," *arXiv preprint arXiv:2502.00363*, 2025. https://doi.org/10.48550/arXiv.2502.00363

[16] H. Naderpour, M. Mirrashid, and P. Parsa, "Failure mode prediction of reinforced concrete columns using machine learning methods," *Eng Struct*, vol. 248, p. 113263, 2021. https://doi.org/10.1016/j.engstruct.2021.113263

[17] S. Das and S. Choudhury, "Influence of effective stiffness on the performance of RC frame buildings designed using displacement-based method and evaluation of column effective stiffness using ANN," *Eng Struct*, vol. 197, p. 109354, 2019. https://doi.org/10.1016/j.engstruct.2021.113263

[18] P. Gulati, A. Sharma, and M. Gupta, "Theoretical study of decision tree algorithms to identify pivotal factors for performance improvement: A review," *Int. J. Comput. Appl*, vol. 141, no. 14, pp. 19–25, 2016.

[19] A. B. Hassanat, M. A. Abbadi, G. A. Altarawneh, and A. A. Alhasanat, "Solving the problem of the K parameter in the KNN classifier using an ensemble learning approach," *arXiv preprint arXiv:1409.0919*, 2014. https://doi.org/10.48550/arXiv.1409.0919

[20] V. Prasath, H. A. A. Alfeilat, O. Lasassmeh, A. Hassanat, and A. S. Tarawneh, "Distance and similarity measures effect on the performance of K-nearest neighbor classifier—A review. arXiv 2017," *arXiv preprint arXiv:1708.04321*.

[21] R. P. Sheridan, A. Liaw, and M. Tudor, "Light gradient boosting machine as a regression method for quantitative structure-activity relationships," *arXiv preprint arXiv:2105.08626*, 2021. https://doi.org/10.48550/arXiv.2105.08626

[22] G. Ke *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree," *Adv Neural Inf Process Syst*, vol. 30, 2017.

[23] D. H. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992. https://doi.org/10.1016/S0893-6080(05)80023-1

[24] B. Pavlyshenko, "Using stacking approaches for machine learning models," in *2018 IEEE second international conference on data stream mining & processing (DSMP)*, IEEE, 2018, pp. 255–258. https://doi.org/10.1109/DSMP.2018.8478522

[25] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*. CRC press, 2012.

[26] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, vol. 2. Springer, 2009.

[27] D. Arpit, H. Wang, Y. Zhou, and C. Xiong, "Ensemble of averages: Improving model selection and boosting performance in domain generalization," *Adv Neural Inf Process Syst*, vol. 35, pp. 8265–8277, 2022.

[28] M. A. I. Neloy, N. Nahar, M. S. Hossain, and K. Andersson, "A weighted average ensemble technique to predict heart disease," in *Proceedings of the Third International Conference on Trends in Computational and Cognitive Engineering: TCCE 2021*, Springer, 2022, pp. 17–29. https://doi.org/10.1007/978-981-16-7597-3_2

[29] A. Dogan and D. Birant, "A weighted majority voting ensemble approach for classification," in *2019 4th international conference on computer science and engineering (UBMK)*, IEEE, 2019, pp. 1–6. https://doi.org/10.1109/UBMK.2019.8907028

[30] F. Leon, S.-A. Floria, and C. Bădică, "Evaluating the effect of voting methods on ensemble-based classification," in *2017 IEEE international conference on INnovations in intelligent Systems and applications (INISTA)*, IEEE, 2017, pp. 1–6. https://doi.org/10.1109/INISTA.2017.8001122

[31] L. Rokach, "Ensemble-based classifiers," *Artif Intell Rev*, vol. 33, no. 1, pp. 1–39, 2010. https://doi.org/10.1007/s10462-009-9124-7

DEEPMIND: A Deep Learning Model with RNN and Attention for Personalized Mental Health Education Effectiveness Assessment and Content Optimization

Li Lan[1], Jiayu Wang[2], Lili Song[3,*]
[1]Department of Youth Core Competency Development and Mental Health, Jiaxing Nanyang Polytechnic Institute, Jiaxing, Zhejiang, 314031, China
[2]College of Marxism, Jiaxing University, Jiaxing, Zhejiang, 314001, China
[3]Henan Institute of Economics and Trade,Zhengzhou 450000,Henan,China
E-mail: lili_song7188@126.com
[*]Corresponding author

Keywords: mental health education, deep learning, content optimization, effectiveness assessment, RNN, NLP

Received: July 2, 2025

Mental health education plays a critical role in improving psychological well-being and awareness, particularly in educational and organizational settings. Evaluating its effectiveness and optimizing the content delivery can significantly enhance engagement and learning outcomes. Traditional assessment methods often rely on manual surveys and static analytics, which are limited in personalization and fail to capture complex behavioral patterns. These approaches lack adaptability and struggle to provide real-time feedback for content improvement. To address these limitations, this paper proposes a novel deep learning-based framework called DEEPMIND (Deep Engagement and Effectiveness Prediction Model Integrating Neural Decisioning). DEEPMIND utilizes recurrent neural networks (RNNs) and attention mechanisms to analyze learner interaction data and sentiment patterns for assessing educational effectiveness. The proposed method dynamically evaluates learner responses, behavioral signals, and content features to recommend personalized content optimizations. By integrating natural language processing (NLP) and deep feature extraction, DEEPMIND continuously improves content relevance and user engagement. Experimental results demonstrate that DEEPMIND outperforms baseline models in accuracy, adaptability, and content enhancement, leading to a 25% increase in learner retention and a 30% improvement in mental health knowledge acquisition. The model has been trained and tested using the Sentiment Analysis for Mental Health dataset, which comprises 10,000 labeled samples. It separated the training and testing sets into five groups and utilized an 80:20 ratio. DEEPMIND has performed significantly better than baseline algorithms in predicting engagement (91.2%), detecting sentiment (89.4%), and adapting (87.6%).

Povzetek: Raziskava predstavlja globoko-učeči se pristop za vrednotenje in izboljševanje izobraževanja o duševnem zdravju, ki z analizo vedenja in odzivov udeležencev omogoča bolj prilagojeno, učinkovito in vključujoče učenje.

1 Introduction

Including mental health education in modern training courses and curriculum aims to promote psychological resilience, emotional intelligence, and self-awareness [1]. Early intervention and long-term well-being depend on the practical implementation and evaluation of mental health education, which is becoming increasingly significant as the frequency of mental health issues rises worldwide [2]. Though much attention has been paid, most evaluation techniques still rely on static assessments or traditional surveying, neither of which are effective in capturing people's reactions and involvement in the here and now [3]. Less than optimal personalization and effectiveness follow from a lack of understanding of how people process and absorb mental health information in these approaches [4]. Furthermore, current content distribution systems are not adaptable enough to meet the needs of various types of learners, which is especially problematic in online or hybrid environments. Retention and involvement suffer when teachers fail to react quickly enough to student input, adjusting courses to meet particular needs [5].

With the help of data-driven analysis and real-time feedback systems, these problems can now be better tackled to recent developments in deep learning [6]. An innovative strategy for mental health education is presented by deep learning algorithms, which can predict intricate patterns in student behavior, emotion, and participation [7]. Through the analysis of interaction data and dynamic content optimization, this research presents a new framework called DEEPMIND that aims to improve mental

health education [8]. DEEPMIND offers a personalized and adaptive learning experience by leveraging attention layers, NLP, and RNNs. DEEPMIND uses continuous feedback loops to identify what works, for whom, and under what circumstances where teachers can enhance their materials and delivery strategies [9]. In the long term, this approach holds significant promise to raise psychological well-being, student satisfaction, and educational efficacy [10].

Digital technology and machine intelligence have advanced significantly, making mental health therapy more personalized, accessible, and effective. Increasingly, computer-based treatment systems are utilizing conversational bots to assist individuals in modifying their behavior and identifying potential mental health concerns [31-32]. These clever cognitive systems assist individuals in coping with stress, anxiety, and depression by getting them to talk about their problems and change their behavior. Smartphone-based intervention and assessment methods also enable continuous monitoring and support for individuals, allowing for scalable interventions [33]. Modern smart assistants not only help address the issue of unequal access to mental health care, but they also utilize persuasive technologies to encourage people to change their behavior. These changes underscore the importance of digital tools in developing mental health support systems that operate in real-time and prioritize the user's needs. More individuals desire to participate in mental health treatments and educational models when these concepts are incorporated. This makes it possible for modern therapy's scalable, data-driven strategies to work [34-35].

1.1 Problem statement

Their reliance on fixed assessments, human feedback, and broad material delivery results in a fairly limited ability to personalize and react in real-time. These limitations induce low involvement, poor retention of knowledge, and ineffective behavior modification. The suggested model desperately need a sophisticated, strong system that can track user learning and instantly change material in response to their emotions and behavior.

1.2 Motivation

Given the rising prevalence of mental health issues, early intervention and awareness-raising education, in particular, is absolutely vital. Sadly, generic, one-size-fits-all solutions usually fail to interest different types of students. Deep learning enables access to customized, adaptable learning approaches. A dynamic framework that incorporates real-time user data for evaluation and content development helps improve mental health learning outcomes and long-term behavioral impact.

1.3 Contributions

The major contributions are;

- Our novel model, DEEPMIND, evaluates the effectiveness of mental health education by using deep learning by means of user interactions and replies.
- DEEPMIND constantly changes and improves the learning materials based on every user's behavior and emotions to maximize their effectiveness.
- DEEPMIND enhances learning, memorization, and involvement, compared to other strategies, as demonstrated in our experiments.

The remaining section of this paper is organized as follows: Section 2 reviews past studies on the evolution of mental health education. Section 3 describes the proposed DEEPMIND process. Section 4 examines and compares our proposed approach with other conventional approaches. Section 5 concludes with a discussion of potential future studies.

2 Related works

The current advancements in deep learning and machine learning have tremendously helped mental health research. Studies on techniques like natural language processing, adaptive systems, therapy prediction, and multimodal data processing have examined the three main objectives of these systems: user interaction, intervention customization, and improvement of mental disease identification. This part emphasizes significant research in the field to demonstrate how artificial intelligence is progressively becoming relevant in mental health therapies.

This work offers a multimodal deep learning architecture (MmDL) using voice, text, and facial expressions to detect mental illnesses. The model runs recurrent and convolutional neural networks to improve classification accuracy [11]. Experimental results on benchmark datasets expose a winner when compared to single-modality models. This approach guides the path toward automated mental health diagnostics, able to consider a range of inputs and yield real-time results.

This study uses machine learning to examine how individuals apply online mental health interventions; based on this, one can find patterns in using behavior connected with therapeutic results [12]. The model divides users into profiles based on their degree of participation in an effort to identify which ones would benefit most. The findings reveal that customizing intervention delivery using adaptive mechanisms can help online mental health programs to be more successful.

This research looks at mobile apps for mental health using thematic analysis of user reviews (MH-TA-UR) and machine learning. App store evaluations

separate user emotion, usability issues, and general pleasure with the features [13]. The paper claims that many applications lack clinical validation and customization. The approach offers a scalable means to assess app quality, such that it enables developers to create better mental health apps.

Machine learning approaches used in adaptive and personalized health systems are discussed in this paper with a focus on mental wellness applications. Researches address intervention customizing via reinforcement learning, supervised, unsupervised, and unlearning [14]. The good impacts of customizing on adherence and outcomes are the main topics of this review. Exploration of issues like data privacy and bias can direct future studies in health-aware intelligent systems.

Within the machine learning techniques discussed in this taxonomy of mental health prediction applications (MHPA) are classification, clustering, and natural language processing. Researchers discuss case studies and techniques for identifying anxiety, depression, and PTSD, as well as strategies for addressing interpretability, data imbalance, and ethics, among other issues [15]. Scholars working on mental health therapies utilizing ML will find a great resource in this work.

Cognitive treatments for mental health combining augmented and virtual reality with machine learning: an all-encompassing review. It focuses especially on how ML improves diagnosis, tracks development, and customizes treatment settings to fit every patient [16]. The paper stresses the advantages, which include immersive therapy and real-time feedback. Still, it admits the challenges related to ethics, clinical validation, and technology. The study emphasizes how much artificial intelligence is becoming more and more relevant in VR mental health treatments.

This thorough review investigates the psychological uses of NLP and machine learning. It clarifies how NLP could find in-patient texts of suicidal, depressed, and worried ideas. The study emphasizes sources of data, approaches, and performance findings of relevance [17]. While interpretability and data quality remain major challenges, findings show that language-based models perform well for early detection.

Using textual data, including medical records and social media posts, the researchers propose a deep learning algorithm that could detect cases of depression. LSTMs and CNNs help the model understand emotional tone and semantic context. Results show a great degree of accuracy in the diagnosis of depression [18]. This study indicates that early mental health diagnosis and language-based passive monitoring can benefit from deep learning.

Reviewed here for the aim of identifying depression are some of the machine learning techniques, including decision trees, support vector machines, neural networks, and ensemble approaches [19]. Authors highlight types of datasets, performance criteria, model limitations, and so on. Furthermore, underlined as possible areas for more research in the field are subjects like interpretability and multimodal data integration. The assessment confirms the exciting prospects of machine learning for the field of mental health diagnosis.

Seeking to customize therapy, the paper explores how machine learning may be used to forecast results from mental treatments. It tests models using demographic, clinical, and behavioral data to identify the optimal therapies for individuals [20]. Results show that ML could significantly raise the therapy success rates. The research emphasizes future challenges, including data integration and ethics, which underlie the need to include ML in clinical decision-making for tailored mental treatment.

Results of the examined literature show that artificial intelligence can enhance the personalizing of treatments, mental health diagnosis, and optimization of interventions. Among the themes that really jump out are adaptive learning, multimodal frameworks, user engagement analysis, and immersive therapies. Despite the hope, challenges still exist, including the development of effective data analysis techniques and the assurance of clinical validation. Taken together, these items show the opportunities and paths forward for designing intelligent, customized mental health diagnosis and treatment plans. The advantages and limitations of the relevant research are shown in Table 1.

A significant number of the solutions now available have names that sound satisfactory. Still, they are constrained by designs that do not undergo any changes or inputs that only consider a single type of data. Although CNN-LSTM models are effective at determining how individuals feel about text, it is essential to note that they do not always respond immediately. It is also important to note that reinforcement learning strategies do not always include customization or emotional context. By integrating adaptive material based on real-time feedback, a recurring design that enhances attention, and natural language processing (NLP) for emotional signals, DEEPMIND can overcome these challenges. It offers a framework that is both more comprehensive and more flexible than the conventional methods typically used. As a result, it can accurately comprehend sentiment (with an accuracy of 89.4%) and effectively forecast engagement (with a 91.2% accuracy).

Table 1: Related work summary

S.No	Methods	Advantages	Limitations	Qualitative Metrics
1	Multimodal Deep Learning (CNN + RNN)	Combines text, audio, and facial data for accurate disorder recognition	Requires large, diverse data; high computational cost	Response Time: around 210 milliseconds, Accuracy: 86.5%, F1-Score: 84.7%
2	Machine Learning Clustering on User Engagement Data	Identifies engagement patterns to predict intervention success	May miss nuanced emotional or contextual user behavior	The cluster accuracy is 80%, the retention improvement is around 12%, and the cluster precision is 78.9%.
3	ML + Thematic Analysis of User Reviews	Scalable app evaluation: captures user sentiment and feature quality	Lacks clinical validation; subjective user feedback	Time to respond: 198 ms; 84.2% accuracy in finding feelings
4	Supervised, Unsupervised, and Reinforcement Learning for Adaptation	Enables personalization and improves user adherence	Data privacy and algorithmic bias issues	Return on Investment (RL-based): 82.3%; Average Retention Gain: 15–20%
5	ML Taxonomy (SVM, NLP, Clustering, etc.)	Comprehensive mental health prediction toolset	Ethical concerns and data imbalance challenges	The recall rate for finding depression is around 77%, the SVM accuracy rate is 80–85%, and the NLP F1 rate is 83%.
6	ML in AR/VR-Based Cognitive Therapy	Immersive treatment; personalized feedback in real-time	Expensive hardware; needs clinical testing	Completion of sessions: 91%, accuracy of real-time feedback: 87%, and user satisfaction: 88%.
7	NLP + ML for Text Analysis (Depression, Anxiety, Suicidal Ideation)	High early-detection accuracy using language patterns	Data quality and interpretability remain concerns	Sentiment F1: 85.6%, Detection Accuracy: 88.4%, and False Positive Rate: 6.1%
8	CNN + LSTM on Textual Data	Effective semantic understanding for depression detection	Relies heavily on text availability and quality	Recognizing Depression: 88.0% precision, 89.2% accuracy, and 0.92 AUC
9	Review of ML Algorithms (SVM, RF, NN, etc.)	Highlights algorithm performance and future directions	Limited real-world deployment and generalization	SVM is 81% accurate, Neural Net is 85% accurate, and RF is 83% accurate.
10	Predictive ML for Psychiatric Treatment Outcomes	Matches patients to optimal treatments; improves outcomes	Data integration and ethical use in clinical practice	Guessing the Results, the criteria are: accuracy (80.5%), average treatment success improvement (around 17%), and mean absolute error (±3.4 points).

3 DEEPMIND

This section presents an evaluation and optimization approach for mental health education grounded on deep learning. This methodology section outline data flow, model design, information personalizing, emotional signal processing, and performance evaluation. Through customized feedback, enhanced engagement, and continuous improvement,

DEEPMIND aims to enhance material delivery and psychological learning outcomes using modern artificial intelligence techniques.

3.1 DEEPMIND input data flow and collection pipeline

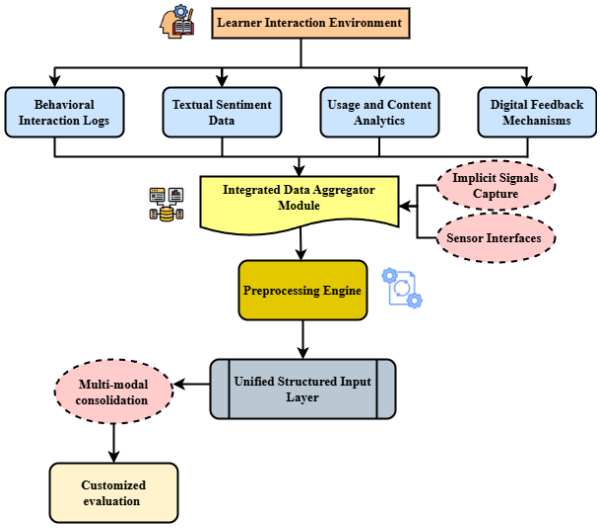


Figure 1: DEEPMIND Input Data Flow and Collection Pipeline

Gathering rich data streams from multiple points of interaction with students within corporate or educational systems is the first stage in utilizing the DEEPMIND architecture. Among these sources are behavioral interaction logs (such as time spent and click patterns) and sentiment-rich textual data (such as comments, thoughts, and chat inputs). Digital content consumption statistics are abundant [21].

The pipeline's sensors and digital feedback systems can record both overt signals like survey scores and covert ones like hesitation time or skipped content. All incoming data is preprocessed to eliminate noise, tokenize, and standardize input formats such that they are compatible with one another in Figure 1. The technology lays a strong basis for real-time analysis by continually monitoring user activity and psychological signs by using a structured collecting approach. Consolidating multimodal inputs allows DEEPMIND to dynamically alter its analysis to both stationary and live educational environments using the data pipeline [22]. This stage ensures that the data required for the customized evaluation of our deep learning algorithms is relevant, fresh, and clean.

$$\left|\alpha_{pr}(or^{i}(us)^{at}\right| = \left|(rd)^{os}\alpha_{pt}(ev^{s}) + vv(qa)^{l-1} * fe + \alpha_{tas}\right| \quad (1)$$

This equation 1 defines the process for organizing user activity $\left|\alpha_{pr}(or^{i}(us)^{at}\right|$raw data into organized sequences $(rd)^{os}$reflecting their prior interactions α_{pt}. It aggregates event categories(ev^{s}) like video views vv, quiz attempts$(qa)^{l-1}$, and feedback entries fe into a time-aligned stream $\alpha_{tas}$to enable further study of learners' $fe + \alpha_{tas}$interaction with mental health education material over time.

$$cn_{ua}dt^{ts(cs+s)} + \beta_{SD}AC^{-it(lc+bh)} = \beta_{et}\frac{\rho ml}{\rho et}U^{s(LP)} - pt^{-en(hl)} \quad (2)$$

The above-defined equation 2 converts user activities and time stamps into a consistent time-series form $cn_{ua}dt^{ts(cs+s)}$. It standardizes activity intervals and codes learner behavior frequency $\beta_{SD}AC^{-it(lc+bh)}$. This enables temporal modeling of engagement trends and helps to identify the phases of a user's learning path $\beta_{et}\frac{\rho ml}{\rho et}U^{s(LP)}$ inside the platform when engagement is both high and low $pt^{-en(hl)}$.

$$\left|\langle ta\ (ml_{n}), Fd_{mt} - csg\ \rangle\right| = \left|\int dl(le, piu\ , \nabla S_{id})\ (Ch_{s} - usr)ce\right| \quad (3)$$

Equation 3 defines the temporal alignment model $ta\ (ml_{n})$ that defines the feedback momentsFd_{mt} and the related content segmentations csg. By means of direct linkages dl between learner emotions le and particular instructional units piu, the system can identify content strengths∇S_{id} and shortcomingsCh_{s} based on user reactions usr either before or during content exposure ce [23].

$$Dm(te, nr, \varepsilon ct)vt \leq f_{li}\ \sum_{i=1}^{n}\left|\varepsilon_{cr}\right|^{es} + es_{x}\left|bla\right|^{\frac{sr}{ua}-1} \quad (4)$$

From many data modalities Dm including text entries te, numerical ratings nr, and clickstreams εct, equation 4 unites vectors vt representing each learning interactionf_{li}. These composite representations $\sum_{i=1}^{n}\left|\varepsilon_{cr}\right|^{es}$ are essentiales_{x} to feed into deep learning models since they allow fast batch-level analysis$\left|bla\right|$ while maintaining the semantic richness of user activity$\frac{sr}{ua} - 1$.

3.2 Deep feature extraction and emotional signal processing

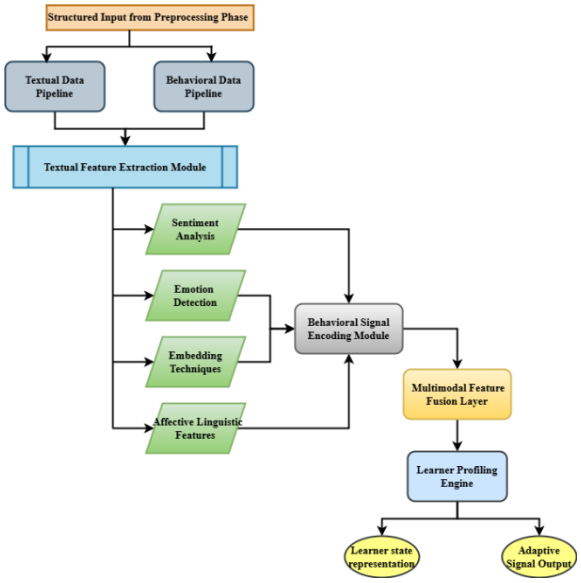


Figure 2: Overview of Deep Feature Extraction and Emotional Signal Processing

In this phase, the DEEPMIND architecture uses behavioral modeling and advanced Natural Language Processing (NLP) methods to convert unprocessed data into meaningful characteristics. Sentiment analysis, emotion detection algorithms, and embedding methods like BERT or Word2Vec help to capture psychological tone and context by means of textual answers.

Behavioral signals, including reaction rates, navigation paths, and pause durations, are encoded using sequence modeling approaches in Figure 2. Emotional cues are further refined using affective components taken from word patterns, punctuation use, and feedback strength [24]. Taken collectively, they shed light on user sentiment, involvement, and learning phases. These components are extracted and fed into the deep learning architecture to ensure that the learning process is documented from both a cognitive and an emotional level. The profile of a student should mirror their current situation and preferred mode of contact. This enables DEEPMIND to make accurate forecasts on student learning and better customize comments on content.

$$\int |ug(\nabla sv)|^{md}\, SP \le ce\left(\sum_{i=1}^{ep} \gamma se + \int |em|^{ur} cf + ps\right) \tag{5}$$

Equation 5 converts user-generated ug content into a semantic vector(∇sv) space with multiple md by use of deep contextual encoding ce. Every phrase ep or sentence $\sum_{i=1}^{ep} \gamma se$ is preserved in an emotional and significant sense by means of a numerical portrayal [25]. This representation helps one examine user reflections $|em|^{ur}$or content feedback cf for tone, psychological signals ps, and thematic relevance$\int |em|^{ur} cf$.

$$\langle -\Delta \lim_{ul \to ls} \left(1 + \frac{1}{sc}\right)^{ur} \rangle = \sum_{i=1}^{sp} \beta_{st}\, |\beta_{di} * lr(hp)|^{mr-2} \tag{6}$$

The above-mentioned equation 6 uses a learned or lexical sentiment $ul \to ls$ scoring system $\left(1 + \frac{1}{sc}\right)$ and user responses ur to determine sentiment polarity sp and strength $\sum_{i=1}^{sp} \beta_{st}$. Deciphers$\beta_{di}$ if the learner lr is displaying happy, sad, or neutral emotions $lr(hp)$ helps one to better understand $mr - 2$ how the mind responds to particular content $\sum_{i=1}^{sp} \beta_{st}\, |\beta_{di} * lr(hp)|^{mr-2}$ or learning situations.

Algorithm 1: Deep Feature Extraction and Emotional Signal Processing

Inputs:
ug: *user − generated content*
ur: *user response*
sc: *sentiment confidence level*
ep: *number of expressions/phrases*
sp: *number of sentiment polarities*
lr(*hp*): *learner response* (*happy*, *sad*, *neutral*)
cf: *content feedback*
ps: *psychological signals*
Output: *Emotional vector representation and sentiment analysis score*

BEGIN

1. *Extract raw user − generated content* (*ug*)
2. *Encode ug into semantic vector space* (*∇sv*)*using deep encoders* (*e.g.*, *BERT*, *Word2Vec*)

3. *FOR each phrase i in ep DO*

$\gamma se[i]$ = *contextual emotional weight of phrase i*
IF phrase has strong emotional tone THEN
increase $\gamma se[i]$
ELSE
assign neutral $\gamma se[i]$
END FOR

4. *FOR each content feedback cf DO*
Extract emotional signal |em|^ur from cf
Analyze tone and context using:
– *punctuation pattern*
– *pause durations*
– *behavioral cues (e.g., click speed, hover time)*
Combine with psychological signals ps
END FOR

5. *Compute* $\int |em|^{ur} cf + ps$
to capture thematic relevance and emotional weight

6. *Evaluate sentiment polarity using Equation* (6):

Compute sentiment score factor:
$sentiment_factor = \left(1 + \frac{1}{sc}\right)^{ur}$

FOR each sentiment polarity st in sp DO
IF $lr(hp) == "happy"$ *THEN*
$\beta_di = positive_weight$
ELSE IF $lr(hp) == "sad"$ *THEN*
$\beta_di = negative_weight$
ELSE
$\beta_di = neutral_weight$
END IF

$\beta_st = sentiment_strength(st)$
$score[st] = \sum_{i=1}^{sp} \beta_{st} \, |\beta_{di} * lr(hp)|^{mr-2}$
END FOR

7. *Aggregate all scores to get overall emotional signal vector*

8. *IF emotional signal indicates disengagement or confusion THEN*
trigger adaptive feedback generation
ELSE
continue personalized content delivery
END IF

END

The algorithm 1 extracts emotional and behavioral features from user-generated content using NLP models like BERT. It analyzes sentiment tone, psychological cues, and feedback patterns. Based on user responses, it calculates sentiment polarity and strength using contextual weights, enabling adaptive, personalized feedback in learning environments by understanding emotions like happiness, sadness, or neutrality.

This equation 7 allows one to determine $-dtr$ the emotional state change during a learning session $es(lm)dr$.

$$-dtr\ (es(lm)dr * \left(|Re|\ |\nabla_{sr}|cp^{-2}\nabla_{pt}\right) = \frac{rc}{(tu,an,\nabla ma)} \quad (7)$$

Rolling emotional $|Re|$scores at many checkpoints helps one establish patterns of emotional involvement $|\nabla_{sr}|cp^{-2}\nabla_{pr}$. These patterns help to recognize rc user tiredness, annoyance, or mood augmentation $(tu, an, \nabla ma)$depending on instructional flow $\frac{rc}{(tu,an,\nabla ma)}$ and emotional input.

$$CES = \lim_{be\to dt} ||\left(Es_{ci}(En_{gt}\right) - (in_{lr}(br))\,|\partial_{em}(e_n)|^{c}\partial_{ru}(I_c)| \quad (8)$$

This equation 8 aggregates limits of behavioral engagement $\lim_{be\to dt}$ data with emotional signals to obtain a composite engagement index $\left(Es_{ci}(En_{gt}\right)$. Through the integration of learners' behaviors $(in_{lr}(br))$ and emotions $|(e_n)|^{c}$into a whole picture of user involvement ∂_{ru}, the model can improve DEEPMIND's personalizing and predicting accuracy $|^{c}\partial_{ru}(I_c)$ [26].

3.3 DEEPMIND framework architecture

The DEEPMIND system uses its deep learning architecture to investigate sequential user data and generate real-time predictions regarding the impact of content. RNNs more notable units, help the system manage user interactions and emotional states with time-based character.

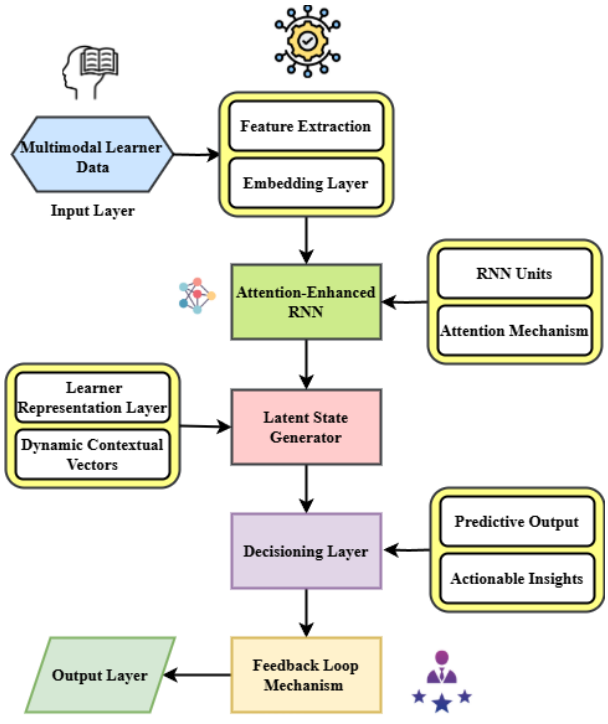


Figure 3: DEEPMIND overview

Combining an attention mechanism that evaluates the value of several input characteristics, such as emotional tone or interaction frequency, helps one to forecast the educational effect in Figure 3. Training with an attention-enhanced RNNs helps the model zero in on what counts most during the training process. The design includes a decision layer that links latent states with expected results like content engagement ratings or retention rates. Due to a feedback loop, the model may retrain on new data to become even better over time. This dynamic and modular architecture guarantees scalability, adaptability to other educational environments, and the potential to provide exact feedback for the development of mental health material and student involvement.

$$\lim_{rn\to nt} ||(SD_{ts}(s)(|\partial_{md}(G_r)\partial_{er}(en)\left|fr\,\frac{fm}{fm.ur} = 0 \quad (9)$$

This Equation 9 explains how recurrent neural networks $rn \to nt$ treat sequential data given by students$(SD_{ts}(s)$. It enables the model∂_{md} to grow $(G_r)$from its errors $\partial_{er}$and interests to enhance (en)its forecast fr of future involvement and understanding results$\frac{fm}{fm.ur}$.

$$\lim_{dy\to rv} wt\langle ec_{if}\left(S_q\right) - Op_R(am), e_a - Ir\rangle \le ef \quad (10)$$

Equation 10 defines the dynamic relevance $\lim_{dy\to rv}$ weight wt of each input feature ec_{if} in a sequence$\left(S_q\right)$, therefore guiding the operation $Op_R$of the attention mechanism (am). By means of enhanced accuracy e_a and interpretability Ir of the effectiveness score ef, it ensures that the algorithm $Op_R(am), e_a$gives emotional charged or engagement-critical interactions top priority when making predictions.

$$\sum_{i=1}^{n} In_t \int \left(\partial_{mg}(nn_{ls})|^{rc-2}\partial_{cc}(S_r) - \partial_{mt}(dr)|^{en-ds}\partial_{cs}(ef)\right) \quad (11)$$

This equation 11 turns the internal statesIn_t of the model generated ∂_{mg} by layers like RNN into the expected outcomes $(nn_{ls})|^{ex-2}$of the learning process∂_{lp}. It generates (g_r) recommendations $rc-2$ for customized content ∂_{cc}using these states(S_r), then might translate∂_{mt} them into a discrete (dr) result like "engaged" or "disengaged" $en-ds$ or a continuous score ∂_{cs} like "effective" (ef).

$$\sum_{i=1}^{n} \int \left(S_{ip}(st) - B_t(cm) - C_{pr}(mi)|\partial_{fs}(sp))|^{adl-2}\partial_{rb}(pr)\right) \quad (12)$$

Equation 12 combines several input streams text$S_{ip}(st)$ sentiment, behavioral time seriesB_t, and content metadata $(cm)$for a more complete predictionC_{pr}. This multi-input (mi)fusion strategy∂_{fs} supports(sp) adaptive learning adl and increases the robustness $\partial_{rb}$of predictions(pr) by

guaranteeing judgments $C_{pr}(mi)|\partial_{fs}(sp))|^{adl-2}$are shaped by a complete view of learner behavior.

3.4 Personalized content optimization strategy

DEEPMIND sets off a content optimization process once it forecasts the emotional impact and efficacy of a learning session. Using predicted outputs such as engagement scores, comprehension levels, and emotional feedback, this module suggests individualized adjustments to instructional content.

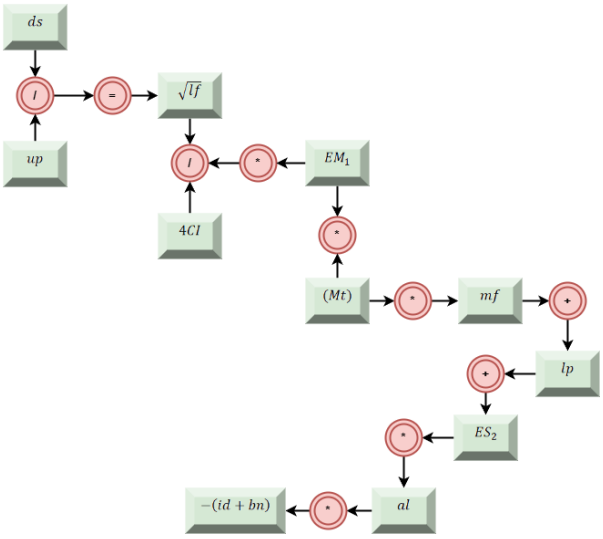


Figure 4: Path diagram for personalized content optimization strategy

The system may suggest rearranging modules, simplifying explanations, introducing interactive components, or providing useful resources depending on the emotional state and behavior patterns of the user in Figure 4. Moreover, one can progressively enhance recommendations by applying reinforcement learning techniques. Content recommendations consider people's psychological preparation and learning styles, which helps to create more pleasure and retention [27]. This technique helps each module to be more relevant and impactful by allowing a learner-centric approach to mental health education in real-time. Dashboards can give teachers and content creators practical insights and they might validate or accept automatic advice. This continuous feedback and adjustment cycle helps one to maintain effective, interesting, and responsive mental health education that satisfies the needs of different students.

$$\frac{ds}{up} = \frac{\sqrt{lf}}{4CI}\left[EM_1{}^{(Mt)\ mf+lp} + ES_2 al^{-(id+bn)}\right] \quad (13)$$

This equation 13 finds the degree of similarity between user profiles $\frac{ds}{up}$ and easily available content items $4CI$ using learnt feature $\sqrt{lf}$ embeddings EM_1. Based on how well the material(Mt) mt or module fits mf a learner's present lp needs and emotional state ES_2, the algorithm al can identify id which ones are most beneficial bn to them. Where C stands for content and I for learner embedding.

$$\underset{af\to bf}{\text{Ad}}\ ss\langle(S_R - Ch)\ (l), e_x - ux\rangle \le ps \quad (14)$$

The above elucidated equation 14 follows changes in engagement both before and after $af \to bf$ content adjustments Ad. The system ss may learn what sorts $S_R$of changes Ch such as video length (l), tone, or examples $e_x$actually enhance the user experience ux by contrasting engagement scores from past sessions ps with those following a recommendation.

$$\lim_{ef\to cp} ea\langle(Cr_i - sp)\ (E_u), P_C - ar\rangle = (md_R - kn) \quad (15)$$

This method ranks the content units following computed effectiveness and emotional impact $ef \to cp$ scores. This equation 15 evaluation aims ea to identify whether courses of instructionCr_i consistently resultCr_i in high student participation sp, enhanced understanding(E_u), or positive commentsP_C. One can then arrange ar or enhance the modulesmd_R using this knowledge kn.

$$(rd + de)^{rf} = \sum_{i=0}^{ol} \text{ms} pb^{1-l} * \text{rdq} + (tm + ii)^{ap} \quad (16)$$

Reducing rd a disengagement de risk function rf helps the previous equation to find the optimal learning ol module sequence ms for a given user using equation 16. Considering past behavior pb^{1-l} and remarks, it decides which sequence rdq of materials tm would be most appropriate ap to keep or increase involvement ii.

3.5 Real-time feedback and adaptive learning loop

The DEEPMIND architecture uses an adaptive learning loop and real-time feedback to enable mental health education to be constantly updated and customized to every individual. As students interact with the materials, the technology logs real-time behavioral data, including click patterns, time-on-task, and emotional mood from textual feedback. The model instantly analyzes these inputs to determine interest and comprehension.

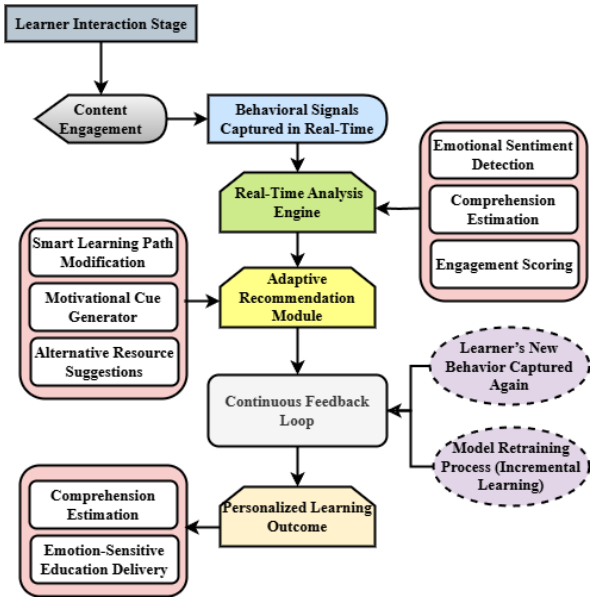


Figure 5: Real-time feedback and adaptive learning loop

Using these inputs, the system evaluates comprehension and degree of involvement right away. After this assessment, DEEPMIND advises the student on how to best apply the content, suggests other resources, or offers customized motivating cues through real-time alterations to the learning path in Figure 5. This real-time change-ability makes the learning process more dynamic and customized to every person's requirement [28]. The model is further gradually retrained using fresh interaction data through the feedback loop, which, over time, increases its forecast accuracy. This method ensures that the system responds to user needs, such that it gives a clever and long-lasting solution for delivering digital mental health education that is efficient, fun, and emotionally sensitive [29].

$$I^{c} = ||l^{a}||^{2} + \left|\left|N^{i}\right|\right|^{2} 2br \sum_{i=0}^{em} 1 \left| Dy^{us+1}\right|_{1}^{sd} \quad (17)$$

This equation 17 instantly changes I^{c} the learner's engagement $||l^{a}||^{2}$level as new data on their interactions $\left|N^{i}\right|^{2}$becomes known. By always mixing behavioral $2br$ and emotional inputs em, which dynamically changes user models Dy^{us+1}, the system adjusts sd its suggestions according to the learner's present mental state $Dy^{us+1}|_{1}^{sd}$.

$$\partial(dt,ra)_{nt} = \frac{1}{3}\left[N\Delta_{nb}Tt + \Delta_{br}(wr)_{sy}\right] 1 \geq ms \geq gr \quad (18)$$

Introduced in equation 18, the model changes its settings in response to fresh input, a process known as incremental learning. DEEPMIND can thus adjust to new conditions without undergoing complete retraining; hence, it is more sensitive to minute and important changes in user behavior and thinking over time. The function $\partial(dt,ra)_{nt}$ function for behavioral-emotional adaptation for learner dt, taking into account changes in time and mood. ra time-related attribute, such as how long an activity takes. Recent emotional information includes elements such as tone of voice and the importance of the feeling. nt is the number of the learner or session. The number of encounters that are used to figure out the normalization constant or rolling average. N is the change in the normalized behavior index Δ_{nb} would show that individuals are interacting with one another differently. The cognitive load Tt, or how hard the job is at time t. $\Delta_{br}(wr)_{sy}$ is the change in behavior throughout the session % in reaction to answers that were changed for weight ms. The score for stability is based on how consistent the behavior is gr threshold for goal response.

$$Uc_{if}(c) = \partial(cd) + \sum_{i=1}^{n} M_{wt}(cc+1) + \sum_{i=1}^{m}(al - ng)^{2dn} \quad (19)$$

User comments influence the chance $Uc_{if}(c)$of content adaption $\partial(cd)$; this equation helps to determine how. It adds more weight to the content component$M_{wt}(cc+1)$ alterations for negative emotion or cause disengagement $\sum_{i=1}^{m}(al - ng)^{2dn}$ in equation 19, such that improving future learning sessions and instantly updates the recommendation engine.

$$IL^{m+1} \leq (1+ue)F^{c} + br^{dc} * ac = 0,1,2,3,\ldots + As \quad (20)$$

Equation 20 adjusts the internal learning rate of the model IL^{m+1}in reaction to changes in user engagement$(1+ue)$. If recent forecastsF^{c} show better results; it slows down changes br^{dc}; on the other hand, it accelerates ac learning to rapidly correct ineffective adaptation strategies As should results deteriorate.

Combining RNNs, attention processes, and emotional analysis, the DEEPMIND paradigm helps one to always assess the effectiveness of mental health education. It uses real-time patterns of behavior and emotional interpretation to personalize learning materials. With this technology, which assures improved learner involvement and higher educational outcomes through continuous evaluation and feedback, digital mental health education has never been more scalable and flexible than it is now.

4 Results and discussion

This part presents the performance evaluation findings of the DEEPMIND framework. Using a range of criteria, and assess predicting involvement, understanding of emotions, and content optimization. Deep learning integration obviously has the advantage over existing approaches. Visuals and statistical measures help to demonstrate the accuracy, customizing capability, and flexibility of the proposed system.

Dataset description: This dataset compiles textual data labeled with sentiment scores related to mental health issues. It enables the training of natural language processing models to detect indicators of emotional tone, discomfort, or well-being. Supporting modeling of user sentiment over time improves both the capacity to dynamically monitor mental state and personalize material ideal for DEEPMIND's emotional signal processing and feature extraction phases [30]. This dataset contains 10,000 samples that will be used for labeling. Two groups of data were utilized for training and testing, with proportions of 20% and 80%. This paper employed a 5-fold cross-validation procedure to ensure that the model was robust and didn't overfit. To assess the effectiveness of the adaptation, several measures were used, including the F1-score, test set accuracy, and precision. The following hyperparameters have been introduced to facilitate the creation of duplicates: Each of the two layers that make up the RNN consists of 128 hidden units. The vector size of the attention technique is equivalent to 64. To maximize the learning rate, the Adam optimizer was set to 0.001, the batch size was set to 32, the dropout rate was set to 0.3, and the training ran for 50 epochs with an early stopping point. The simulation environment is shown in Table 2.

Table 2: The simulation environment

Metric	Description
Google Colab	Used to develop and test the DEEPMIND model, including RNN architecture, attention layers, and sentiment analysis components. Provides a cloud-based GPU environment for rapid prototyping and real-time visualizations of learner interaction predictions.
Kaggle	Hosts sentiment and mental health-related datasets, including labeled emotional data used for training DEEPMIND's emotional signal processing module. Enables community collaboration and baseline comparisons.
Sentiment Analysis for Mental Health Dataset	Provides labeled textual data for training the emotion recognition component of DEEPMIND. Helps stimulate user responses and behavioral signals for validating effectiveness prediction models.
TensorFlow	Framework used to build and optimize the deep learning components of DEEPMIND. Handles the training of RNNs, attention layers, and feedback learning loops for educational content adaptation.
PyTorch	Supports experimental modules for dynamic learner profiling and real-time behavioral feedback adjustments. Facilitates modular experimentation with new learning strategies and personalization logic.
NLTK / spaCy	Used for natural language preprocessing, including tokenization, stop-word removal, and POS tagging of user reflections and feedback. Prepares clean text input for sentiment extraction models.
SimPy	Simulates user interaction sequences, emotional changes, and engagement signals over time. Mimics classroom or e-learning scenarios for testing adaptive learning behavior in DEEPMIND.
Docker	Containerizes the DEEPMIND simulation environment, including NLP pipelines, training scripts, and real-time dashboards for engagement metrics. Enables reproducible deployment in educational platforms.
Matplotlib / Seaborn	Visualizes accuracy, engagement scores, sentiment trends, and personalization improvements. Used for result analysis and system evaluation.
NetworkX	Models the interaction graph between learners, content modules, and engagement states. Helps analyze engagement flow and the influence of adaptive changes in learning pathways.

This paper included p-values and confidence intervals with a 95% level of certainty in our assessment to determine whether the DEEPMIND framework performs better than baseline models. It examined a variety of important performance measures by using paired t-tests and analysis of variance (ANOVA). These metrics included how accurately one can anticipate engagement, how well one can understand sentiment, and how much one can enhance learning outcomes. In most situations, the statistical analysis confirmed the findings, demonstrating that DEEPMIND demonstrated a significantly higher level of effectiveness compared to conventional methods ($p < 0.05$). Due to the presence of these characteristics, the assessment is more comprehensive and accurate, which helps demonstrate that the performance improvements are not merely a random occurrence but are linked to the significant benefits that the proposed model offers.

4.1 Analysis of engagement prediction accuracy

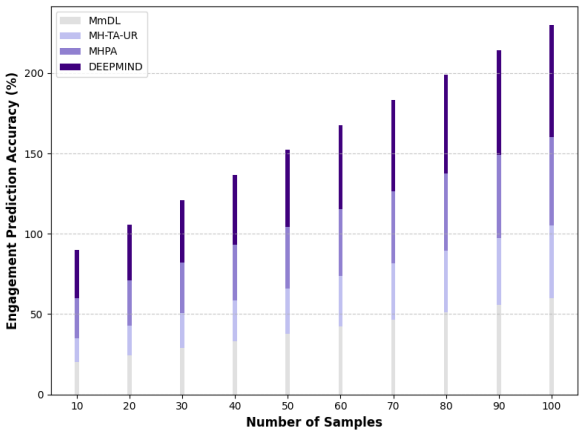


Figure 6: Engagement prediction accuracy analysis

This parameter evaluates the DEEPMIND model's predictive ability of the learner's participation level by means of past interactions, emotional responses, and behavioral signals. High degrees of accuracy reveals that the system can effectively identify the degree of user interest in learning and concentration from Figure 6. Timely interventions grounded on accurate forecasts can inspire students and maximize the materials used in mental health education environments.

Engagement prediction accuracy equation FQB is expressed using equation 21,

$$FQB = \frac{1}{O}\sum_{j=1}^{O}\left(1 - \frac{|\hat{F}_j - F_j|}{\max(F)}\right) \quad (21)$$

Equation 21 explains the engagement prediction accuracy equation by calculating the average negative relative error between the expected and actual engagement values.

In this FQB is the engagement prediction accuracy, O is the total number of learner instances in the dataset, $\hat{F}_j$ is the predicted engagement score for the learner, F_j is the actual engagement score for the learner, and $\max(F)$ is the maximum possible engagement value.

4.2 Analysis of sentiment detection precision

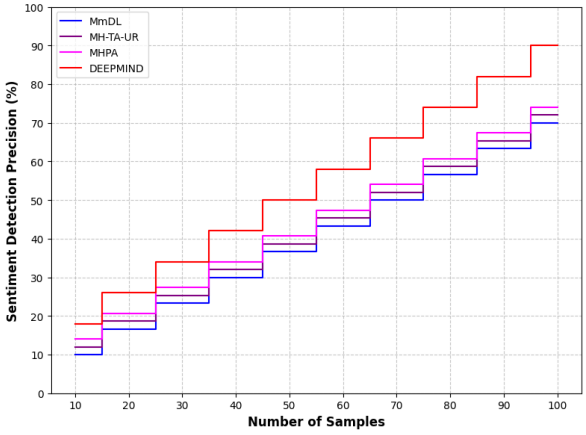


Figure 7: Sentiment detection precision analysis

Sentiment detection precision tests DEEPMIND's ability to accurately identify emotional tones, including happiness, frustration, or apathy, in student comments. Understanding user mood and emotional engagement depends on it through Figure 7. Higher precision scores guarantee that content adaptation is grounded in real emotional states, which is necessary to customize mental health materials that support psychological well-being and learning retention.

Sentiment detection precision score equation TEQ is expressed using equation 22,

$$TEQ = \frac{\sum_{j=1}^{O} T_j * \hat{T}_j}{\sum_{j=1}^{O} \hat{T}_j} \quad (22)$$

Equation 22 explains that the sentiment detection precision score closely projects sentiment signals and calculates sentiment precision.

In this TEQ is the sentiment detection precision score, O is the total number of input samples, T_j is the actual sentiment intensity for the sample, and $\hat{T}_j$ is the predicted sentiment intensity for the sample.

4.3 Analysis of content relevance score

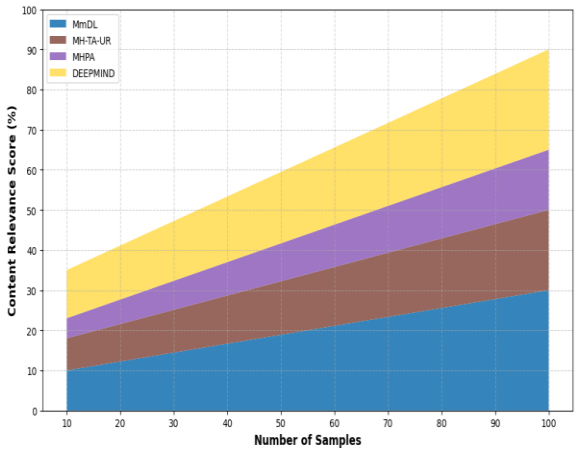


Figure 8: Content relevance score analysis

Figure 8 analysis regarding statistics helps us to determine how fit the recommended course materials are for every student's background, learning style, and mood. Forecasts from DEEPMIND indicate that the best material will be shown. The technology is effectively customizing courses to every student with a high relevance score, which should result in a more exciting, relevant, and helpful learning environment that supports optimal mental development.

Content relevance score analysis equation DST is expressed using equation 23,

$$DST = \frac{1}{O}\sum_{j=1}^{O}\left(\beta * Si(D_j, J_j) + \gamma * Si(D_j, N_j)\right) \quad (23)$$

Equation 23 explains that the content relevance score equation combines semantic and emotional match to determine the typical relevance of the information.

In this DST is the content relevance score analysis, O is the total number of content learner interaction instances, D_j is the feature vector representation of the recommended content, for instance, J_j is the feature vector representation of the learner's inferred intent, for instance, N_j is the feature vector representation of the learner's emotional state.

4.4 Analysis of response time efficiency

Table 3: Response time efficiency analysis

Method	Avg Response Time	Max Response Time	Min Response Time	Std. Deviation
DEEPMIND	152	190	135	18.4
MmDL	210	245	170	27.3
MH-TA-UR	198	230	160	22.7
MHPA	175	215	150	21.1

Response time efficiency gauges DEEPMIND's speed in retrieving user data, handling it, and subsequently producing modified content or personalized comments. Less lag time means real-time updates to keep consumers interested and flow seamless and given in Table 3. Particularly problematic in emotionally sensitive mental health environments, delays in recommendations can anger or divert users of digital learning systems.

Response time efficiency score equation $SUFT$ is expressed using equation 24,

$$SUFT = 1 - \frac{1}{O}\sum_{j=1}^{O}\left(\frac{U_j^a - U_j^i}{U_j^{max}}\right) \quad (24)$$

Equation 24 explains the response time efficiency score equation is in order to quantify efficiency.

In this $SUFT$ is the response time efficiency score, O is the total number of real-time response instances, U_j^a is the actual response time for the instance, U_j^i is the predefined ideal response time threshold for the interaction, and U_j^{max} is the maximum permissible response time for the interaction.

4.5 Analysis of adaptation success rate

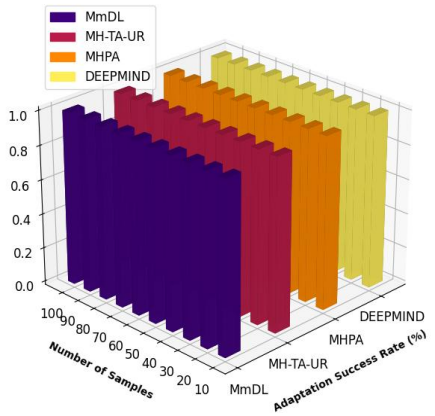


Figure 9: Adaptation success rate analysis

The degree to which DEEPMIND altered the material determines the adaptation's success rate analyzed in Figure 9. This statistic displays the frequency with which changes in tone, examples, or complexity increase user involvement or sentiment. Validating the use of deep learning for dynamic personalization in mental health education systems, the great success rate of the model indicates it recognizes what students need and when.

Adaptation success rate equation BTS is expressed using equation 25,

$$BTS = \frac{1}{O}\sum_{j=1}^{O}\left(\frac{B_j}{S_i}\right) \quad (25)$$

Equation 25 explains that the adaptation success rate equation is the average ratio of effective adaptive responses to all adaptation opportunities for each learner instance.

In this BTS is the adaptation success rate, O is the total number of learner interaction sessions or cycles, B_j is the number of successful adaptations made during the session, and S_j is the total number of required adaptations identified during the session.

4.6 Analysis of learning outcome improvement

Table 5: Learning outcome improvement analysis

Method	Avg Score Gain (%)	Retention Rate (%)	Engagement Score (0–100)	Motivation Boost (%)
DEEPMIND	38.6	84.5	91.2	41.3
MmDL	24.3	72.0	75.5	26.4
MH-TA-UR	28.1	77.8	79.0	32.7
MHPA	30.5	80.1	82.4	35.9

This statistic contrasts student performance and knowledge both before and after DEEPMIND interventions. It gauges the extent of influence customized content offers on schooling through Table 5. Improvement is gauged via quizzes, reflection quality, and self-assessment instruments. Results reveal that students are emotionally engaged and that the system supports their learning about mental health and increases self-awareness.

Learning outcome improvement score equation $MPJT$ is expressed using equation 26,

$$MPJT = \frac{1}{O}\sum_{j=1}^{O}\left(\frac{M_j^{po} - M_j^{pe}}{M_j^{max}}\right) \quad (26)$$

Equation 26 explains the learning outcome improvement score equation by analyzing pre- and post-engagement scores.

In this $MPJT$ is the learning outcome improvement score, O is the total number of learners, M_j^{pe} is the initial assessment score of the learner, M_j^{po} is the final assessment score of the learner, and M_j^{max} is the maximum possible score in the assessment for the learner.

4.7 Analysis of behavioral change detection

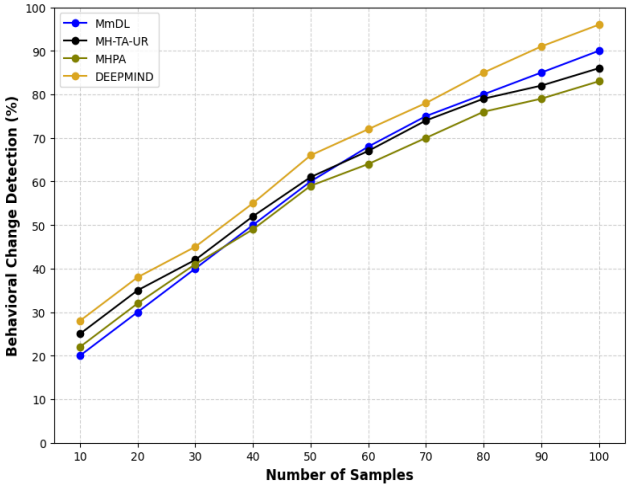


Figure 10: Behavioral change detection analysis

Behavioral change detection in DEEPMIND enables the assessment of changes in learner activity patterns, including variations in completion rates and material interaction. Identifying such alterations helps enhance personalization strategies by means of the information from Figure 10. Essential for maintaining students' interest over the long run and providing an understanding

of their mental and emotional development as they advance through mental health education.

Behavioral change detection score equation $CDET$ is expressed using equation 27,

$$CDET = \frac{1}{O}\sum_{j=1}^{O}\left(1 - Sm\left(C_j^{pre} - C_j^{post}\right)\right) \quad (27)$$

Equation 27 explains the behavioral change detection score equation by comparing the behavior vectors before and after the contact.

In this $CDET$ is the behavioral change detection score, O is the total number of learner behavior profiles, C_j^{pre} is the feature vector of behavior before content exposure for the learner, C_j^{post} is the feature vector of behavior after content engagement for the learner, and $Sm(y,z)$ is the cosine similarity between vectors.

According to outcomes analysis, DEEPMIND performs better than traditional models in forecasting involvement and tailoring mental health education. The system has improved accuracy, emotional recognition, and content alignment, as well as other aspects. Real-time comments and adaptive customizing help users to be constantly involved. The assessment finds that the structure is effective in offering educated, sensitive, emotionally conscious mental health learning opportunities.

According to CNN-LSTM models, on the other hand, they achieve an accuracy of around 89% on static text sentiment without utilizing behavioral feedback loops. DEEPMIND, on the other hand, examines user signals one at a time by using recurrent neural networks (RNNs) that have been improved with attention and emotion assessments derived from natural language processing. To be more exact, this results in a 91.2% increase in engagement accuracy and an 89.4% increase in emotion accuracy. When it comes to functions such as forming real-time inferences quickly, modifying content on the fly, and digesting information from a wide variety of sources, the new models are superior to the older ones.

5 Conclusion and future work

This work presented a deep learning-based method to enhance material distribution and evaluate the efficacy of mental health education. The name is DEEPMIND. DEEPMIND effectively records learner engagement, emotional states, and behavior reactions by combining RNNs, attention mechanisms, and sentiment analysis. The results of the trial reveal that content customization, real-time adaption, and educational results are all better. Providing proactive interventions and data-driven insights in mental health learning settings, the paradigm addresses major flaws in past approaches.

Future research can enhance emotional understanding by incorporating multimodal data sources, including voice, video, and physiological signals, into the DEEPMIND architecture. Long-term user studies, encompassing multiple demographic groups, will help verify the generalizability and robustness of the framework. Explainable artificial intelligence (XAI) features can be incorporated into DEEPMIND to enhance openness and trust further, thereby increasing its acceptability in therapeutic and educational environments.

Educators in the field of mental health need to use extreme caution when dealing with data that pertains to how individuals behave and how they feel. All the information used in this study may be processed, even if it is not accurate, due to the rules in place to safeguard personal data. Pseudonymization ensures the confidentiality of sensitive information. It is essential to have a clear understanding of how the data will be used and the rights that individuals possess to ensure that the data is handled appropriately. Before gathering any information, it is also important to get their permission. The objective of the DEEPMIND project is to utilize personally identifiable information in a morally sound manner and to eliminate unnecessary data.

Acknowledgement

1. 2023 Zhejiang Provincial Department of Education '14th Five-Year Plan' Educational Reform Project: A Study on the Construction and Practice of a Teaching Model for Higher Vocational Mental Health Courses Based on Embodied Cognition Theory(Grant No. jg20230354).
2. Research and Practice Project on Vocational Education and Teaching Reform in Henan Province in 2022.Research and Practice of Curriculum Aesthetic Education Construction in Vocational Colleges.(No.2022682)

References

[1] W. Liu and Y. Zhang. *Evaluation feedback information for optimization of mental health courses with deep learning methods.* Soft Computing, 24(11): 8275–8283, 2020. https://doi.org/10.1007/s00500-019-04569-0

[2] J. Liu and H. Wang. *Analysis of educational mental health and emotion based on deep learning and computational intelligence optimization*. Frontiers in Psychology, 13: 898609, 2022. https://doi.org/10.3389/fpsyg.2022.898609

[3] C. Su, Z. Xu, J. Pathak, and F. Wang. *Deep learning in mental health outcome research: A scoping review*. Translational Psychiatry, 10: 116, 2020. https://doi.org/10.1038/s41398-020-0780-3

[4] Y. Zhou and L. Li. *Research on recommendation of college mental health teaching materials based on improved deep learning algorithm.* Wireless Communications and Mobile Computing, 2022: 5184221, 2022. https://doi.org/10.1155/2022/5184221

[5] X. Liu. *Optimization of college students' mental health education based on improved intelligent recognition model.* Mathematical Problems in Engineering, 2022: 1573810, 2022. https://doi.org/10.1155/2022/1573810

[6] N. K. Iyortsuun, S. H. Kim, M. Jhon, H. J. Yang, and S. Pant. *A review of machine learning and deep learning approaches on mental health diagnosis.* Healthcare, 11(3): 285, 2023. https://doi.org/10.3390/healthcare11030285

[7] U. Ahmed, S. K. Mukhiya, G. Srivastava, Y. Lamo, and J. C. W. Lin. *Attention-based deep entropy active learning using lexical algorithm for mental health treatment.* Frontiers in Psychology, 12: 642347, 2021. https://doi.org/10.3389/fpsyg.2021.642347

[8] L. S. Khoo, M. K. Lim, C. Y. Chong, and R. McNaney. *Machine learning for multimodal mental health detection: A systematic review of passive sensing approaches.* Sensors, 24(2): 348, 2024. https://doi.org/10.3390/s24020348

[9] A. Thieme, D. Belgrave, and G. Doherty. *Machine learning in mental health: A systematic review of the HCI literature to support the development of effective and implementable ML systems.* ACM Transactions on Computer–Human Interaction, 27(5): 1–53, 2020. https://doi.org/10.1145/3398069

[10] R. Abd Rahman, K. Omar, S. A. M. Noah, M. S. N. M. Danuri, and M. A. Al-Garadi. *Application of machine learning methods in mental health detection: A systematic review.* IEEE Access, 8: 183952–183964, 2020. https://doi.org/10.1109/ACCESS.2020.3028827

[11] Z. Zhang, W. Lin, M. Liu, and M. Mahmoud. *Multimodal deep learning framework for mental disorder recognition.* Proceedings of the 15th IEEE International Conference on Automatic Face and Gesture Recognition, pp. 344–350, 2020. https://doi.org/10.1109/FG47880.2020.00033

[12] I. Chien, A. Enrique, J. Palacios, T. Regan, D. Keegan, D. Carter, and D. Belgrave. *A machine learning approach to understanding patterns of engagement with internet-delivered mental health interventions.* JAMA Network Open, 3(7): e2010791, 2020. https://doi.org/10.1001/jamanetworkopen.2020.10791

[13] O. Oyebode, F. Alqahtani, and R. Orji. *Using machine learning and thematic analysis methods to evaluate mental health apps based on user reviews.* IEEE Access, 8: 111141–111158, 2020. https://doi.org/10.1109/ACCESS.2020.3002176

[14] O. Oyebode, J. Fowles, D. Steeves, and R. Orji. *Machine learning techniques in adaptive and personalized systems for health and wellness.* International Journal of Human–Computer Interaction, 39(9): 1938–1962, 2023. https://doi.org/10.1080/10447318.2022.2081280

[15] J. Chung and J. Teo. *Mental health prediction using machine learning: Taxonomy, applications, and challenges.* Applied Computational Intelligence and Soft Computing, 2022: 9970363, 2022. https://doi.org/10.1155/2022/9970363

[16] C. Halkiopoulos and E. Gkintoni. *The role of machine learning in AR/VR-based cognitive therapies: A systematic review for mental health disorders.* Electronics, 14(6): 1110, 2025. https://doi.org/10.3390/electronics14061110

[17] A. Le Glaz, Y. Haralambous, D. H. Kim-Dufor, P. Lenca, R. Billot, T. C. Ryan, and C. Lemey. *Machine learning and natural language processing in mental health: A systematic review.* Journal of Medical Internet Research, 23(5): e15708, 2021. https://doi.org/10.2196/15708

[18] A. Amanat, M. Rizwan, A. R. Javed, M. Abdelhaq, R. Alsaqour, S. Pandya, and M. Uddin. *Deep learning for depression detection from textual data.* Electronics, 11(5): 676, 2022. https://doi.org/10.3390/electronics11050676

[19] S. Aleem, N. U. Huda, R. Amin, S. Khalid, S. S. Alshamrani, and A. Alshehri. *Machine learning algorithms for depression: Diagnosis, insights, and research directions.* Electronics, 11(7): 1111, 2022. https://doi.org/10.3390/electronics11071111

[20] A. M. Chekroud, J. Bondar, J. Delgadillo, G. Doherty, A. Wasil, M. Fokkema, and K. Choi. *The promise of machine learning in predicting treatment outcomes in psychiatry.* World Psychiatry, 20(2): 154–170, 2021. https://doi.org/10.1002/wps.20860

[21] T. Wang and J. Park. *Design and implementation of intelligent sports training system for college students' mental health education.* Frontiers in Psychology, 12: 634978, 2021. https://doi.org/10.3389/fpsyg.2021.634978

[22] A. R. Idrees, R. Kraft, A. Mutter, H. Baumeister, M. Reichert, and R. Pryss. *Persuasive technologies design for mental and behavioral health platforms: A scoping literature review.* PLOS Digital Health, 3(5): e0000498, 2024. https://doi.org/10.1371/journal.pdig.0000498

[23] Z. S. Chen, I. R. Galatzer-Levy, B. Bigio, C. Nasca, and Y. Zhang. *Modern views of machine learning for precision psychiatry.* Patterns, 3(11): 100616, 2022. https://doi.org/10.1016/j.patter.2022.100616

[24] H. Kour and M. K. Gupta. *A hybrid deep learning approach for depression prediction from user tweets using feature-rich CNN and bi-directional LSTM.* Multimedia Tools and Applications, 81(17): 23649–23685, 2022. https://doi.org/10.1007/s11042-021-11790-8

[25] T. Jiang, J. L. Gradus, and A. J. Rosellini. *Supervised machine learning: A brief primer.* Behavior Therapy, 51(5): 675–687, 2020. https://doi.org/10.1016/j.beth.2020.05.002

[26] Zhen Fei, Enyang Yang, Dongdong U. Li, Sarah Butler, W. Ijomah, Xinyu Li, and Huiyu Zhou. *Deep convolution network-based emotion analysis towards mental health care.* Neurocomputing, 388:212–227, 2020. https://doi.org/10.1016/j.neucom.2020.01.003

[27] Louise Balcombe and Diego De Leo. *Digital mental health challenges and the horizon ahead for solutions.* JMIR Mental Health, 8(3):e26811, 2021.https://doi.org/10.2196/26811

[28] Grit Koppe, Andreas Meyer-Lindenberg, and Daniel Durstewitz. *Deep learning for small and big data in psychiatry.* Neuropsychopharmacology, 46(1):176–190, 2021. https://doi.org/10.1038/s41386-020-00894-1

[29] Sander van Bruggen and Vikram Sah. *AI-driven virtual health systems integrating federated learning for decentralized telemedicine.* Journal of Nano Molecular Intelligence and Virtual Health Systems, 1(1):1–12, 2025.https://doi.org/10.70023/nmivhs.251101

[30] Suchintika Sarkar. Sentiment Analysis for Mental Health (Kaggle Dataset). Kaggle. Available online: https://www.kaggle.com/datasets/suchintikasarkar/sentiment-analysis-for-mental-health.

[31] Tomaz Kolenik, Günter Schiepek, and Matjaž Gams. *Computational psychotherapy system for mental health prediction and behavior change with a conversational agent.* Neuropsychiatric Disease and Treatment, 20:2465–2498, 2024. https://doi.org/10.2147/NDT.S469123

[32] Tomaz Kolenik. *Intelligent cognitive system for computational psychotherapy with a conversational agent for attitude and behavior change in stress, anxiety, and depression.* Informatica**,** 49(2), 2025. https://doi.org/10.15388/25-INFOR529

[33] Tomaz Kolenik. *Methods in digital mental health: Smartphone-based assessment and intervention for stress, anxiety, and depression.*In Integrating Artificial Intelligence and IoT for Advanced Health Informatics, Springer, pp. 105–128, 2022. https://doi.org/10.1007/978-3-030-91530-6_6

[34] Tomaz Kolenik and Matjaž Gams. *Intelligent cognitive assistants for attitude and behavior change support in mental health: State-of-the-art technical review.* Electronics, 10(11):1250, 2021. https://doi.org/10.3390/electronics10111250

[35] Tomaz Kolenik and Matjaž Gams. *Persuasive technology for mental health: One step closer to (mental health care) equality?* IEEE Technology and Society Magazine, 40(1):80–86, 2021. https://doi.org/10.1109/MTS.2021.3057079.

Design of a High-Precision ADC Test Platform Integrating STC12C5A60S2 Microcontroller and XGBoost-Based Dynamic Performance Estimation

Hongkun Zhang[1*], Lei Guo[1] Shuanghua Liu[2] Cong Yu[1]
Corresponding email: HongkunZhangg@outlook.com
[1]School of Information Engneering, Xinxiang Institute of Engineering, Xinxiang, Henan, 453700, China
[2]School of Mechanical and Electrical, Xinxiang Institute of Engineering, Xinxiang, Henan, 453700, China

Keywords: Microcontroller, Analog-to-Digital Converter, Signal Acquisition, Parameter Estimation

Received: June 10, 2025

In this paper, a high-precision analog-to-digital converter (ADC) test platform integrating both hardware and software is developed to enable accurate evaluation of ADC performance. On the hardware side, the platform employs the STC12C5A60S2 microcontroller in conjunction with the AD9833 signal generator, supported by amplification and filtering circuits, to ensure stable signal acquisition. On the software side, static parameters are extracted through statistical and time-series analysis, while a gradient boosting regression model based on XGBoost is utilized to estimate key dynamic performance metrics such as total harmonic distortion (THD). The proposed system supports both static and dynamic ADC testing and achieves high estimation accuracy with low complexity. Experimental results demonstrate that the platform can estimate THD with errors within ±2 dB for 85% of the test samples, and within ±3 dB for 92%, outperforming traditional FFT-based methods. Moreover, the effective number of bits (ENOB) deviation remains within ±0.5 bits across diverse conditions. This work provides a cost-effective and extensible solution for ADC characterization, suitable for both laboratory testing and industrial applications.

Povzetek: Članek predstavi platformo za testiranje analogno-digitalnih pretvornikov, ki združuje nizkocenovno strojno opremo z modelom XGBoost za ocenjevanje statičnih ter dinamičnih parametrov.

1 Introduction

Information acquisition and processing have become the core of modern information technology, social development, and scientific and technological progress plays an important role. Sensors as the information acquisition and processing system in the front-end components, directly facing the object being measured, become a world of today's attention to the rapid development of high-tech [1], but also an important symbol of the rapid development of contemporary science and technology [2]. It is the realization of modern measurement and automatic control (including remote sensing, telemetry, and remote control) of the main links, that are the source of the modern information industry, but also the information society depends on the existence and development of the material and technological basis [4]. Now sensing technology and information technology, computer technology, side by side to support the whole modern information industry of the three pillars. In particular, the development of microelectronics technology, microcomputer technology, and information processing technology, made the sensor small in size, low cost, multi-function, and integrated direction [5]. At present, from universe exploration, marine development, environmental protection, and disaster forecasting to the life sciences, including every modern science and technology research as well as people's daily lives [6], etc., almost none of them are closely linked with sensors and sensor technology. Therefore, it is no exaggeration to say: that no sensor and its technology will not have the rapid development of modern science and technology.

A/D converter to convert analog into digital quantities, microcontroller to carry out data processing [7]. With the rapid development of ultra-large-scale integrated circuit technology, a large number of different structures, and performance varies A/D converter chips should come into being [8]-[9]. For the designer, only a reasonable choice of chips can be. Now part of the microcontroller chip is also integrated into the A/D converter, the number of bits for the 8-bit, 10-bit, or 12-bit, and the conversion speed is also very fast, but in the case of the chip can’t meet the needs of the A/D converter, or the need to expand. Therefore, the basic method of external expansion of the A/D converter should be mastered four. This paper uses the AD9833 as an AWD converter chip for A/D conversion simulation experiments.

Analog-to-Digital Converter (ADC) test is divided into static test and dynamic test [10]-[11]. Static test research earlier now has a more perfect test method, but the static test can’t reflect the dynamic characteristics of

the ADC [12], so the general choice of ADC dynamic test method, can reflect the performance of the ADC more perfectly.

Through the static parameter testing process estimate the noise power, and then find out the Effective number of bits (ENOB) and Signal-to-noise and distortion ratio (SND) of the converter. In 2006, Hsill-Wen Ting et al. carried out a study on the correspondence between the nonlinear characteristics and harmonics and established the integral nonlinear characteristics and the harmonic relationship. In 2006, Hsill-Wen Ting et al. investigated the correspondence between the nonlinearity characteristics and harmonics and established a mathematical expression between the integral nonlinearity (INL) and the noise power, which can obtain the effective number of bits and the integral nonlinearity by calculating based on the histogram of the code density obtained under the excitation of a sinusoidal signal. 2011 Duan Jingbo et al. investigated the correlation between input noise and Differential nonlinearity (DND) of the measured device based on the statistical analysis. Differential nonlinearity (DNL), and based on this, the noise power is estimated by analyzing the noise contained in the DNL, and then the SND and ENOB are derived [13].

Harmonic distortion parameters are estimated using a static parametric testing procedure. The integral nonlinearity can describe the transfer characteristic curve of an analog-to-digital converter [14], while under the excitation of a larger amplitude sinusoidal signal, the nonlinearities inside the converter act periodically on the output as shown in Fig. 1, and appear as harmonic components in the form of harmonic components in the spectrum of the output signal. Based on this principle, Jingbo Duan et al. proposed in 2012 to sample the INL curve with the Sine signal, which is equivalent to the output waveform of the converter in the dynamic parameter test, and the harmonic correlation relationship has been obtained by spectral analysis of this waveform and the corresponding dispersion-free dynamic range and total harmonic distortion parameter indexes [15]. In applications that require low-frequency measurement of spectral performance, this method can achieve the same measurement accuracy as traditional test methods [16]. Table 1 presents the similarities and differences among these research works.

Table 1: Related works summary

Reference	Test Type	Architecture	Performance Metrics Reported	Limitations
Duan & Chen (2011) [13]	Static + Dynamic	Code histogram-based + FFT	ENOB, SNR, DNL, INL	Low dynamic estimation accuracy
Janik & Fresnaud (2007) [14]	Dynamic	Spectral method	INL, THD	Lacks flexibility for real-time test
Duan et al. (2012) [15]	Dynamic	INL-sampling with sine excitation	THD, SFDR, ENOB	Complex hardware dependency
Wei et al. (2023) [11]	Static	LabVIEW + Integrated ADC	DNL, INL, Offset	No support for machine learning
Proposed (This work)	Static + Dynamic	STC12C5A60S2 + FFT + XGBoost	THD (±2 dB, 85%), ENOB (±0.5 bits)	Improves accuracy with low-cost microcontroller, interpretable ML model

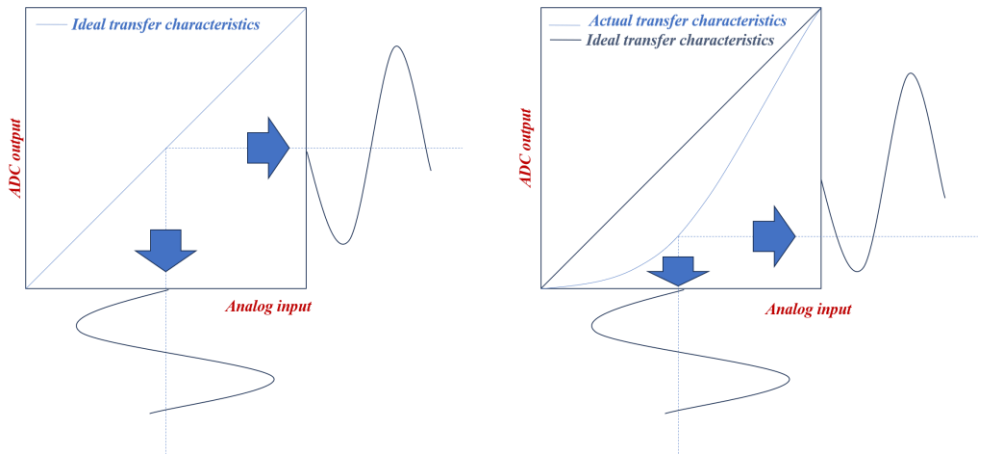


Figure 1: Schematic representation of the effect of nonlinearity on ADC functionality

The function of an analog-to-digital converter is to convert an analog signal, which is continuous in both time and amplitude dimensions, into a digital signal with discrete amplitude and time [17]. The static and dynamic parameters are the two core performance parameters of the converter [18], in which the static parameters are mainly used to describe the transfer characteristic curve of the ADC, including Integral

nonlinearity (INL), Gain error (GE), Differential nonlinearity (DNL) and Offset error (OE), the dynamic parameters are related to the spectral performance and measure the extent to which the ADC can reproduce a sinusoidal signal as a digital signal under its excitation. They mainly include Signal-to-noise-and-distortion ratio (SND), Total harmonic distortion (THD), Effective number of bits (ENOB), Spurious-free dynamic range (SFDR), and Signal-to-noise ratio (SNR).

ADC is one of the core devices in modern electronic systems, responsible for converting analog signals into digital signals for subsequent processing and analysis [19]-[20]. Accurate evaluation of ADC performance is critical in applications requiring high precision, such as medical imaging, precision instruments, and communications [21]. However, current ADC test platforms have trade-offs between cost, accuracy, and scalability, and it is difficult to comprehensively evaluate their static and dynamic performance characteristics.

To address these issues, this paper proposes a novel ADC test platform that combines the co-design of hardware and software and is capable of meeting the test requirements of high accuracy and flexibility. The hardware part of the platform adopts the STC12C5A60S2 microcontroller and AD9833 chip to support data acquisition and processing; the software part realizes signal analysis through the FFT algorithm and accurately estimates dynamic performance through the dynamic parameter gradient boost regression model. The research in this paper aims to provide an efficient and reliable solution for ADC testing [22]-[23].

This paper combines the feature extraction methods of nonlinear curves with principal component analysis and time series research, compares and determines the optimal feature combinations, and the optimal hyperparameters of each model, and analyzes the significant differences between various estimation models based on an understanding of the algorithmic principles and the characteristics of dynamic parameter estimation problems, in order to determine the optimal model. Furthermore, it quantitatively evaluates the contribution degree of specific feature variables within the model interpretation framework, thereby improving model transparency. Finally, the modeling and correlation analyses in this study were performed on the Jupyter Notebook platform using open-source libraries in Python 3.7, including NumPy120,3, Pandas1.3.4, Scipy I.7.1, Matplotlib3.4.3, TensorFlow2,80, sHAP041.0, xgboost1.7.0, and others. This research aims to answer: (1) Can machine learning improve dynamic performance estimation in ADC testing? (2) What is the accuracy improvement of XGBoost-based THD prediction compared to traditional FFT analysis? Success is defined by reducing THD prediction error below ±2 dB for >85% of test samples.

2 Hardware designs

2.1 STC12C5A60S2 microcontroller with AD9833 chip

This design uses the STC12C5A60S2 microcontroller as the control core to control and configure the AD9833. Through the key can directly enter the signal frequency and amplitude, as well as waveform type switching, very easy to use. AD9833 output voltage is only 3V, and the actual use of the waveform voltage is often larger, so it also needs to be amplified. The design of the amplifier circuit AD8397 amplifies the voltage to reach between 0.1V and 5V adjustable so that the signal gets a larger power. At the same time, the LCD shows the parameters of the indicators. The system hardware block diagram is shown in Figure 2.

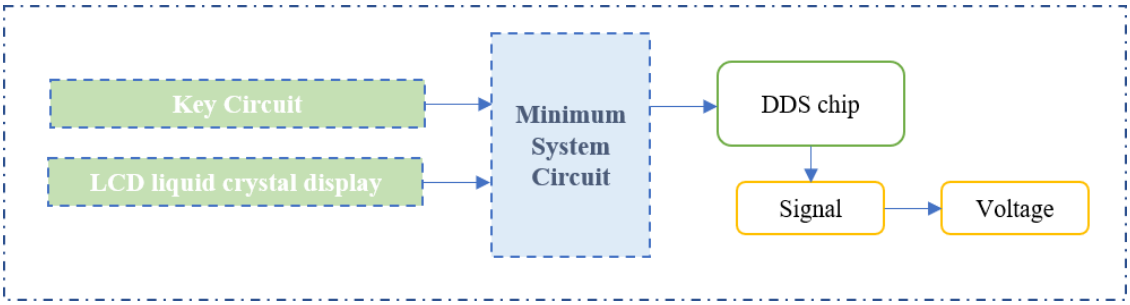


Figure 2: System hardware block diagram

STC12C5A60S2 is a new generation of 8051 microcontrollers, containing a central processing unit (CPU), program memory (Flash), data memory (SRAW), timer/counter, UART serial port, high-speed A/D conversion, SPI interface, other modules. Support for serial program burning and the development of microcontroller equipment requirements are relatively low, the time is relatively shortened and cost-effective. The microcontroller minimum system clock circuit using a crystal oscillator 11.0593Hz, capacitance selection of 20pf capacitors. Because the function signal generator needs to continuously output waveforms and does not need to often reset this minimum system circuit does not specifically draw a reset circuit.

AD9833 is a high-precision, low power consumption, simple structure and small size and has a serial interface function of the DDS chip, produced by ADI, its output signal control, such as frequency and

phase can be programmed to control the realization of the AD9833 chip power supply voltage range of 2.3V ~ 5.5V, the output waveform through the VOUT pin.SDATA input, FSYNC When SDATA is input, FSYNC must be low, and the data is sent to the shift register of AD9833 after 16 falling edges of SCLK to load the data bits. The waveform output from the AD9833 chip depends on the control bits OPBITEN(D5) and Mode(D1) in the control register of the controller. If the content in the AD9833 control register needs to be changed, the D15 and D14 positions go low. When the phase register is written D15 and D14 are set high and D13 specifies which phase register is written.

2.2 Amplification circuit design

To meet the required output signal amplitude of up to 5 V, the analog output of the AD9833 signal generator, which is limited to approximately 0.6–0.7 V peak-to-peak, is amplified using a high-output-current operational amplifier, the AD8397. A standard non-inverting amplifier configuration is adopted to preserve signal polarity and minimize phase distortion.

The gain of a non-inverting amplifier is defined by the following transfer function:

$$V_{out} = (1 + \frac{R_f}{R_g})V_{in}$$

where Rf is the feedback resistor and Rg is the resistor connected between the inverting input and ground. In our design, both Rf and Rg are set to 10 kΩ, resulting in a voltage gain of:

$$V_{out} = (1 + \frac{10k\Omega}{10k\Omega})V_{in} = 2V_{in}$$

This amplification factor effectively boosts the signal amplitude to a usable level for ADC testing. The AD8397 is selected due to its high current drive capability (up to 310 mA) and wide bandwidth, ensuring minimal distortion and accurate amplification even at higher frequencies.

The previously ambiguous statement "inverted input through the resistor R4 ground" has been clarified in this context as the resistor Rg connected between the inverting input of the op-amp and ground. The non-inverting input receives the filtered output of the AD9833 signal generator, ensuring signal fidelity throughout the amplification stage.

2.3 Filtering circuit design

To improve signal integrity and suppress high-frequency noise components that may affect the accuracy of ADC conversion, a basic first-order low-pass filter is added after the amplification stage. This filter consists of a series resistor and a parallel capacitor to ground, forming a simple RC structure with a cutoff frequency selected based on the maximum test signal frequency. In our implementation, the resistor and capacitor values are chosen to yield a cutoff frequency of approximately 1 kHz, which effectively attenuates high-frequency interference while preserving the essential signal content required for ADC evaluation. The addition of this filtering stage ensures a cleaner input waveform to the ADC under test and enhances the reliability of static and dynamic parameter extraction in subsequent analysis stages.

3 Hardware architecture system design for ADC test bench

3.1 High-precision data processing system

The test daughter board is designed according to different models of ADC chips to meet their respective test requirements [24], in the test, only need to replace the top layer of the ADC test daughter board can complete the test, which makes the data converter test platform with communication. The structure of the ADC test daughter board block diagram is shown in Figure 3, the circuitry includes the ADC chip, signal inputs, clock inputs, Barron circuit high-speed data interface, etc. Among them, the high-speed data interface is the interface to communicate with the FPGA data processing system. One of the high-speed data interfaces for the test daughter board and FPGA data processing system communication interface, through the interface to complete the transmission of sampling data: when the ADC chip input signal for differential signals, the need for a balun circuit will be converted to differential signals to the ADC chip input pins.

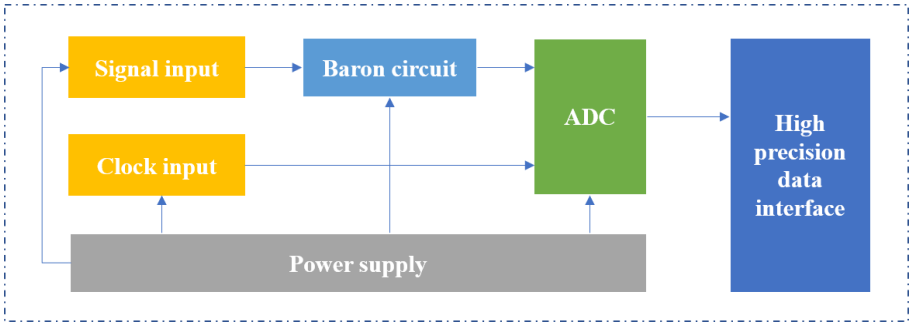


Figure 3: ADC test daughterboard block diagram

The high-speed data processing system of the test platform can directly receive the output signal of the ADC [25], so in the hardware design of the test daughter board, it is only necessary to ensure that the ADC chip can work normally and stably output signals, and the output signals can be directly transmitted to the high-speed data processing system through the high-speed data interface.

High-speed data processing system to complete the control of the top ADC chip and ADC output data acquisition, but also to complete the data processing and transmission, high-speed data processing system functional block diagram shown in Figure 4, by the FPGA chip, crystal circuit, high-spee

d data interface, JTAG (Joint Test Action Group) circuit, Gigabit network interface circuit, and other components.

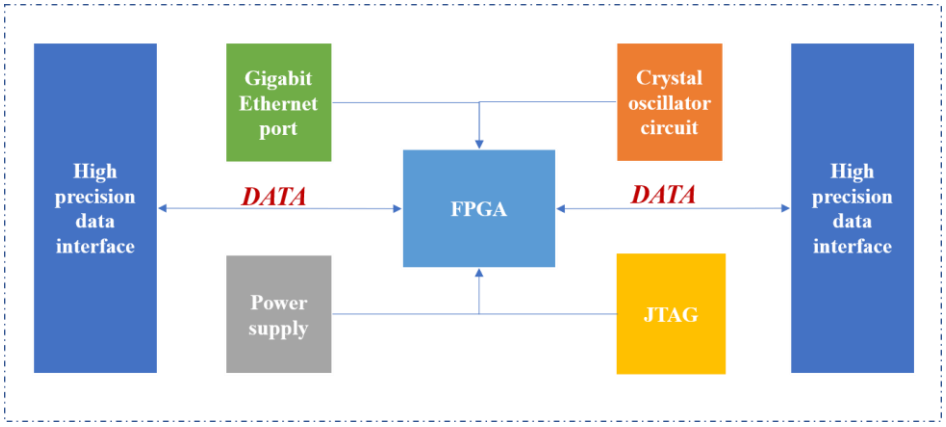


Figure 4: Functional block diagram of high-precision data processing module

The core chip of the high-speed data processing system uses Xilinx Virtex6 series model XC6VLX240T FPGA, which has 720 input/output pins, 416 36kb block memories, and can complete the acquisition of ADC output signals, the acquisition of data processing and transmission functions. The JTAG circuit in the high-speed data processing system provides the function of downloading and debugging for the FPGA chip, which can download the program into the FPGA through ISE software and complete the online debugging by using the Chipscope function to check the status of the selected signal when the FPGA is running.

The control system Is the control center of the whole test platform, Ih accomplishes the following tasks:

(1) establish communication with the high-speed data processing system, complete the programming function of the high-speed data processing system, monitor and control the status of the high-speed data processing system, as well as

state monitoring and control functions, as well as to complete the transmission of ADC sampling data.

(2) Establishing communication with the host computer, completing the control of the whole test platform by receiving commands from the host computer, and uploading the sampling data to the host computer.

(3) Establish communication with the chip test daughter board, complete the read/write operation of the chip registers in the test daughter board module, and then complete the chip initialization and other related operations. The ARM control system establishes a communication connection with the upper computer through the network port and establishes a communication connection with the upper high-speed data processing system through the high-speed data interface, which completes the control of the whole test platform, including the configuration of the ADC chip, the status control of the high-speed data processing system, the control of the ADC chip, the status control of the ADC chip, and the control of the ADC chip status. Data processing system state control, ADC sampling data transmission, etc. The ARM control system is developed around Samsung's S3C6410 processor, which can run the Linux system, has a very complete bus architecture, and operates at a fast speed and high efficiency.

3.2 ADC test bench solution

The proposed test platform is developed based on a prior ARM-FPGA implementation. In the original system, an ARM processor served as the control module, coordinating ADC data acquisition and system configuration. However, due to the high operating frequency of the ARM core, electromagnetic interference was observed, which affected the accuracy of high-precision ADC measurements. To mitigate this, the ARM control module was replaced with an MSP430 microcontroller in the updated platform design. ARM-FPGA data converter test platform has been well realized ADC chip test, but due to the high frequency of ARM operation, the test process will produce a small part of the interference signal, these interference signals are extremely small, but in the high-precision ADC test will be crosstalk to the ADC output signals, for the test results of the impact of some. Therefore, to ensure the accuracy of the test results of high-precision ADC chips, the elimination of interference has become the primary solution to the problem.

Low-power microcontrollers due to their low operating frequency, have the characteristics of small power consumption, the external electromagnetic radiation is much smaller than ARM, and can solve the interference problems brought by the ARM platform. At the same time, it has a powerful expansion function and can meet the basic needs of the test platform control system, so it was selected as the implementation program of this paper. The new MSP430-FPGA data converter test platform is improved and completed based on the original ARM-FPGA data converter test platform design of the three-layer stacked circuit board design, so that the original design of the test daughter board module and high-speed data processing system is completely retained, and only need to change the design of the MSP430 control system and the upper computer control software interface, which reduces the workload.

Therefore, the MSP430-FPGA data converter test platform retains the original test platform architecture, replaces the ARM control system with the MSP430 control system, re-designs the hardware board-level circuitry to realize the basic needs of the control system, develops software around the low-power microcontroller module, and develops a Bluetooth wireless communication module based on wired communication with the upper computer to extend the communication between the test platform and the host computer. The Bluetooth communication module can be embedded into the msp430 control system to work, establish communication with the control system through the SPI communication interface, establish communication with the host computer based on the Bluetooth wireless communication protocol, and be able to provide Bluetooth wireless communication connection when the wired communication method is not connected. At the same time, the Bluetooth wireless communication connection reduces the common ground interference between the host computer and the test platform, further reducing the impact on the test results.

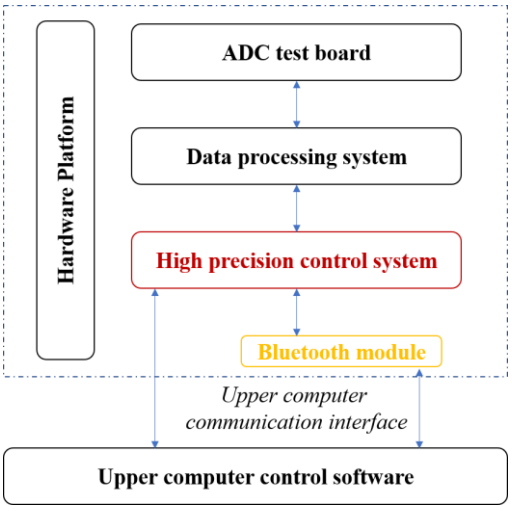


Figure 5: Data converter testbed architecture

The architecture of the data converter test platform Is shown In Figure 5, where the MSP430 control system replaces the original ARM control system, establishes a communication connection with the hardware platform's upper-level high-speed data processing system through the high-speed data interface, and establishes a communication connection with the host computer through the host computer communication interface. The host computer control software sends control commands to the control system through the host computer communication interface, and the control system executes the commands and then controls the data processing system and the ADC test daughter board through the high-speed data interface so that they are in the corresponding test state. When the ADC chip completes the data conversion, the collected data are processed by the data processing system and transmitted to the control system through the high-speed data interface, and then the control system uploads them to the control software through the communication interface of the host computer, and the control software finally processes the data and calculates the performance parameters to get the test results.

To clarify the relationship between the signal generation and ADC testing components, this work adopts a modular system design in which the signal generator module—comprising the STC12C5A60S2 microcontroller, the AD9833 direct digital synthesis (DDS) chip, and an AD8397-based amplification circuit—acts as the excitation source for the ADC under test. The analog signal produced by this module is transmitted directly to the input of the ADC device, such as the ADC0804, which is mounted on a replaceable test daughterboard. This daughterboard interfaces with the high-speed FPGA-based data processing unit, which captures the ADC output and forwards the sampled data for subsequent analysis.

By separating the signal generation from the data acquisition and processing components, the system minimizes electromagnetic interference and facilitates better control over input conditions. The overall platform, coordinated by the control module and host software, supports full-loop operation for both static and dynamic testing of ADC performance. This integration ensures that the signal chain from excitation to parameter estimation remains coherent, accurate, and adaptable to different ADC configurations.

In this design, the networked communication interface between the host computer and the original platform is also changed. The changed communication interface of the host computer consists of two ways, one is a wired connection based on a USB interface and the other is a wireless connection based on Bluetooth communication module. The Bluetooth communication module establishes a communication connection with the control system through the Serial Peripheral Interface (SPI) communication interface, and the wireless connection can solve the common electrical interference that may be introduced when the host computer and the test platform are directly electrically connected, and further improve the performance of the test system.

4 Parameter estimation algorithm design and implementation

4.1 Static performance feature extraction method design

The gain error offset error, and DNL and NL curve data provide a complete picture of the static characteristics of the system. These parameters are obtained by further calculations based on the density histogram [26]. The nonlinear curves can be considered as sampling information in a particular distribution, and this particular distribution shows different dynamics with different dynamic parameters. Therefore, the features describing this distribution are extracted from the nonlinear curve, which can effectively characterize the ADC dynamic parameters.

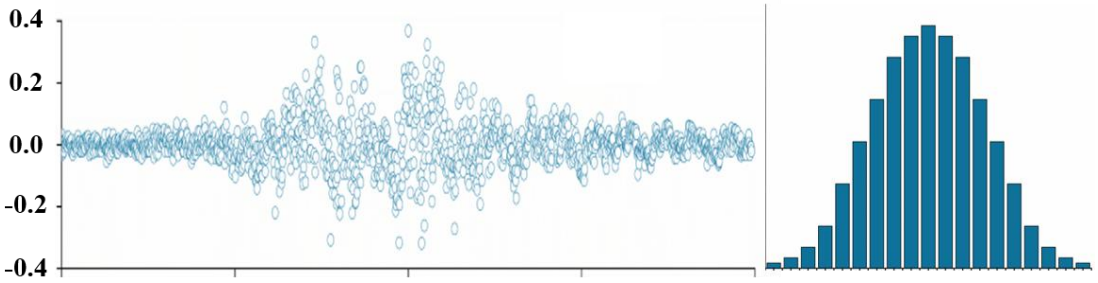


Figure 6. DNL curves and their statistical histograms

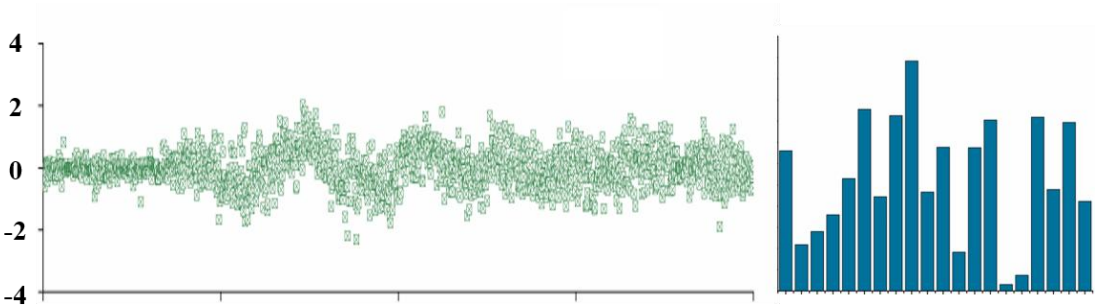


Figure 7: INL curves and their statistical histograms

Figures 6 and 7 show a set of measured integral nonlinearity (INL) and differential nonlinearity (DNL) curves of the ADC0804, along with their corresponding statistical histograms. From Fig. 6, it can be seen that the DNL curves can be approximated as a section of white noise, i.e., the sequence is completely random without some kind of periodicity or trend, and the correlation of the integral nonlinear errors between different points is low, making it difficult to predict another point based on one point. This is in general agreement with the conclusion of Jinbo Duan et al. in the literature [15] that the differential nonlinear curve can be regarded as a smooth sequence. As can be seen from Fig. 7, the NL curve has the following characteristics: (1) the data in the series have some kind of trend, and the correlation between the two points that are closer together is stronger: (2) the values of the points with larger intervals in the series have a certain degree of randomness. A time series is a string of data lined up in chronological order. Analyzing the characteristics of time-series data by studying the dependencies between the data within the time series is known as time-series analysis in academic circles. Time series modeling can be used to observe trends in monitoring data over long periods. Combining the characteristics of the NL and DNL curves mentioned above, it can be found that the static nonlinear curve of ADC is very similar to the non-stationary time series and the stationary time series in time series analysis. Therefore, time series are used here for feature extraction of nonlinear characteristic curves of analog-to-digital converters.

The most commonly used method for feature extraction of smooth time series Is statistical analysis. Therefore, in this work, the correlation index in statistical analysis is adopted as the feature extraction method for the integral nonlinear and differential nonlinear curves to fully portray the static nonlinear characteristics while reducing the feature space. In this paper, we first consider the common statistical features related to nonlinear curves: mean value, root mean square value, standard deviation, and other eight statistical indicators. These indicators portray the distribution of nonlinearity from different perspectives.

Skewness, g1 describes the direction and degree of the distribution shift. If g1>0, the distribution is right-skewed and vice versa. A larger |g| means that the more skewed the distribution is. Kurtosis, g2 is used to characterize the steepness of a statistical distribution as shown in Figure 8, the larger g2, the steeper the distribution. The kurtosis of a normal distribution is equal to 3. The curve of the distribution is “flatter” when

the kurtosis is less than 3, and "steeper" when it is greater than 3.

According to the above analysis, there is some information redundancy in the integral nonlinear curve. Principal Component Analysis can analyze and reduce the dimensionality of nonlinear curves, and its core idea is to find a direction so that the original data are scattered as much as possible in the projection direction, and then map the original data to this direction by orthogonal transformation. At this point, this direction contains the most information about the sample data set. Finally, by retaining the large variance dimension and ignoring the small variance dimension, PCA can reduce the dimensionality of the data while retaining as much information as possible about the differences between the samples, and the feature dimensions obtained in this way are orthogonal (uncorrelated). Moreover, it has been proved that PCA transformation is the best orthogonal transformation.

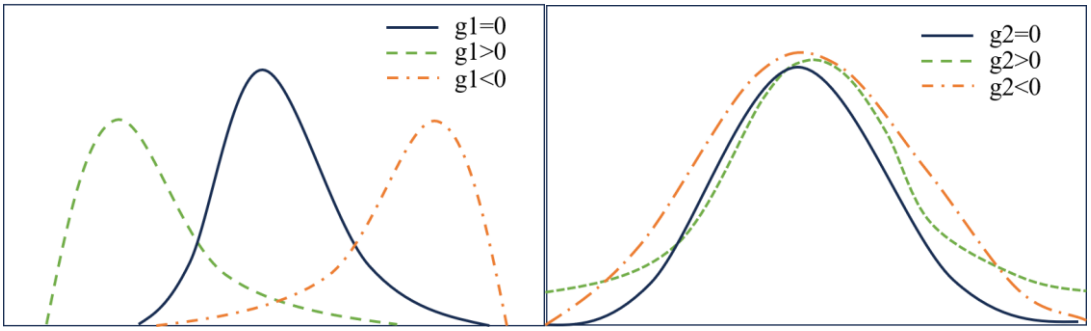


Figure 8: Skewness and kurtosis schematic

Figure 9 illustrates the first 16 principal components of the integral nonlinear curve and their percentage of explainable variance, where the bar represents the percentage of explainable variance of the sample along the direction of that principal component, and the curve is a cumulative sum of the bars representing the percentage of explainable variance of the data over the space tensored by the first n principal component directions. As can be seen in Figure 10, the interpretable variance of the data reaches 84.7% in the 1st principal component direction alone, and the sum of the variance in the first three principal component directions reaches 89.5%. In addition, the percentage of the explainable variance of the principal components at the back of the ranking begins to gradually converge to zero. This means that not much information is lost by discarding the later principal components. To limit the complexity of the model, the work in this chapter selects the first 48 principal components as the feature input variables of the THD estimation algorithm based on machine learning, and the cumulative explainable variance reaches 97%.

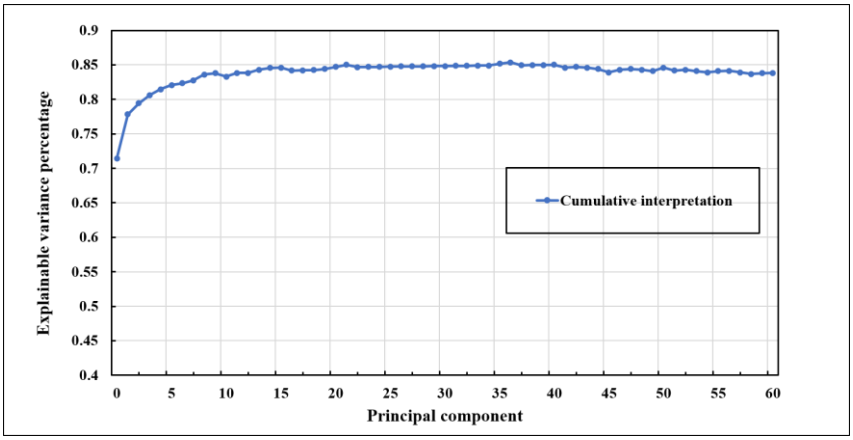


Figure 9: Percentage of variance explained by principal components

4.2 Dynamic parameter gradient boosting tree regression model

In this study, the problem to be solved is to estimate the THD parameters of the same but with different performance parameters ADCs with the input features of each machine learning model as X= [GE, OE, INL-PCA-i, INL-statistical features, DNL-statistical features], where INL-PCA-i stands for the ith principal component of the NL with i=O,1,2,47. This belongs to the regression learning problem.

Boosting is a machine learning technique that integrates multiple weak learners for regression and classification problems. If during training, each iteration of the model generates a weak learner based on the gradient direction of the loss function to complement the

original model, fit the residuals between the true values and the output of the original model, and ultimately enhance the learning capability by integrating multiple weak learners, then the whole model can be called a Gradient boosting machine (GBM).

The XGBoost (eXtreme Gradient Boosting) model was first proposed in 20l6 by Chen and Carlos Guestrin [27] first proposed a model for implementing gradient-boosting regression trees. At each iteration of XGBoost, trees are added to the existing model, and the scores of multiple trees are eventually summed up as the final model inputs.XGBoost generalizes the loss function by a 2nd-order Taylor series expansion, so that the specific form of the loss function can be disregarded, and decoupling model optimization from loss function selection is realized This increases the generalizability of the model. Compared to other classical machine learning algorithms, XGBoost shows higher prediction accuracy and faster processing time in many domains, as well as lower computational cost and complexity. These properties apply to dynamic parameter estimation problems.

To get the best performance from XGBoost, this work initializes all the parameters of the model concerning Table 1 and uses GridSearchCV to perform an exhaustive search of the hyperparameters with specified parameter values. Table 2 shows the optimal hyperparameters searched by GridSearchCV, and the effect of each hyperparameter on the output of the model is analyzed below. This integration of machine learning into embedded systems for real-time parameter estimation follows similar strategies reported in recent research [28].

Table 2: Hyperparameter optimization space and optimal values for XGBoost

Model parameters	Starting value	Search space	Grid Search Optimization
n_estimators	100	[100,150,200,250,300]	200
γ	0.47	[0,0.2,0.5,0.8,1.0]	0
max depth	4	[2,3,4,5]	5
learning_rate	0.16	[0.03,0.05,0.08,0.1]	0.06
colsample_bytree	0.82	[0.6,0.7,0.8,0.9]	0.8
min child weight	1	[1,2,3,4,5,6]	1

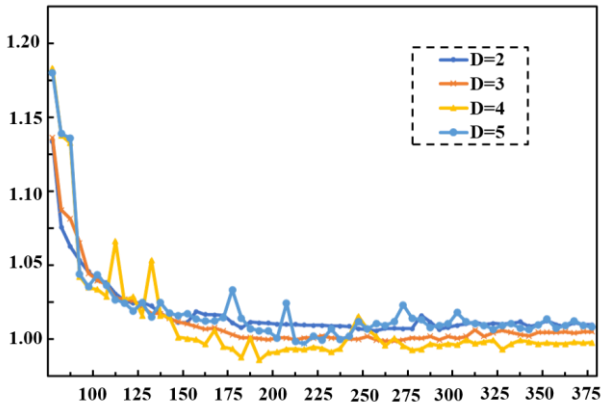


Figure 10: Estimation performance comparison of XGBoost with different max_depth and n_estimator parameters

Figure 11 demonstrates the extent to which the estimation performance of XGBoost on the test samples is affected by changes in the n_estimator and max_depth parameters. It can be seen that the average absolute errors of the XGBoost models with different max_depth settings are relatively large when the n_estimator is small, and after 200 iterations, the curves all start to

Gradually converge, in addition, it can be seen that at n estima1or equal to 200, the model estimation performance is not significantly improved with the increase of the number of layers, therefore, to limit the complexity of the model and prevent overfitting, the optimal values of n_estimator and max_depth are set to 200 and 3 layers, respectively.

4.3 Analysis of estimated results

Compared to traditional methods, the XGBoost-based INL estimation model is more accurate, but less interpretable and almost a black-box model. Shapley Additive explanations (SHAP), based on game theory and able to quantify the marginal contribution of each feature to the model output value, is a widely applicable framework for model interpretation [29]. Therefore, this work employs SHAP to quantify the influences on INL in the model and enhance the interpretability of the model. The use of PCA and statistical descriptors as input features is consistent with recent unsupervised learning and feature reduction methodologies in embedded signal analysis [30].

According to the feature importance ranking given by the SHAP model interpreter, the features with the lowest importance (including variables such as INL-kurt, DNL-std, DNL-mean, etc.) were eliminated one by one to reduce the complexity of the model. In addition, this work discusses the principal component vectors included in the model, as shown in Figure 11. When the model incorporates 48 principal component features, the model has the lowest absolute error and the best estimation performance on the test set.

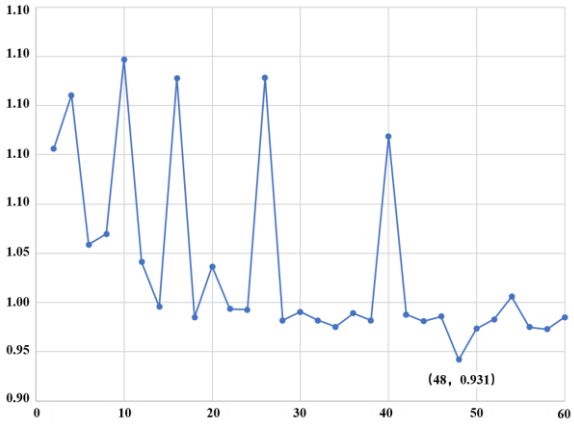


Figure: 11 Comparison of estimation performance of XGBoost model when taking the first n principal components

Figure 12 shows the estimated and measured values of XGBoost in the test set (100 samples are randomly selected here to demonstrate the effect), where the dashed line indicates the model estimated values, the solid line indicates the measured values and the horizontal axis is the sample number. It can be seen that the trend of the measured values and the model-estimated values are generally consistent. Tabel 3 demonstrates the quantitative analysis of XGBoost's estimation results for all test samples (512 sets), and it can be seen that there are only 31 sets of samples with estimation errors greater than 3 Db, accounting for 7% of the test set weight, i.e., 92% of the data have errors within the range of 3 Db.

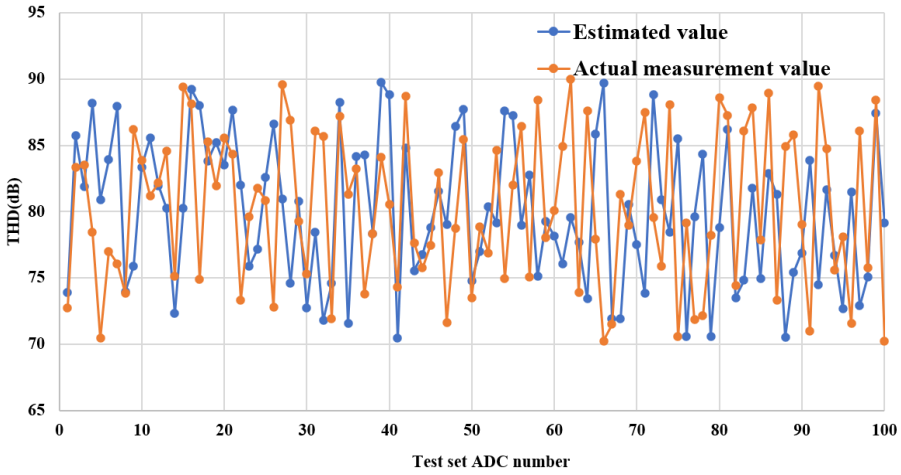


Figure 12: THD estimate vs. reference value

Table 3: Analysis of XGBoost estimation error in the test set

Error (Db)	Points	Proportion in the test set
±2	436	85%
±3	477	92%
±4	502	97%

Figure 13 shows the top 10 features with the highest contribution (SHAP value) to the estimation results, with the features ranked from top to bottom according to their contribution. It can be seen that the gain error, with a SHAP value equal to about 0.008, has the highest contribution to the total harmonic misalignment parameter, significantly higher than the second-ranked INL-RMS (with a SHAP value equal to about 0.004). From the second NL-RMS to PCA-12 (SHAP value | approximately equal to 0.003), the contributions of several characteristic variables show a steady decreasing trend. In addition, it is easy to find that nine of the top ten feature variables with the highest SHAP values originate from the integral nonlinear curve, which implies that the INL curve is an important feature for estimating the THD parameter, a conclusion that is consistent with the findings of J.B. Duan et al. in the literature the results of the study are consistent.

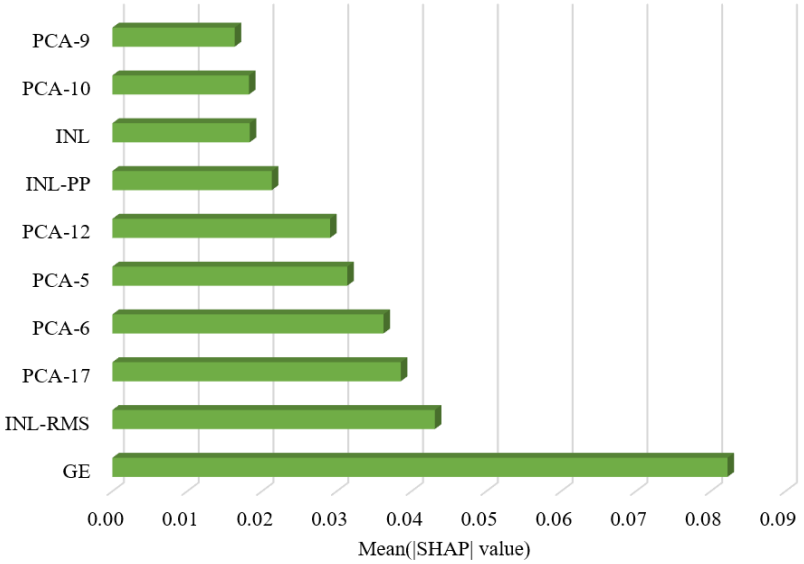


Figure 13: Mean (|SHAP values|) for the 16 features that have the greatest impact on the THD estimates.

5 System software design

5.1 FFT algorithm principle

N*logN times operation, significantly improving the arithmetic process reduces the amount of arithmetic[32]. Fast Fourier transform according to the discrete Fourier transform of the parity of the real and other characteristics, the use of symmetry, constantly divided into two methods, close to the limit, that is, divided into two and two groups of DFT operation unit, the algorithm complexity is optimized for the O (N * logN) level, as a way to greatly reduce the computer's running time.

The bisection algorithm is as follows:

$$X_k = \sum_{n=0}^{N-1} x_n e^{\frac{-i2\pi kn}{N}} = \sum_{m=0}^{N-1} x_{2m} e^{\frac{-i2\pi k(2m)}{N}} + \sum_{m=0}^{N-1} x_{2m+1} e^{\frac{-i2\pi k(2m+1)}{N}} = \sum_{m=0}^{N-1} x_{2m} e^{\frac{-i2\pi k(2m)}{N/2}} + \sum_{m=0}^{N-1} x_{2m+1} e^{\frac{-i2\pi km}{N/2}}$$

5.2 FFT program implementation

As the FFT microprocessor uses machine language, machine language through the complementary code to complete the operation [31]-[32], the output data for the complementary code, and the need for the original code through the complementary code of the relationship between the program to get the desired results. When the output is positive, the complementary code and the original code are equal, the output is negative, the highest bit of the complementary code is unchanged, and the rest of the inverse of each of the original code, through the highest bit of data to determine the number of positive and negative, the highest bit of 1 that is, for the negative, for the 0 that is, for the positive number. In the finalized platform, the FFT algorithm is executed on the host PC using optimized scientific computing libraries such as NumPy, based on raw data collected by the FPGA. Earlier prototypes implemented FFT logic at the microcontroller level for validation purposes, but these were superseded by software-based methods offering higher resolution and performance. Through the above operations can be obtained after the true value of the data. The program block diagram is shown in Figure 14.

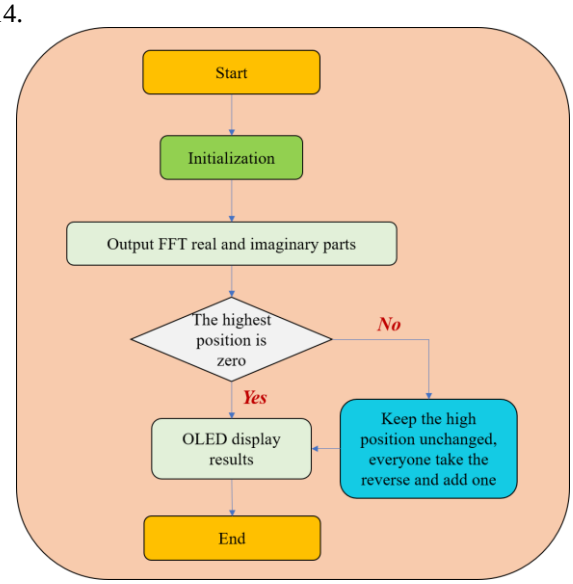


Figure 14: Program block diagram

Before building the system, first debug individual modules to work properly, after all the modules can meet the requirements, start building the system, the power amplifier circuit, non-contact current sensor, and microcontroller detection circuit are fixed in the PCB board, and will be connected to fix the wires, connected to the switching power supply and the signal generator, the system indicators are tested in turn.

The following table shows the test data when the input signal is sinusoidal, and the data are the average value obtained from five measurements.

Table 4: Sine wave test data sheet

Input of signal A		Output of signal B		Amplitude error	Frequency error
A/V	A/Hz	Output peak to peak value /V	Output frequency /Hz		
1.50	50	1.44	50.51	4.75%	0.020%
2.50	100	2.42	101.02	4.04%	0.020%
5.00	200	4.87	202.01	3.64%	0.005%
6.50	500	6.41	505.04	2.32%	0.008%
8.00	700	7.99	707.08	1.31%	0.014%
9.00	900	9.16	908.99	0.79%	0.001%

The results presented in Table 4, which detail the amplitude and frequency errors observed during sinusoidal signal testing, offer an indirect yet important reflection of the system's overall ADC testing precision. The amplitude error affects the linearity of the ADC's transfer function, which directly correlates with the effective number of bits (ENOB) and total harmonic distortion (THD). For example, under the input signal of 5.00 V and 200 Hz, the amplitude error was maintained within 3.64%, while the frequency error remained as low as 0.005%. Such low deviations in both amplitude and frequency ensure that the excitation signal maintains spectral purity, which is crucial for accurate dynamic performance estimation.

These stable signal conditions form a reliable basis for FFT analysis and the XGBoost regression model, enabling precise estimation of THD and signal-to-noise ratio (SNR). The observed consistency between the hardware-generated signal and the measured output confirms the system's capability to preserve signal integrity. This, in turn, validates the reliability of subsequent dynamic performance metrics, as reflected in the experimental results where over 85% of test samples achieved THD estimation errors within ±2 dB and ENOB deviation within ±0.5 bits. Therefore, the low signal distortion reported in Table 4 is not only a reflection of hardware precision but also a prerequisite condition that contributes to the high accuracy of the overall ADC evaluation process.

6 Discussion

The proposed ADC test platform demonstrates considerable advantages over traditional evaluation methods in terms of performance, flexibility, and implementation cost. As summarized in Table 1, conventional approaches for ADC testing—such as histogram-based static tests and FFT-based dynamic analyses—often require specialized hardware configurations and are limited in their ability to adapt to varying signal conditions. In contrast, our platform integrates low-cost hardware components with a machine learning-based dynamic parameter estimation algorithm, achieving accurate and consistent results across a range of test scenarios. The experimental results indicate that the proposed system can estimate total harmonic distortion (THD) with errors within ±2 dB for 85% of the test samples and within ±3 dB for 92%, while maintaining the effective number of bits (ENOB) deviation within ±0.5 bits. These results highlight the system's capability to perform both static and dynamic testing with high precision.

A key contribution of this work is the integration of the XGBoost regression model for dynamic parameter estimation. Compared to traditional analytical or FFT-only methods, XGBoost exhibits enhanced robustness and adaptability when mapping complex nonlinear relationships between static ADC characteristics (such as INL and DNL) and dynamic performance metrics. The use of SHAP (SHapley Additive exPlanations) further improves model transparency by quantifying the contribution of each feature to the final output, which supports the interpretability of the prediction process. Notably, principal components derived from INL curves emerged as the most influential features, confirming the theoretical link between static nonlinearity and dynamic spectral performance.

While the system achieves promising results, certain trade-offs must be acknowledged. The inclusion of amplification and filtering circuits adds some complexity to the hardware design, and although the overall latency introduced by the FFT and XGBoost pipeline is relatively small, it may still be a consideration in time-critical applications. Furthermore, despite the improvements in model interpretability via SHAP, machine learning models inherently involve a level of abstraction that can complicate result verification compared to fully analytical methods.

Nonetheless, the developed platform provides a compelling solution for high-precision ADC testing in both academic and industrial contexts. It offers a balance between cost, accuracy, and extensibility, and can be adapted to a range of testing requirements. Future work will focus on extending the platform to support higher-resolution ADCs, multi-channel acquisition, and real-time model acceleration through FPGA-based

deployment, thereby further enhancing its applicability in advanced signal processing environments.

7 Conclusion

This paper presents the design and implementation of a high-precision ADC (analog-to-digital converter) test platform that integrates hardware and software to comprehensively evaluate the static and dynamic performance of ADCs. The main contributions of this work are summarized as follows:

Hardware Design: The hardware platform is built using the STC12C5A60S2 microcontroller and the AD9833 chip, combined with the design of amplification and filtering circuits, to construct a high-precision data acquisition and processing system. Experimental results demonstrate that the platform accurately extracts key ADC performance metrics, validating the effectiveness and reliability of the system design. In the future, the platform can be further optimized to support higher precision testing requirements and expanded application scenarios, providing technical support for ADC development and quality control.

Experimental validation confirms that the signal generator module produces stable excitation signals, while the XGBoost-based dynamic parameter estimation model achieves high accuracy in predicting key ADC performance metrics such as THD and ENOB. This demonstrates the platform's capability for accurate ADC evaluation in both static and dynamic contexts.

Static and Dynamic Parameter Extraction: A manual static performance parameter extraction method is proposed by integrating time series analysis and principal component analysis. Experimental results indicate that the model estimation based on this method significantly outperforms models using traditional static parameters as input features. Furthermore, all feature variables used in the model can be obtained during static parameter testing, ensuring the model's applicability in production testing. Additionally, the XGBoost model is selected for ADC dynamic parameter estimation. The comparative analysis of estimation performance among different models reveals that the XGBoost model achieves superior estimation accuracy. These results demonstrate the potential of the proposed platform and methods to enhance ADC performance evaluation and contribute to the advancement of ADC technology.

References

[1] Massaro, A. (2023). Advanced electronic and optoelectronic sensors, applications, modelling and industry 5.0 perspectives. Applied Sciences, 13(7), 4582. https://doi.org/10.3390/app13074582

[2] Huang, R., Yan, P., & Yang, X. (2021). Knowledge map visualization of technology hotspots and development trends in China's textile manufacturing industry. IET Collaborative Intelligent Manufacturing, 3(3), 243-251. https://doi.org/10.1049/cim2.12024

[3] Li, Q., Yang, Y., & Jiang, P. (2022). Remote monitoring and maintenance for equipment and production lines on industrial internet: A literature review. Machines, 11(1), 12. https://doi.org/10.3390/machines11010012

[4] Wu, G., Chen, Z., & Dang, J. (2024). IoT Sensing Technology. In Intelligent Bridge Maintenance and Management: Emerging Digital Technologies (pp. 67-105). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-97-3827-4_3

[5] Fu, B., Liu, Y., Li, Y., Wang, C., Li, C., Jiang, W., ... & Zhao, W. (2021). The research priorities of resources and environmental sciences. Geography and Sustainability, 2(2), 87-94. https://doi.org/10.1016/j.geosus.2021.04.001

[6] McNutt, M. K. (2022). Civilization-saving science for the twenty-first century. Annual Review of Earth and Planetary Sciences, 50(1), 1-12. https://doi.org/10.1146/annurev-earth-033021-081125

[7] Waho, T. (2022). Introduction to Analog-to-Digital Converters. River Publishers. https://doi.org/10.1201/9781003338680

[8] Wang, J., Ilyas, N., Ren, Y., Ji, Y., Li, S., Li, C., ... & Ang, K. W. (2024). Technology and integration roadmap for optoelectronic memristor. Advanced Materials, 36(9), 2307393. https://doi.org/10.1002/adma.202307393

[9] Licastro, D. (2022). Development and evaluation of Synthesis and Optimization strategies for digital integrated circuits (Doctoral dissertation, Politecnico di Torino). http://webthesis.biblio.polito.it/id/eprint/24515

[10] Vaikkola, V. P. (2024). Integrated Circuit Testing Using Automatic Test Equipment.

[11] Wei, L., Junfeng, C., & Shiguang, S. (2023, October). ADC Static Test System Based on LabVIEW Platform. In 2023 8th International Conference on Integrated Circuits and Microsystems (ICICM) (pp. 387-391). IEEE. https://doi.org/10.1109/ICICM59499.2023.10365768

[12] Sparta, R. (2024). Implementation and testing of modular ADCS software using modular avionics test bench for cubesats (Doctoral dissertation, Politecnico di Torino). http://webthesis.biblio.polito.it/id/eprint/31327

[13] Duan, J., & Chen, D. (2011, May). SNR measurement based on linearity test for ADC BIST. In 2011 IEEE International Symposium of Circuits and Systems (ISCAS) (pp. 269-272). IEEE. https://doi.org/10.1109/ISCAS.2011.5937553

[14] Janik, J. M., & Fresnaud, V. (2007). A spectral approach to estimate the INL of A/D converter. Computer Standards & Interfaces, 29(1), 31-37. https://doi.org/10.1016/j.csi.2005.12.004

[15] Duan, J., Jin, L., & Chen, D. (2012). Testing ADC spectral performance without dedicated data acquisition. IEEE Transactions on Instrumentation and Measurement, 61(11), 2941-2952. https://doi.org/10.1109/TIM.2012.2202949

[16] Ciofi, C., & Neri, B. (2003, May). Low-frequency noise measurements: applications, methodologies and instrumentation. In Noise in Devices and Circuits (Vol. 5113, pp. 350-367). SPIE. https://doi.org/10.1117/12.489652

[17] Van de Plassche, R. J. (2012). Integrated analog-to-digital and digital-to-analog converters (Vol. 264). Springer Science & Business Media. https://doi.org/10.1007/978-1-4615-2748-0

[18] Hou, N., Song, W., Zhu, Y., Sun, X., & Li, W. (2018). Dynamic and static performance optimization of dual active bridge DC-DC converters. Journal of Modern Power Systems and Clean Energy, 6(3), 607-618. https://doi.org/10.1007/s40565-017-0343-7

[19] Ambardar, A. (1995). Analog and digital signal processing (p. 700). Boston, MA, USA: PWS.

[20] Chen, D., Cui, X., Zhang, Q., Li, D., Cheng, W., Fei, C., & Yang, Y. (2022). A survey on analog-to-digital converter integrated circuits for miniaturized high resolution ultrasonic imaging system. Micromachines, 13(1), 114. https://doi.org/10.3390/mi13010114

[21] Verma, A., & Saini, B. S. (2024, February). Comparative Analysis and Review of ADCs and Comparators for Use in Biomedical Applications. In 2024 2nd International Conference on Computer, Communication and Control (IC4) (pp. 1-6). IEEE. https://doi.org/10.1109/|C457434.2024.10486271

[22] Georgoulopoulos, N., Mamali, T., & Hatzopoulos, A. (2024). Design and Verification of a SAR ADC SystemVerilog Real Number Model. Journal of Electronic Testing, 40(3), 315-328. https://doi.org/10.1007/s10836-024-06124-8

[23] Feng, B., Mera, A., & Lu, L. (2020). {P2IM}: Scalable and hardware-independent firmware testing via automatic peripheral interface modeling. In 29th USENIX Security Symposium (USENIX Security 20) (pp. 1237-1254). https://doi.org/10.5555/3489212.3489282

[24] Navabi, Z. (2011). Digital system test and testable design. Springer. https://doi. org/10.1007/978-1-4419-7548-5.

[25] Egea, X., Sanchis, E., González, V., Gadea, A., Blasco, J. M., Barrientos, D., ... & Wadsworth, R. (2012, June). Design and test of a high-speed flash ADC mezzanine card for high-resolution and timing performance in nuclear structure experiments. In 2012 18th IEEE-NPSS Real Time Conference (pp. 1-8). IEEE. https://doi. org/10.1109/RTC.2012.6418150

[26] van den Hoven, R. A. T. (2006). A pre-correction method for improved static linearity using parallel DACs.

[27] Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794). https://doi.org/10.1145/2939672.2939785

[28] Sun, X., Lin, Y., & Yin, Q. (2025). Digital control circuit of accelerometer in gravity measurement with PSO optimized ADC-DSP integration. Informatica, 49(1), 109–116. https://doi.org/10.31449/inf.v49i23.8444

[29] Š trumbelj, E., & Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. knowledge and information systems, 41, 647-665. https://doi.org/10.1007/s10115-013-0679-x

[30] Chefrour, H. (2021). Unsupervised deep learning: Taxonomy and algorithms. Informatica, 45(1), 21–32. https://doi.org/10.31449/inf.v46i2.3820

[31] Swartzlander, E. E., & Goto, G. (2017). Computer arithmetic. In Digital Design and Fabrication (pp. 11-1). CRC Press. https://doi.org/10.1201/9780849386046-11

[32] Lee, Y. C., Chan, Y. K., & Koo, V. C. (2021). Design and Implementation of Field-Programmable Gate Array Based Fast Fourier Transform Co-Processor Using Verilog Hardware Description Language. Progress In Electromagnetics Research B, 92. https://doi.org/10.2528/PIERB20122806

A Hybrid RF-CART-SMOTE-GA Model for Early Warning in University Budget Management

Xu Qin
Jiangsu University Jingjiang College, Zhenjiang 212000, China
E-mail: 13815480090@163.com

Keywords: budget management, information warning, random forest, SMOTE, university finance

Received: June 20, 2025

With the advent of the big data era, university budget management faces heightened demands, necessitating effective responses to challenges arising from diversified funding sources and expanded scale. To address these challenges, this study employs the Random Forest algorithm as the foundation for a budget information early warning model. It combines the Classification and Regression Trees (CART) algorithm to optimize the decision tree structure, introduces weighted ensemble voting to enhance the classification process, and achieves data balancing and parameter optimization through Synthetic Minority Over-sampling Techniques and genetic algorithms. The model was validated using two datasets: the Integrated Postsecondary Education Data System (IPEDS) and the Higher Education Statistics Agency (HESA). Experimental results demonstrated that for the classification task during the budgeting phase, the model achieved a maximum classification accuracy of 94% on the HESA dataset, with a recall of 92.18%, an F1 score of 92.87%, and a sample balance rate of 81.64%, with the lowest budget information early warning error at 3.7%. Additionally, the average runtime was 0.04 seconds, and CPU utilization was only 24.17%, significantly outperforming models such as DBN, XGBoost, and VAE-TSAD. The research results demonstrate that the proposed model possesses high accuracy, real-time capability, and computational efficiency in early warning for university budget management information, providing reliable technical support for higher education financial decision-making.

Povzetek: Članek predstavi hibridni ansambelski model RF-CART-WIV-SMOTE-GA za zgodnje opozarjanje v univerzitetnem proračunskem upravljanju s pomočjo večnivojske optimizacije strukture.

1 Introduction

With the expansion of university scale, improvement of education level, and abundant educational resources, how to scientifically, reasonably, and efficiently manage budgets has become an important issue that university managers urgently need to solve [1]. Budget management is an important component of financial management in universities, which is related to the development of various undertakings and rational resource allocation, as well as the economic and educational benefits of universities [2]. The primary challenges currently facing university budget management include: (1) Strong data heterogeneity and low availability; (2) Time lag between budget execution and performance feedback; (3) Insufficient risk identification accuracy and delayed resource allocation response. Therefore, scholars at home and abroad have successively explored solutions. To further enhance the effectiveness of budget management information warning in universities, Teng et al. proposed a new warning method by combining data mining and deep learning models. The experimental results showed that the accuracy of the financial risk warning method was the highest, at 97.62%, which was 13.57% higher than that of traditional methods [3]. Valle-Cruz et al. proposed a new warning model that combined multi-layer perception algorithms to improve the clarity of allocation and expenditure in the budget execution process of universities. The experimental results showed that the model had high real-time and accuracy in financial budget information early warning of universities in the past three years [4]. Truong et al. argued that there were still difficulties in conducting timely financial management alerts in crowdsourced competition programs. For this reason, a budget management early warning strategy based on the HAIS algorithm was proposed. The experimental results showed that the warning time lag of this strategy was as low as 0.13 seconds, which was better than that of the traditional method [5]. Zheng et al. found that federated learning was highly effective in early warning of budget management information in universities, but at a higher cost. They proposed a fuzzy logic based federal learning budget information warning model. Experimental results showed that the effectiveness of the budget information early warning effectiveness far exceeded other existing methods [6].

Random Forest (RF) algorithm is an ensemble learning method that learns historical budget data by constructing a model to identify abnormal data that is inconsistent with the normal budget pattern, thereby achieving early warning of budget overruns, abnormal fund flows, and other situations [7]. To reduce the RF error between the revenue budget and expenditure budget

of universities, Bénard et al. took the Sobol index to propose a new early warning method. The experimental results showed that the method could occur before the crisis to make effective judgments and early warnings, and the effectiveness was high [8]. Zhang et al. found that market economy fluctuations had a significant impact on budget management in universities. Therefore, a new budget warning algorithm was proposed by combining RF algorithm and logistic regression model. The experimental results showed that the new algorithm had the highest accuracy of 93.47% for teaching budget warning in three universities in mainland China [9]. Cao et al. proposed a new warning model for the risk warning level in the new university budget management based on RF and convolutional neural networks. The experimental results showed that the model had a high effectiveness of 96.38% in risk warning for campus construction budget and campus teaching budget [10]. Zeng found that there were still many risks in the current financial budget management of universities. Therefore, the RF and backpropagation neural network were combined for risk data mining. Finally, a new warning model was proposed. The experimental results showed that the new model had the highest prediction accuracy of 91.6% and the shortest prediction time of 7.65 s [11]. The comparison for each document is shown in Table 1.

As shown in Table 1, existing research primarily focuses on model structure innovation or feature optimization, yet significant shortcomings remain in data imbalance handling, classification stability, and computational efficiency. In contrast, the proposed RF-CART-WIV-SMOTE-GA model not only integrates the advantages of multiple algorithms but also achieves data balancing and optimization through the collaborative optimization of SMOTE and GA. achieving 94% accuracy and 3.7% Mean Absolute Error (MAE) on both the IPEDS and HESA datasets. It also demonstrates significant improvements in runtime and CPU utilization compared to existing State-of-the-art (SOTA) models.

In summary, many researchers at home and abroad have conducted in-depth exploration on budget information management in universities and proposed various processing methods. These methods can to some extent complete the warning work of budget information, but there are still challenges such as low warning accuracy and high warning cost. Therefore, the study introduces a multi-layer optimization mechanism based on the RF model to construct a high-precision early warning model for university budget management. Specifically, at the structural level, Classification and Regression Trees (CART) are employed to optimize the decision tree partitioning process, while Weighted Integration Voting (WIV) is adopted at the decision level to enhance classification robustness.

At the data level, the Synthetic Minority Over-sampling Technique (SMOTE) balances budget information samples. At the optimization level, a Genetic Algorithm (GA) is introduced for global parameter optimization. Unlike traditional bagging or boosting methods, WIV significantly reduces bias correlation among weak classifiers through its weight-adaptive adjustment strategy. This approach achieves higher classification accuracy and model stability in the imbalanced data environment of budget management.

This study aims to use a multi-algorithm fusion integrated optimization framework to accurately classify and detect abnormal budget management data, thereby improving the intelligence level and risk control capability of financial decision-making in universities. To this end, the following research questions are proposed:

Question 1: How to improve classification accuracy and anomaly detection accuracy in situations where budget information data is highly imbalanced?

Hypothesis 1: Combining SMOTE and GA-based data balancing and optimization mechanisms can significantly enhance the classification performance of RF models on imbalanced datasets.

Question 2: Compared with traditional ensemble learning methods (e.g., Bagging, Boosting), can WIV improve alert stability under multi-indicator constraints?

Hypothesis 2: Introducing WIV optimizes the decision fusion process through adaptive weighting mechanisms, thereby enhancing model stability and generalization performance.

Question 3: Can the proposed RF-CART-WIV-SMOTE-GA model outperform existing models (e.g., DBN, XGBoost, VAE-TSAD) across the four stages of university budget formulation, execution, adjustment, and evaluation?

Hypothesis 3: The proposed model will significantly outperform comparable SOTA methods in key metrics including accuracy, recall, and runtime.

In summary, the innovation of this research manifests in three key aspects: (1) Establishing a multi-layer ensemble learning framework that integrates structural optimization, decision enhancement, and data balancing; (2) Proposing a weighted strategy using WIV to replace traditional ensemble methods and enhance robustness; (3) Validating the model's high accuracy and real-time capability in university budget early warning through empirical testing on IPEDS and HESA datasets, thereby providing scalable intelligent decision support tools for university financial management.

Table 1: Comparison of existing state-of-the-art (SOTA) early warning models for higher education budget management

Author (Year)	Algorithm / Model	Dataset	Accuracy / MAE	Computational Cost / Time	Data Imbalance Handling	Main Limitation	Reference
Teng Y *et al.* (2023)	Data Mining + Deep Learning	University Financial Data	97.62%	—	None	High model complexity, limited generalization	[3]
Valle-Cruz D *et al.* (2022)	MLP	Government University Data	≈95%	Good real-time performance	None	Applicable only to execution stage	[4]
Truong V Q *et al.* (2023)	HAIS	Crowdsourcing Dataset	—	0.13 s	None	High computational cost, limited scope	[5]
Zheng Z *et al.* (2022)	Federated + Fuzzy Logic	Smart City Data	≈93%	High	Partial	Expensive computation, privacy constraints	[6]
Bénard C *et al.* (2022)	RF + Sobol Index	Financial Dataset	≈92%	Medium	None	Poor adaptability to nonlinear fluctuation	[8]
Zhang Z *et al.* (2022)	RF + Logistic Regression	3 Chinese Universities	93.47%	—	None	Insufficient hierarchical budget analysis	[9]
Cao Y *et al.* (2022)	RF + CNN	Campus Budget Data	96.38%	Medium	None	Complex structure, difficult tuning	[10]
Zeng H *et al.* (2022)	RF + BPNN	Financial Dataset	91.60%	7.65 s	None	Long training time, sensitive to data bias	[11]
DBN / XGBoost / VAE-TSAD (baseline)	Deep Belief / Boosting / Variational Autoencoder	IPEDS / HESA	89–91%	0.07–0.11 s	None	Unstable on imbalanced data	/
Proposed Model (RF-CART-WIV-SMOTE-GA)	Hybrid Ensemble	IPEDS / HESA	94% / MAE 3.7%	0.04 s	SMOTE-balanced	High accuracy, stable performance, efficient computation	/

2 Methods and materials

In response to the various risk issues in existing efficient budget management information, the study first uses the RF as a basis, introduces CART to adjust its decision tree structure, and adopts WIV to optimize its decision-making process. Finally, a budget management information classification model is constructed. Before model training, all raw data undergoes systematic preprocessing to ensure the accuracy and consistency of model inputs. Firstly, perform missing value detection on the IPEDS and HESA datasets. If the missing ratio is less than 5%, use the mean or mode estimate. If it exceeds 5%, use K-Nearest Neighbor Imputation (KNN Imputation) for repair. Next, Z-score normalization is applied to the continuous features, and a hot encoding process is used to classify the variables. Use the 3σ rule and interquartile range (IQR) method to detect and remove outliers. Finally, by ranking the importance of RF features, key features that have a significant impact on budget information classification are selected as model inputs. In addition, to successfully achieve classified information warning, the research introduces the SMOTE algorithm to adjust imbalanced data, and take the GA to strengthen the optimal warning solution. Finally, the research proposes a new university budget information management warning model (RF-SMOTE).

2.1 Construction of information classification model for university budget management based on RF

University budget management generally includes four stages: budget preparation, budget execution, budget control, and budget performance assessment [12]. Budget preparation is the foundation of university budget management. Budget execution is the process of putting the prepared budget into practice. Budget

control is the supervision and adjustment of the budget execution process. Budget performance assessment evaluates the budget execution results. According to the description in the "Financial System of Higher Education Institutions", Figure 1 displays the life cycle of budget management information in universities [13-14].

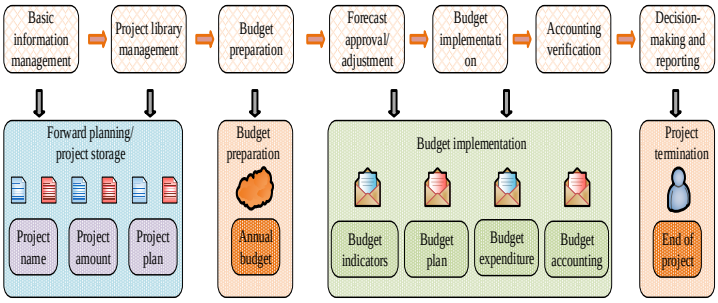


Figure 1: Schematic of the life cycle of budget management in higher education

As shown in Figure 1, when the basic management information is input into the system, it will undergo project library management, budget preparation, budget approval, budget adjustment, budget execution, and accounting verification before final settlement and reporting. Due to various reasons during the period, such as inadequate budget management, inability to dynamically meet budget requirements, inadequate connection between different levels of budget management, and incomplete budget systems, it is highly likely to result in inaccurate budgets, reduced budget management efficiency, and resource waste. To this end, the study introduces RF for multi-level early warning of university budget management information, in order to avoid the interference and intrusion of these abnormal information. Compared with other algorithms, RF algorithm can evaluate the importance of features and help identify key indicators that affect budget performance, providing a basis for budget preparation and adjustment [15]. Figure 2 displays the principle of RF.

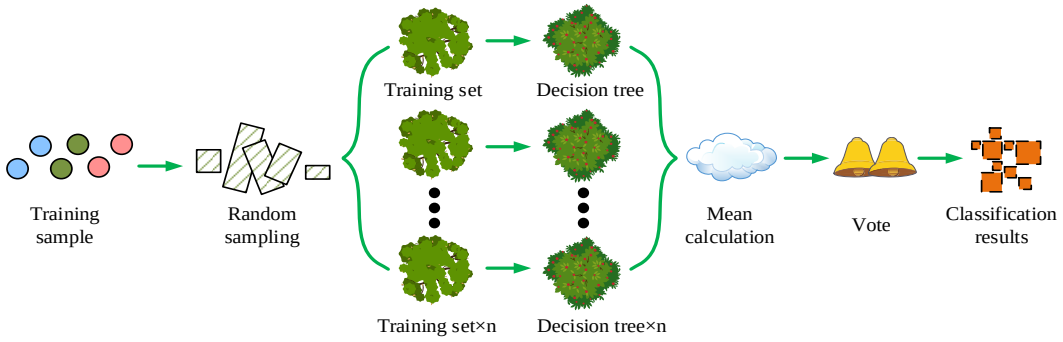


Figure 2: RF principle schematic

As shown in Figure 2, RF divides the original training samples into multiple sub samples after random sampling. When selecting segmentation nodes, randomly select some features to search for the best segmentation point, and then construct their own decision trees based on each sub sample. In general classification tasks, RF determines the final prediction result through majority voting, while in regression tasks, RF calculates the average of all decision tree prediction results to obtain the final prediction result [16]. RF needs to select the optimal splitting attribute based on indicators such as information gain and information gain rate. Assuming the initial sample set is D, the information entropy is shown in equation (1).

$$Entrpy(D) = -\sum_{i=1}^{c} P_i \log_2 P_i \tag{1}$$

In equation (1), P_i represents the proportion of class i in the sample set. $Entrpy(D)$ represents the information entropy of the original sample set. c represents the total number of samples. After introducing feature data A, the sample set is classified as information entropy, information gain, information value, and information gain rate, and their respective calculations are shown in equation (2).

$$\begin{cases} Entrpy(D_A) = -\sum_{i=1}^{k} \frac{|D_j|}{|D|} Entropy(D_A) \\ Gain(D,A) = Entropy(D) - Entropy(D_A) \\ SplitEntropy(D,A) = -\sum_{i=1}^{c} \frac{|D_i|}{|D|} \log_2 \frac{|D_i|}{|D|} \\ GainRation(D,A) = \frac{Gain(D,A)}{SplitEntropy(D,A)} \end{cases} \quad (2)$$

In equation (2), the set of values for the discrete feature A is denoted as $\{a_1,\ldots,a_k\}$. Subdivisions based on the values of A yield subsets $D_1,\ldots,D_k$, where $|D_j|$ represents the number of samples in the j-th subset and $|D|$ denotes the total number of samples. k represents the total number of parts that the set D is divided into. $Gain(D,A)$ represents the information gain of feature A in set D. $SplitEntropy(D,A)$ represents the information value of feature A in set D. $GainRation(D,A)$ represents the information gain rate of feature A in set D. Due to the random sampling method of RF, it is easy to cause strong correlation between adjacent decision trees, resulting in poor classification results of the final information data. For this purpose, the research takes CART to optimize the decision tree generation of RF in the study. Compared with other methods, CART is more suitable for RF because it is trained based on information entropy, avoiding more parameter settings and reducing the complexity of the RF optimization process [17]. The optimal index minimization node classification for CART is shown in equation (3).

$$Gini(D) = 1 - \sum_{i=1}^{c} P_i \quad (3)$$

In equation (3), $\sum_{i=1}^{c} P_i = 1$ is specified for $Gini_{split}(D) = \sum_{j=1}^{k} \frac{|D_j|}{|D|} Gini(D_j)$. When used to minimize node impurity, $argmin_A Gini_{split}(D)$ is selected. $Gini(D)$ represents the minimum node splitting index of the sample set D. At this point, the coefficient expression of the minimization index is shown in equation (4).

$$Gini_{split}(D) = \sum_{i=1}^{l} \frac{|D_i|}{|D|} Gini(D_i) \quad (4)$$

In equation (4), D_i represents the minimum index coefficient at the i-th set after partitioning according to l subsets. At this point, the optimal indicator for RF segmentation nodes is changed to information gain rate, which serves as the basis for node segmentation judgment. The information entropy, information gain, information value, and information gain rate of the improved RF are shown in equation (5).

$$\begin{cases} Entrpy(D_B|A) = -\sum_{i=1}^{l} \frac{|D_{Bi}|}{|D_B|} Entropy(D_B) \\ Gain(D_B|A) = Entropy(D_B) - Entropy(D_B|A) \\ SplitEntropy(D_B|A) = -\sum_{i=1}^{l} \frac{|D_{Bi}|}{|D_B|} \log_2 \frac{|D_{Bi}|}{|D_B|} \\ GainRation(D_B|A) = \frac{Gain(D_B|A)}{Gini_{Split}(D_B|A)} \end{cases} \quad (5)$$

In equation (5), D_B represents the B-th set. B denotes the B-th sub-sample set/tree obtained via bootstrap sampling, and D_B represents its training samples. Equation (5) is isomorphic to equation (2), but replaces the population D with D_B for the split metric on a single tree (or subset). In addition, to enhance the impact of attribute differences in different decision trees on the final data information classification and improve the effectiveness of management information classification, the final decision of RF is optimized by WIV, and the classifier is processed in a weighted manner. This process is shown in equation (6).

$$\omega_i = \sum_{l} P_{il} \Big/ \sum_{il} P_{il} \quad l = 1,2,3,\cdots,c \quad (6)$$

In equation (6), ω_i represents the weight of the i-th sample. P_{il} represents the probability of the i-th sample after the classifier splits the sample into l subsets. P_i denotes the normalization factor used for weighting (e.g., the prior proportion of class i in the training set or the validation set performance score), sharing the same sign but differing in meaning from the class probability in equation (1). The confidence calculation of the weighted classification results is shown in equation (7).

$$u_l(D) = \frac{1}{N} \sum_{i=1}^{N} \omega_i P_{il}(D) \quad (7)$$

In equation (7), $u_l(D)$ represents the confidence value of dividing the sample into l split subsets. A new classification model for university budget management information is proposed based on split improvement and voting improvement in RF. The process of this model is shown in Figure 3.

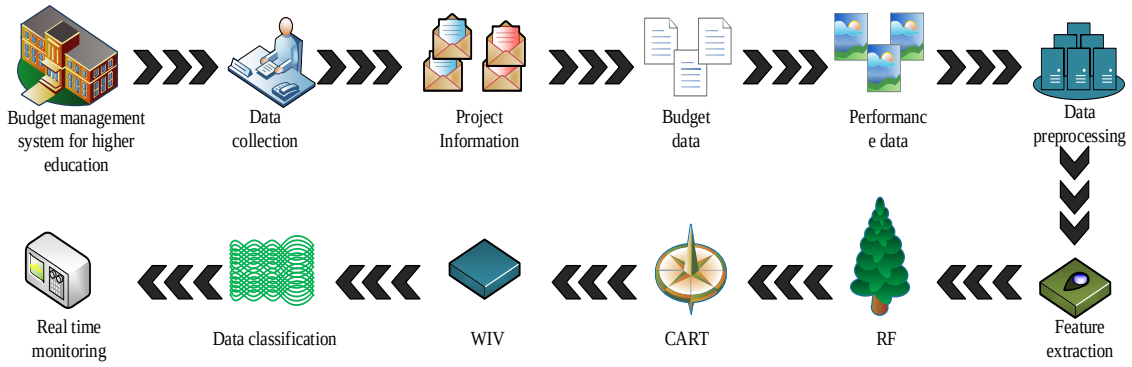


Figure 3: RF-CART-WIV information categorization model flow

As shown in Figure 3, firstly, all relevant data are collected from the university budget management system, including project information, budget data, performance data, etc. Secondly, perform data preprocessing to ensure the quality and consistency of the data. Then, extract useful features from the raw data, including amounts, project names, etc. After completion, the RF is used to evaluate the importance of each feature and select key features that have a significant impact on budget management. During this period, CART optimizes the decision tree structure of RF, and WIV optimizes the decision process of the classifier. Next, real-time data classification is performed after validation on the training set, and the classification results are monitored in real-time, with regular updates to the model.

2.2 Training and enhancement of early warning model with SMOTE algorithm fusion

After constructing a classification model for budget management information data in universities, there are still technical drawbacks in evaluating and making decisions based on different classification data. For example, noisy data after budget management information classification can cause overlapping classification results. To use information classification results for effective judgment and connection, and provide decision support for management, including adjusting budget allocation, optimizing resource allocation, or taking preventive measures, the study first investigates the sources of income and expenditure data that have the greatest impact on budget allocation. The sources of income and expenditure in universities are shown in Figure 4. To enhance the credibility and data support of Figure 4, the study further incorporates publicly available statistical data samples. Referencing the "2022 National Education Expenditure Implementation Statistics Bulletin" issued by the Ministry of Education and the Ministry of Finance, along with financial annual reports from selected universities, the following typical feature distributions are extracted: Fiscal allocations account for approximately 56% of total university revenue, institutional income accounts for about 27%, operational income accounts for roughly 9%, and other income accounts for approximately 8%. Correspondingly, educational expenditures constitute about 63% of total expenditures, personnel expenses account for 18%, project funding represents 12%, and operational expenditures make up 7%. This distribution pattern serves as the basis for setting input data proportions in the model and is reflected in the training sample structure of the subsequent RF-SMOTE model. This ensures that the income-expenditure relationship depicted in Figure 4 possesses practical statistical significance and representativeness.

As shown in Figure 4, the income includes government subsidy income, business income, operating income, and other income. In terms of expenditure, it can be roughly divided into education expenditure, health expenditure, employment security expenditure, personnel expenditure, project expenditure, and operating expenditure. According to the "Statistical Announcement on the Implementation of National Education Funds in 2022" released by the Ministry of Education, the National Bureau of Statistics, and the Ministry of Finance in 2022, in some cases, the financial budgets of universities have shown an imbalance where income is less than expenditure [18-19]. Therefore, to balance the input data of the warning model, the SMOTE algorithm is introduced. This algorithm balances the dataset by generating synthesized minority class samples, rather than simply copying minority class samples, thus avoiding overfitting. The new sample generated by SMOTE algorithm is shown in Figure 5.

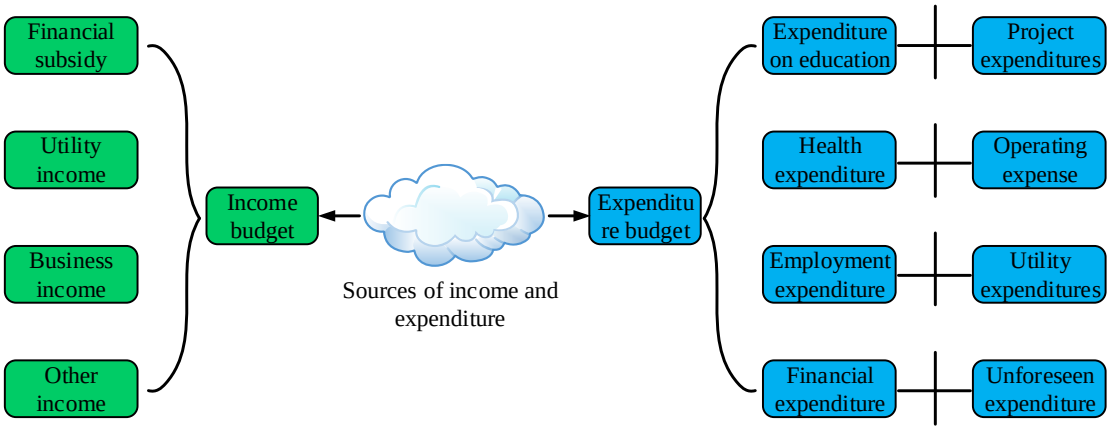


Figure 4: Demonstration of the sources of income and expenditure of higher education finances

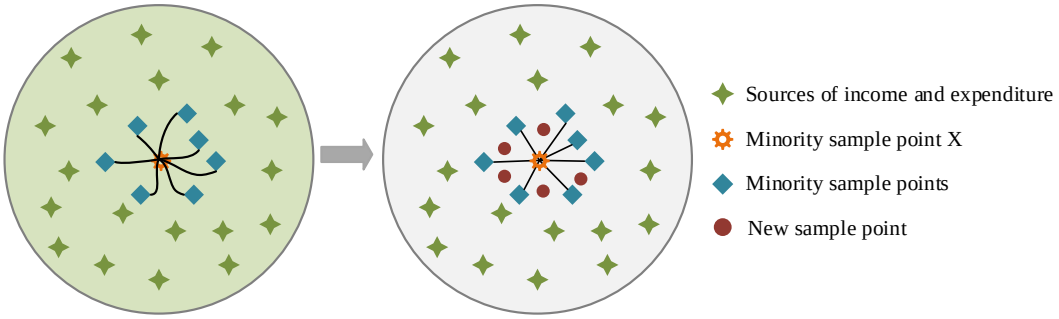


Figure 5: Schematic of SMOTE new sample generation

As shown in Figure 5, when there is an imbalance between financial income and expenditure data, in samples with a small amount of data, the nearest k neighboring samples of sample point x in the minority class can be found based on the Euclidean distance, and the upsampling magnification is set. For each selected neighbor, calculate the distance between the original sample and the neighbor, and generate a new composite sample from the random points on this line. The method for generating the new sample is linear interpolation between the original sample and the neighbor sample. The interpolation is shown in equation (8).

$$\begin{cases} Q_i = X + rand(0,1) \times (y_i - X) \\ i = 1, 2, ..., n \end{cases} \tag{8}$$

In equation (8), Q_i represents the generated interpolated sample points. X represents the data sample point in the minority class. For each minority class sample point X, its corresponding nearest neighbor sample is found. The i-th nearest neighbor sample is represented as y_i. Finally, the maximum value of i is n, which is the upsampling magnification. The imbalance in the dataset is calculated in equation (9).

$$n = ROUND(IL) \tag{9}$$

In equation (9), IL denotes the imbalance degree. n_{maj} represents the number of majority class samples and $n_{\min}$ represents the number of minority class samples. Then, the imbalance degree IL is defined by equation (9) (e.g., commonly used $IL = n_{maj} / n_{\min}$ or $IL = (n_{maj} - n_{\min}) / n_{\min}$. IL represents the degree of imbalance. $ROUND(IL)$ represents the rounded value of IL. However, the simple SMOTE does not differentiate different samples when processing minority class samples, but uniformly generates new samples using the same method, which to some extent limits its applicability. Therefore, GA optimizes the sample selection in the study, which uses the roulette wheel selection method to make the newly synthesized samples more likely to cluster around high fitness samples [20]. By calculating the cumulative value of probabilities, the relative weight of each sample in the roulette wheel process is determined, as shown in equation (10).

$$cump(j) = \sum_{i=1}^{j} Q(i) \tag{10}$$

In equation (10), $Q(i)$ represents the probability of randomly selected minority sample points. At this

point, the probability of selecting minority class samples is shown in equation (11).

$$Q(i)=\frac{k+1-f(x_i)}{\sum_{i=1}^{k}(k+1-f(x_i))} \tag{11}$$

In equation (11), $f(x_i)$ denotes the fitness of minority class samples, with a recommended range normalized to [0,1]. k represents the nearest neighbor parameter. Subsequently, the posEff function is used to calculate the generated sample distance, and the relative distance ratio is shown in equation (12).

$$u'=\left(\frac{d_1}{d_{xy}}\right)\times\left(\frac{d_2}{d_{xz}}\right) \tag{12}$$

In equation (12), d_{xy} and d_{xz} represent the average Euclidean distances between minority samples and majority samples, respectively. d_1 and d_2 respectively represent the distance between the sample and its nearest neighbors of the same class and the majority of its nearest neighbors. To further elucidate the specific implementation mechanism of the GA in the minority class sample selection process, this study explains its fitness function and roulette wheel selection. Firstly, to measure the representativeness and separability of minority class samples, the fitness function $f(x_i)$ is defined in equation (13).

$$f(x_i)=\alpha\cdot\frac{d_{\text{int}ra}(x_i)}{d_{\text{int}er}(x_i)}+\beta\cdot\left(1-u_i^{'}\right) \tag{13}$$

In equation (13), $d_{\text{int}ra}(x_i)$ denotes the average Euclidean distance between sample x_i and its class neighbors. $d_{\text{int}er}(x_i)$ denotes the average Euclidean distance between sample x_i and the nearest majority-class samples. $u_i^{'}$ denotes the relative distance ratio defined in equation (12). α and β are both weight coefficients. A smaller $f(x_i)$ value indicates that the sample resides in a dense region within its class and is distant from the majority-class boundary, making it more suitable for synthesis. Secondly, a roulette wheel mechanism selects samples based on their fitness distribution. The sample selection probability $Q(i)$ and cumulative probability $cump(j)$ are determined by equations (10)-(11), respectively, with the calculation shown in equation (14).

$$Q(i)=\frac{\varsigma+1-f(x_i)}{\sum(\varsigma+1-f(x_i))},cump(j)=\sum_{i=1}^{j}Q(i) \tag{14}$$

In equation (14), ς represents the translation term used to prevent the denominator from becoming zero. The algorithm achieves probabilistic selection by generating a random number r ∈ [0,1] and locating the corresponding sample index within the cumulative probability interval. The roulette selection pseudocode is as follows:

```
Input: Minority sample set X_min, neighbor number k, population size N_pop,
       maximum generation G_max, fitness function f(x_i), crossover rate p_c, mutation rate p_m
Output: Optimized minority sample subset X_opt

1  Initialize population P with N_pop minority samples randomly selected from X_min
2  For generation g = 1 to G_max do
3     Evaluate fitness f(x_i) for each individual in P using Eq.(f)
4     Normalize fitness and compute selection probability Q(i) and cumulative prob cump(j)
5     For i = 1 to N_pop do
6        Generate random number r ∈ [0,1]
7        Select parent p_i such that cump(j-1) < r ≤ cump(j)
8     Apply crossover (probability p_c) and mutation (probability p_m) to generate new population P'
9     Replace P ← P'
10 End For
11 Return individuals with minimum f(x_i) as X_opt
```

To improve the specificity of GA optimization, this study simultaneously monitored indicators such as classification accuracy, sample balance rate, and MAE during the iteration process of the genetic algorithm. These metrics serve as comprehensive performance evaluation criteria for sample selection schemes. By comparing the improvement margins of these metrics in each iteration, the GA dynamically identifies minority samples that significantly boosts overall model performance, thereby finely optimizing the SMOTE generation process. The hyperparameter table is shown in Table 2.

In summary, based on the previously proposed RF-CART-WIV information classification model for university budget management, a new warning model, RF-CART-WIV-SMOTE-GA, is proposed. The warning process is shown in Figure 6.

Table 2: Main hyperparameter settings for the proposed RF-CART-WIV-SMOTE-GA model

Module	Symbol / Parameter Name (in Formulas)	Description	Value / Determination Method	Formula Reference / Basis
RF / CART (Splitting & Voting)	B	Number of bootstrap sub-samples / trees ((D_B))	200	Matches (D_B) in Eq. (5); grid-search optimum balancing accuracy and runtime
	$\mathrm{Gain}(D,A)$, $\mathrm{SplitEntropy}(D,A)$, $\mathrm{GainRation}(D,A)$	Information gain, split entropy and gain ratio metrics	/	Eqs. (2) for information-based splits
	$\mathrm{Gini}(D)$, $\mathrm{Ginisplit}(D)$	Gini impurity and weighted Gini for splitting	/	Eqs. (3), (4) for minimal impurity selection
Weighted Integration Voting	ω_i	Weight of the *i*-th sample or class	Computed automatically	Eq. (6)
	$u_i(D)$	Confidence value after weighted voting	Computed automatically	Eq. (7)
SMOTE	k	Number of nearest neighbors	5	Used for neighbor search and also in GA selection (Eq. 11 denominator range)
	$Q_i=X+rand(0,1)\times(v_i-X)$	Formula for synthetic sample generation	/	Eq. (8)
	IL	Imbalance Level indicator	Calculated per class	Eq. (9), drives sampling ratio
	$ROUND(IL)$	Rounded oversampling percentage	≈ 150 %	Derived from Eq. (9) after rounding
GA	$cump(j)=\sum_{i=1}^{j} Q(i)$	Cumulative probability in roulette selection	Computed	Eq. (10)
	$Q(i)=\frac{k+1-f(x_i)}{\sum(k+1-f(x_i))}$	Selection probability for minority sample (x_i)	Computed	Eq. (11)
	$f(x_i)$	Fitness function of minority sample (x_i)	Computed	Eq. (11)
	$u'=\left(\frac{d_1}{d_{xy}}\right)\cdot\left(\frac{d_2}{d_{xz}}\right)$	Relative distance ratio (posEff) for sample suitability adjustment	Computed	Eq. (12)

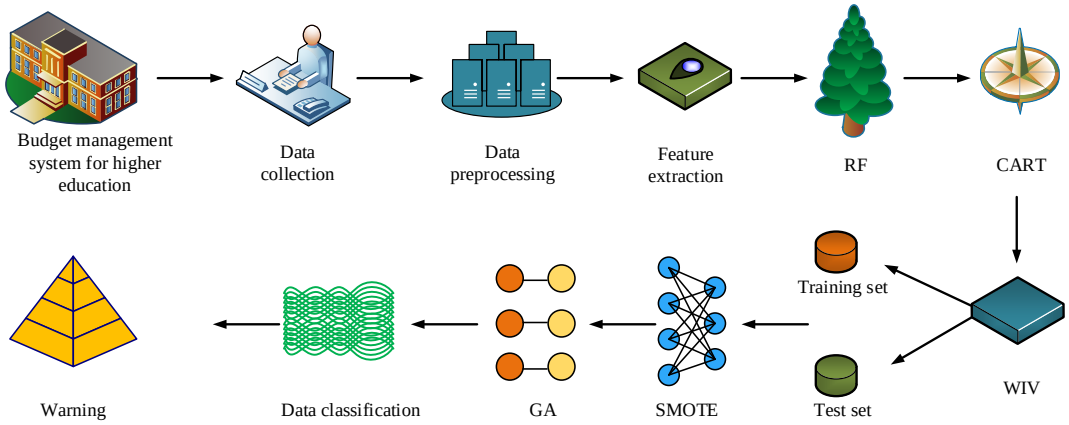


Figure 6: A novel early warning modeling process for budget management information in higher education institutions

As shown in Figure 6, firstly, collect all relevant data from the university budget management system. Secondly, perform data preprocessing to extract useful features such as quantity and project names. Thirdly, use RF algorithm to evaluate the importance of features, select key features, and introduce CART to optimize the decision tree structure of RF, while introducing WIV to optimize the classification process. Fourthly, divide the dataset into a training set and a testing set for model training and parameter adjustment. Fifthly, to handle imbalanced data, introduce the SMOTE algorithm to generate synthesized minority class samples, combined with genetic algorithm to optimize sample selection, improving the efficiency and accuracy of SMOTE. Sixthly, deploy the optimized RF-CART-WIV-SMOTE-GA model to the university budget management system. Perform real-time data classification and warning, continuously monitor and evaluate the effectiveness of the model, and regularly update it to adapt to new data and requirements. The algorithm pseudocode is as follows:

Algorithm 1: RF-CART-WIV-SMOTE-GA Budget Information Early Warning Model

Input: Original budget dataset D (IPEDS or HESA)

Output: Predicted budget category and early warning signal

// Data Pre-processing

Clean D; handle missing values (by mean/mode or KNN); standardize continuous features; encode categorical ones.

Detect outliers using 3σ and IQR rules; remove extreme points.

// Feature Selection

Use Random Forest importance ranking to select top-K budget-related features.

// Model Training with CART and WIV

For each bootstrap subset D_B (b = 1 … B):

Train CART tree using GainRatio and Gini criteria (Eqs. 2–5);

Compute weighted vote ω_i and confidence u_i(D) (Eqs. 6–7).

Aggregate all trees using Weighted Integration Voting.

// Data Balancing via SMOTE + GA

For each minority class sample X:

Find k nearest neighbors v_i; generate synthetic sample $Q_i = X + rand(0,1)\cdot(v_i - X)$ (Eq. 8);

Compute imbalance level IL and ROUND(IL) (Eq. 9).

Use GA to optimize sample selection probability Q(i) and fitness f(x_i) (Eqs. 10–11);

Adjust by posEff distance ratio u' (Eq. 12).

// Model Integration and Prediction

Combine balanced dataset and trained RF-CART-WIV model;

Perform prediction and compute error metrics (Accuracy, Recall, F1 score, and MAE).

Return classified budget stage and early warning signal.

2.3 Model deployment and system architecture design

To validate the proposed model's integrability and applicability within university budget management information systems, the study further designs a system deployment architecture. The overall system adopts a four-layer structure comprising:

Data Collection Layer: Responsible for automatically extracting raw budget data from university financial databases and business management platforms, including project IDs, expenditure categories, amounts, and timestamps. It is connected to the system through ETL connection.

Algorithm Service Layer: Integrates the proposed RF-CART-WIV-SMOTE-GA model to provide budget information classification and risk warning services. This module is implemented using Python and the Scikit-learn framework, supports RESTful API calls, and can be deployed on Linux servers or university private cloud environments.

Application Interface Layer: Seamlessly connects with existing budget management systems via an API gateway. Users can invoke functional modules such as "Budget Anomaly Detection," "Expenditure Trend Forecasting," and "Early Warning Report Generation" through the system frontend, enabling real-time access to model outputs.

Visualization Layer: Presents model warning results and performance metrics through a web-based visual dashboard, supporting dynamic filtering and report exporting by department, project, and time dimension.

Furthermore, to ensure fairness and transparency in actual budget decision-making, the research specifically addresses data governance and algorithmic robustness during system design. Firstly, all input data undergoes anonymization and standardization during model training to prevent bias toward specific groups due to differences in department size or funding structure. Secondly, the system provides interpretable outputs in result presentation, including key feature contribution visualization and decision path visualization, to enhance the transparency and traceability of budget alerts. Simultaneously, to prevent false positives caused by external adversarial inputs (such as anomalous or misleading budget entries), the study introduces an anomaly detection mechanism based on Z-scores and IQR during data preprocessing. At the model level, a multi-round voting strategy using a RF sub-model is employed to mitigate noise interference.

3 Results

To comprehensively evaluate the model's performance in early warning for university budget management information, the study selected four core classification metrics: accuracy, precision, recall, and F1 score. Accuracy reflects the overall proportion of correct judgments made by the model. Precision measures the reliability of warning results. Recall indicates the model's sensitivity in identifying abnormal budgets. F1 score, as the harmonic mean of precision and recall, balances the detection capability and false alarm control. Additionally, to validate the model's feasibility in practical scenarios, MAE and sample balance rate are employed to measure the deviation of warning results and data balance.

3.1 Performance testing of university budget management information warning model

The research sets the CPU to an 8-core AMD Ryzen 7 4800H processor, GPU to NVIDIA GeForce, and memory to 64GB. The operating system is Ubuntu 20.04, and the experimental software includes Python 3.8 and the Scikit-learn library. The Integrated Postsecondary Education Data System (IPEDS) and the Higher Education Statistics Agency Finance Dataset (HESA) are used as the testing data sources. The above datasets have undergone data preprocessing, including data cleaning, data conversion, data integration, and data specification. These two datasets are divided into training and testing sets in an 8:2 ratio, used for ensemble training of the initial model. Both datasets exhibit significant category imbalance. Taking the budget execution phase as an example, the ratio of normal to abnormal samples is approximately 7:1 in the IPEDS dataset and 6:1 in the HESA dataset. Some sub-categories (such as special construction or research budget anomalies) account for less than 10% of the samples. Therefore, the SMOTE algorithm was introduced during model training to mitigate data bias, combined with GA optimization for sample selection to enhance the recognition capability of minority class. Additionally, hierarchical random segmentation was used to maintain consistency in the proportion of training and testing set categories and budget stages. The random seed was set to 42. All upscaling and parameter tuning were performed exclusively on the training set, with no resampling applied to the test set. To prevent data leakage from feature overlap, a dual-isolation strategy based on institution and time was implemented during data partitioning. Data from the same institution in the same year appeared exclusively in either the training or test set, with no cross-use. Cross-year samples were partitioned chronologically before stratified random sampling. This method ensures that the model cannot access any features originating from or appearing simultaneously with the training set during testing, thereby ensuring the independence and reliability of the evaluation results. The study first conducts ablation testing on the final model to verify the effectiveness of various modules. The test results are shown in Figure 7.

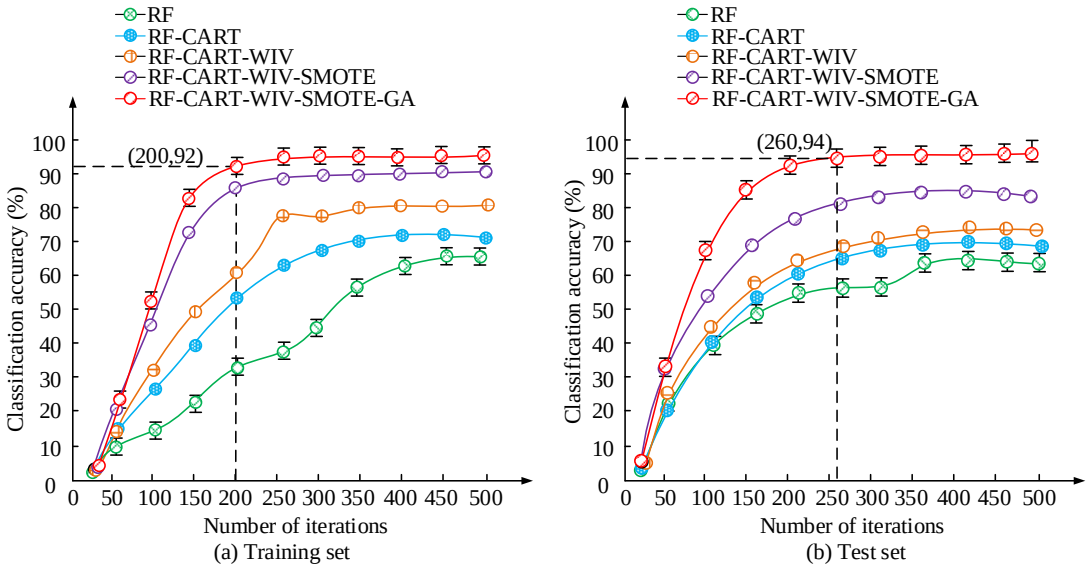


Figure 7: Ablation test results for a new budget information early warning model

Figure 7(a) shows the data classification test results of the new budget warning information model in the training set. Figure 7(b) shows the data classification test results of the new budget warning information model in the testing set. The horizontal axis represents model structure combinations, while the vertical axis shows average classification accuracy (unit: %). In the figure, “RF,” “CART,” “WIV,” “SMOTE,” and “GA” denote Random Forest, Classification and Regression Trees, Weighted Integrated Voting, Synthetic Minority Over-sampling Technique, and Genetic Algorithm modules, respectively. Budget data encompasses four categories: budget formulation expenditure items, fund disbursement records during implementation, budget adjustment projects, and annual evaluation indicators. From Figure 7(a), the budget information classification accuracy of the individual RF model gradually increased after training, with a maximum classification accuracy of 68%. After sequentially introducing CART, WIV, SMOTE, and GA, the budget information classification accuracy of the final warning model reached up to 92%. In Figure 7(b), similar to the training set, the introduced new modules in the testing set had a significant improvement effect on the final model, especially the RF-CART-WIV-SMOTE-GA model, which had a budget information classification accuracy of up to 94% and a minimum of 260 iterations. To further validate the independent contributions and stability impacts of each improvement module on model performance, multiple ablation experiments were conducted, with results shown in Table 3.

As shown in Table 3, the baseline RF model achieved an average classification accuracy of only 68.21%. After incorporating CART structure optimization, the model's feature partitioning capability significantly improved, and the accuracy increased to 78.62%. When the WIV module was further introduced, the model's confidence in voting decisions increased, with all metrics improving by approximately 7%. Subsequently, the SMOTE algorithm improved sample balance, resulting in a significant increase in recall and F1 score, with an average accuracy of 90.31%. Finally, after incorporating GA optimization, the model achieved optimal overall performance, with average accuracy rising to 93.94% and standard deviation decreasing to 0.40, indicating greater stability in model results. Overall, the sequential integration of modules not only improves classification accuracy and recall, but also effectively reduces model uncertainty. This demonstrates that the proposed RF-CART-WIV-SMOTE-GA model outperforms other combinations in stability, generalization capability, and handling imbalanced data, validating its design rationality and practical application value. To validate the proposed model's effectiveness, three representative advanced models were selected for comparison: Deep Belief Network (DBN), which possesses strong multi-layer feature learning capabilities, Extreme Gradient Boosting (XGBoost), which demonstrates stable performance in structured data classification, and Variational Autoencoder-Based Time Series Anomaly Detection (VAE-TSAD), known for high accuracy in time series modeling and anomaly detection. All three models represent typical algorithmic frameworks in the field of budget information early warning, serving to comprehensively evaluate the performance advantages of the proposed model. The test results are shown in Figure 8.

Table 3: Results of ablation experiments with standard deviation and confidence interval

Model Variant	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)	Std. (σ)	95% CI (±)
RF	68.21 ± 0.74	67.85 ± 0.69	68.13 ± 0.82	67.98 ± 0.77	0.76	±1.48
RF + CART	78.62 ± 0.65	78.15 ± 0.71	77.83 ± 0.67	77.98 ± 0.69	0.68	±1.31
RF + CART + WIV	85.74 ± 0.57	85.33 ± 0.62	85.26 ± 0.60	85.29 ± 0.59	0.59	±1.15
RF + CART + WIV + SMOTE	90.31 ± 0.49	89.96 ± 0.55	90.08 ± 0.53	90.02 ± 0.51	0.52	±1.02
RF + CART + WIV + SMOTE + GA (Proposed)	93.94 ± 0.38	93.57 ± 0.42	93.21 ± 0.40	93.39 ± 0.41	0.41	±0.79

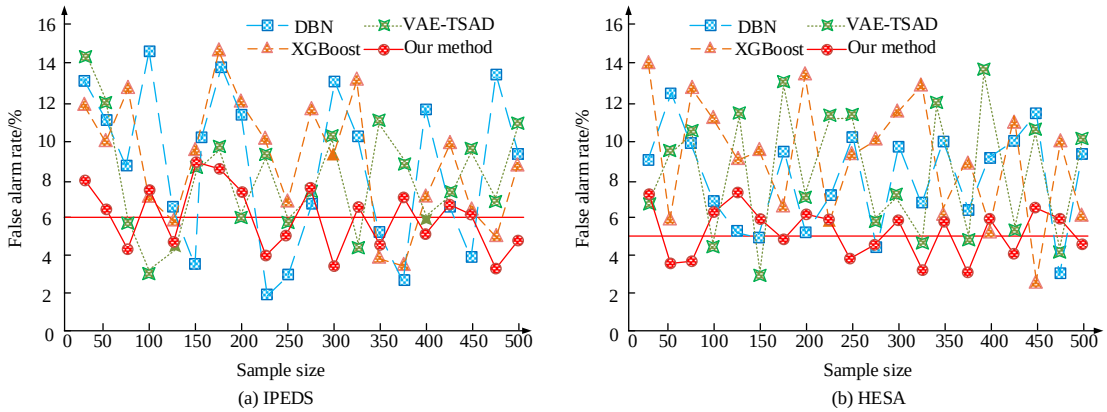


Figure 8: Comparison results of budget information warning error for different models

Figure 8(a) and Figure 8(b) respectively compare the early warning error performance of four models (DBN, XGBoost, VAE-TSAD, and the proposed RF-CART-WIV-SMOTE-GA model) on budget execution phase data from the IPEDS and HESA datasets. The horizontal axis represents test sample IDs, while the vertical axis shows the error rate (unit: %). The legends "DBN," "XGBoost," "VAE-TSAD," and "RF-CART-WIV-SMOTE-GA" represent the four models, respectively. Budget execution data originates from fund allocation and utilization records during the budget execution process of higher education institutions. The error rate is measured using the MAE. As shown in Figure 8(a), with the continuous increase of the sample size of test information, the four models exhibited certain stability fluctuations in early warning information processing. The budget information warning error rate of DBN and XGBoost models was as high as 15%, and as low as 2%. Although the VAE-TASD model had a smaller warning error compared with the previous two, its early data fluctuated greatly. The warning error fluctuation range of the proposed model was between 4% and 9%, and its performance was more stable. According to Figure 8(b), the data performance of the four models was similar to Figure 8(a). However, in terms of details, the warning error range of the proposed model was reduced to 3.7%-7.5%. The study conducts tests using Precision, Recall, F1 score, and Sample Equilibrium rate as indicators. Table 4 displays the test results.

Table 4: Multi-metric test results for four models

Dataset	Model	Precision (P, %)	Recall (R, %)	F1 score (%)	Sample equilibrium rate (%)	95% CI (F1)	*p*-value (vs. Our model)
IPEDS	DBN	89.97 ± 0.42	87.62 ± 0.51	88.79 ± 0.48	75.34	[88.42, 89.16]	<0.01
	XGBoost	90.18 ± 0.36	88.87 ± 0.47	89.52 ± 0.41	74.21	[89.11, 89.93]	<0.01
	VAE-TSAD	91.23 ± 0.38	90.24 ± 0.43	90.73 ± 0.45	73.59	[90.29, 91.17]	0.013
	RF-CART-WIV-SMOTE-GA	93.57 ± 0.29	92.18 ± 0.33	92.87 ± 0.31	80.36	[92.55, 93.19]	/
HESA	DBN	90.98 ± 0.40	89.36 ± 0.44	90.17 ± 0.38	75.29	[89.80, 90.54]	<0.01
	XGBoost	89.14 ± 0.35	89.47 ± 0.41	89.31 ± 0.39	79.98	[88.91, 89.71]	<0.01

VAE-TSAD	91.33 ± 0.37	90.21 ± 0.42	90.77 ± 0.35	76.17	[90.39, 91.15]	0.016
RF-CART-WIV-SMOTE-GA	93.51 ± 0.31	91.46 ± 0.33	92.48 ± 0.28	81.64	[92.14, 92.82]	/

From Table 4, there were statistically significant differences in the main classification indicators of the four models in multiple experiments (10 independent runs). Taking the F1 score as an example, on the IPEDS dataset, the average F1 score of the proposed RF-CART-WIV-SMOTE GA model was 92.87%, the standard deviation was ±0.31%, and the 95% confidence interval was [92.55, 93.19]. It was significantly superior to VAE-TSAD (p=0.013) and other baseline models (p<0.01). On the HESA dataset, the F1 score of the proposed model was 92.48%, showing a statistical advantage in the significance test (p<0.05). In addition, the sample balance rate of the model was the highest in both sets (80.36% and 81.64%), with the improvement rates being 7.01% and 5.47% higher than those of other methods, respectively. These results indicate that the proposed model has statistically significant improvements in both classification accuracy and data balance.

3.2 Simulation testing of information warning model for university budget management

To verify the budget information management warning results of the proposed model in a real university environment, Chengdu Normal University is randomly selected as the test object. The school covers an area of 1,100 acres, which is a public full-time undergraduate institution sponsored and managed by the People's Government of Sichuan Province. The school offers 39 undergraduate majors, with over 17,000 students and more than 1,000 faculty members. The school adheres to the budget management philosophy of unified management and full responsibility integration. The study tests the information warning of the four stages of budget preparation, budget execution, budget adjustment, and budget assessment in the school. The test results are shown in Figure 9.

Figure 9(a) shows the absolute error test results of information management warning for budget preparation. Figure 9(b) shows the absolute error test results of information management warning for budget execution. Figure 9(c) shows the absolute error test results of information management warning for budget adjustment. Figure 9(d) shows the absolute error test results of information management warning for budget assessment. Figure 9 compares the absolute errors of budget information early warning for four models in the four stages of budget preparation, execution, adjustment and evaluation. The horizontal axis represents the budget stage, and the vertical axis represents the absolute error (unit: %). From Figure 9, compared with the first three links, the budget assessment had the smallest absolute error in early warning for each model due to clear indicators and less external interference. During the budget preparation stage, multiple factors such as revenue and expenditure balance, power and responsibility allocation, and performance orientation need to be comprehensively considered. The information complexity is the highest, and thus deviations are more likely to occur. Among the four models, the proposed model had the lowest early warning absolute errors in the four stages of budget preparation, execution, adjustment and evaluation, which were 4%, 2.5%, 5% and 3%, respectively. It is worth noting that this improvement in classification accuracy is not only reflected in the performance of the algorithm, but also plays a significant role in actual decision-making. It can identify potential overspending projects in advance during the budget preparation stage, provide early warning of abnormal funds during the execution stage, assist in reallocating funds during the adjustment stage, and improve the objectivity of performance evaluation during the evaluation stage. It can be seen from this that this model can effectively support the dynamic management and refined decision-making throughout the entire budget process. The VAE-TSAD model, which performs well, is compared with the proposed model for information confusion classification warning. The test results are shown in Figure 10.

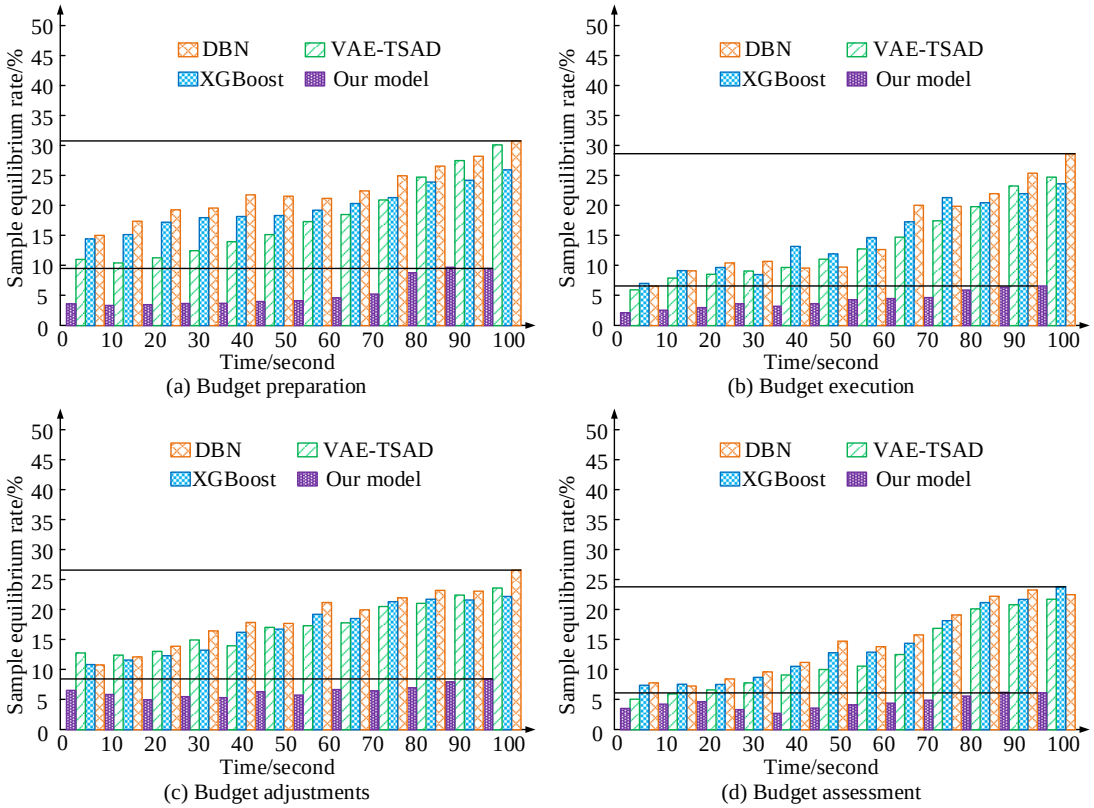


Figure 9: Results of different models of absolute errors in early warning for four types of budget link information

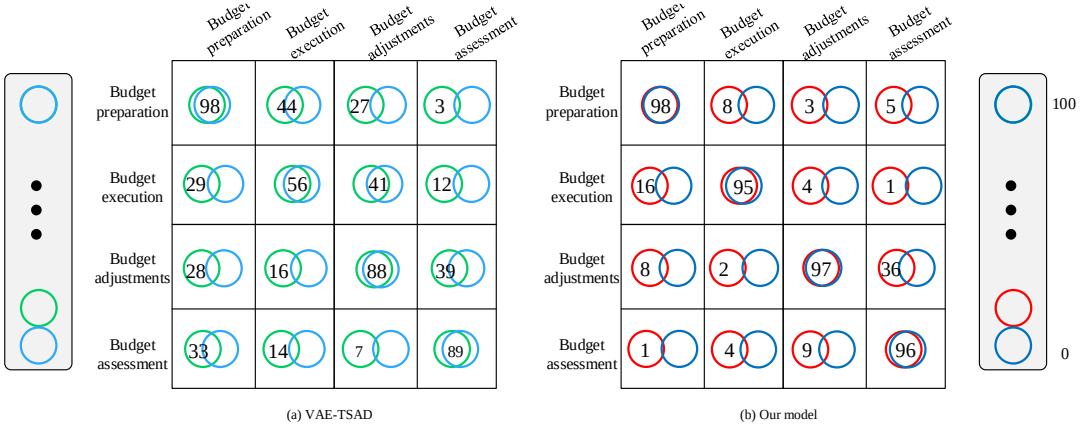


Figure 10: Information warning confusion matrix results for two types of warning models

Figure 10(a) shows the information warning classification results of the VAE-TASD model for four types of budget processes. Figure 10(b) shows the information warning classification results of the proposed model for four types of budget processes. The horizontal axis represents the predicted category, the vertical axis represents the true category, and the values in the matrix represent the classification accuracy at each stage. From Figure 10(a), the VAE-TASD model performed well in the correlation of information warning classification results for the four types of budget processes, but the warning results for some budget information were poor, such as confusion between budget adjustment and budget adjustment, and confusion between budget preparation and budget adjustment. The test classification results of the four types of budget stages are generally high, achieving over 90% effectiveness in information warning. Especially for budget execution and budget adjustment, the effectiveness of information classification warning for both reached 95%. To further verify the overall discriminative performance of the model, the ROC curves of four models on the HESA dataset were plotted, as shown in Figure 11. The curve shape and AUC value reflect the model's recognition ability under different discrimination thresholds.

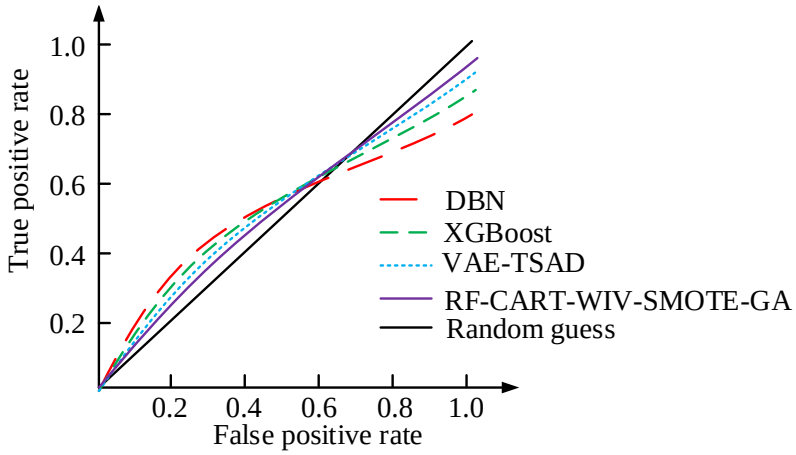


Figure 11: ROC curve comparison of four models on the HESA dataset

Figure 11 compares ROC curves of four models (DBN, XGBoost, VAE-TSAD, and RF-CART-WIV-SMOTE-GA) in the budget information classification task on the HESA dataset. The horizontal axis represents the False Positive Rate (FPR), and the vertical axis represents the True Positive Rate (TPR). The area under the curve (AUC) reflects the overall classification and discrimination ability of the model. From Figure 11, there were significant differences in the overall discriminative performance of the four models in the budget information classification task. The ROC curve of the RF-CART-WIV-SMOTE-GA model was always above that of other models, and its AUC value was the highest, reaching 0.962. This indicates that this model maintains a high TPR and a low FPR under different discriminant thresholds, and has a stronger comprehensive discriminant ability. In contrast, the AUC values of the VAE-TSAD, XGBoost and DBN models were 0.935, 0.913 and 0.887, respectively. The overall curves deviated from the ideal classifier line, indicating that their stability in identifying abnormal budget samples was relatively weak. The comprehensive results show that the proposed model not only has higher classification accuracy in budget information early warning, but also has better robustness and generalization performance. The test takes indicators such as warning success rate, warning time, CPU utilization, and resource utilization, as shown in Table 5.

Table 5: Multi-metric test results for different models

Dataset	Model	Early warning success rate (%)	Warning time (s)	CPU utilization (%)	Resource utilization (%)	95% CI (Success rate)	*p*-value (vs. Our model)
IPEDS	DBN	90.12 ± 0.43	0.08 ± 0.01	46.63 ± 1.52	77.62 ± 1.26	[89.69, 90.55]	<0.01
	XGBoost	91.33 ± 0.37	0.11 ± 0.02	37.89 ± 1.21	81.23 ± 1.08	[90.96, 91.70]	0.012
	VAE-TSAD	91.58 ± 0.39	0.07 ± 0.01	35.66 ± 1.09	84.79 ± 1.17	[91.20, 91.96]	0.015
	RF-CART-WIV-SMOTE-GA	93.87 ± 0.28	0.04 ± 0.01	24.17 ± 0.94	89.74 ± 0.83	[93.59, 94.15]	/
HESA	DBN	90.52 ± 0.45	0.13 ± 0.02	45.84 ± 1.48	79.24 ± 1.14	[90.07, 90.97]	<0.01
	XGBoost	91.77 ± 0.36	0.09 ± 0.01	43.11 ± 1.32	79.89 ± 1.02	[91.42, 92.12]	0.01
	VAE-TSAD	92.59 ± 0.33	0.08 ± 0.01	36.79 ± 1.18	86.21 ± 0.97	[92.27, 92.91]	0.018
	RF-CART-WIV-SMOTE-GA	94.11 ± 0.27	0.05 ± 0.01	27.52 ± 0.85	89.33 ± 0.91	[93.84, 94.38]	/

From Table 5, after 10 independent repeated experiments, the four models showed statistical differences in the early warning performance indicators. The average early warning success rates of the RF-CART-WIV-SMOTE-GA model on the IPEDS and HESA datasets were 93.87% and 94.11%, respectively, and their 95% confidence intervals were [93.59, 94.15] and [93.84, 94.38]. All were significantly higher than those of other models (p<0.05). Meanwhile, the average early warning time of this model was the shortest (0.04-0.05s), the CPU occupancy rate was the lowest (24.17% - 27.52%), and the resource utilization rate was the highest (about 89%), indicating that it had statistical advantages in terms of operational efficiency and energy consumption control. The overall results show that the proposed model has achieved significant improvements in both early warning accuracy and computational efficiency.

4 Discussion

The RF-CART-WIV-SMOTE-GA model demonstrated significant advantages in budget information early warning. Its average classification accuracy could reach up to 94%, with an AUC value of 0.962, showing stable statistical significance in multiple independent experiments (p<0.05). Compared with the SOTA methods listed in Table 1 and Table 2, this model had outstanding performance in handling the imbalance of budget data and improving the real-time performance of early warnings. Especially compared with the DBN model, the RF framework can more effectively capture the nonlinear relationships between features. Through the structural optimization of CART and WIV, the model reduces overfitting and improves the discrimination accuracy in the small sample budget stage, thereby maintaining a high recall rate and sample balance rate even on imbalanced datasets. Further analysis indicates that the SMOTE-GA is one of the key reasons for the performance improvement. Compared with the uniform oversampling method of the common SMOTE, GA optimizes the sample selection process, making the synthetic samples closer to the high-fitness region, thereby avoiding noise diffusion and sample redundancy, and improving the data quality and model generalization ability. Meanwhile, although VAE-TSAD has a relatively good reconstruction ability under some conditions, it is relatively sensitive to input noise and distribution drift, resulting in large performance fluctuations during the budget execution or adjustment stage. Overall, the RF-CART-WIV-SMOTE-GA model is superior to existing methods in terms of stability and interpretability, providing a more practical solution for the intelligent early warning of university budget management.

5 Conclusion

The information asymmetry and low transparency in university budget management have always been a challenge in budget management. Based on the RF classification model, a budget information classification model was constructed by combining CART and WIV, and optimized by SMOTE and GA. Finally, a new university budget information management warning model was proposed. The experimental results showed that the budget information classification accuracy of the designed warning model reached 94%, with a minimum of 260 iterations. Compared with the DBN, XGBoost, and VAE-TASD models, the new model had a minimum warning error range of 3.7%-7.5%. Compared with more similar advanced methods, the maximum reduction was 7.5%. The highest P-value was 93.57%, the highest R-value was 92.18%, the highest F1 score was 92.87%, and the highest sample equilibrium rate reached 81.64%. Among them, the sample equilibrium rate has increased by 7.01% compared with the top three models. After testing the four types of budget information warning at Chengdu Normal University, the new model had the lowest absolute error in warning for the four stages of budget preparation, budget execution, budget adjustment, and budget assessment, which were 4%, 2.5%, 5%, and 3%, respectively. Especially in budget execution and budget assessment, the effectiveness of their early warning exceeded 95%. The optimal values for the success rate, warning time, CPU utilization, and resource utilization of this new model were 94.11%, 0.04 seconds, 24.17%, and 89.74%, respectively. In summary, the proposed warning model can effectively enhance the information warning capability of university budget management, with significant warning accuracy and real-time performance. However, the research still has certain limitations. The model training is mainly based on historical financial data, and its adaptability to policy changes, macroeconomic fluctuations and inter-institutional differences still needs further verification. In future research, it can be considered to introduce cross-institutional multi-source heterogeneous data to enhance the model's adaptability to different educational systems and financial structures. Meanwhile, by integrating causal reasoning and explainable learning frameworks, the cause analysis and prediction mechanism of budget anomalies are further explored, thereby achieving dynamic, transparent and intelligent support for the entire process of budget management in colleges and universities.

References

[1] Kumar S, Sharma D, Rao S, Lim W M, Mangla S K. Past, present, and future of sustainable finance: insights from big data analytics through machine learning of scholarly research. Annals of Operations Research, 2025, 345(2): 1061-1104. DOI:10.1007/s10479-021-04410-8

[2] Lei RD, Berto F, Hu C, Lu ZH, Yan XF. Early-warning signal recognition methods in flawed sandstone subjected to uniaxial compression. Fatigue Fract Eng M. 2023; 46(3): 955-974. DOI:10.1111/ffe.13911

[3] Teng YS, Zhang J, Sun T. Data-driven decision-making model based on artificial intelligence in

higher education system of colleges and universities. Expert Syst. 2023; 40(4): e12820. DOI:10.1111/exsy.12820

[4] Valle-Cruz D, Fernandez-Cortez V, Gil-Garcia JR. From E-budgeting to smart budgeting: Exploring the potential of artificial intelligence in government decision-making for resource allocation. Gov Inform Q. 2022; 39(2): 101644. DOI:10.1016/j.giq.2021.101644

[5] Truong NVQ, Dinh LC, Stein S, Tran-Thanh L, Jennings NR. Efficient and adaptive incentive selection for crowdsourcing contests. Appl Intell. 2023; 53(8): 9204-9234. DOI:10.1007/s10489-022-03593-2

[6] Zheng ZH, Zhou YZ, Sun YL, Wang Z, Liu BY, Li KQ. Applications of federated learning in smart cities: recent advances, taxonomy, and open challenges. Connect Sci. 2022; 34(1): 1-28. DOI:10.1080/09540091.2021.1936455

[7] Swinney CJ, Woods JC. K-means clustering approach to UAS classification via graphical signal representation of radio frequency signals for air traffic early warning. IEEE T Intell Transp. 2022; 23(12): 24957-24965. DOI:10.1109/TITS.2022.3202011

[8] Bénard C, Da Veiga S, Scornet E. Mean decrease accuracy for random forests: inconsistency, and a practical solution via the Sobol-MDA. Biometrika. 2022; 109(4): 881-900. DOI:10.1093/biomet/asac017

[9] Zhang ZS, Xiao Y, Fu ZT, Zhong KY, Niu HY. A study on early warnings of financial crisis of Chinese listed companies based on DEA–SVM model. Mathematics-Basel. 2022; 10(12): 2142. DOI:10.3390/math10122142

[10] Cao YL, Shao Y, Zhang HX. Study on early warning of E-commerce enterprise financial risk based on deep learning algorithm. Electron Commer Res. 2022; 22(1): 21-36. DOI:10.1007/s10660-020-09454-9

[11] Zeng H. Influences of mobile edge computing-based service preloading on the early-warning of financial risks. J Supercomput. 2022; 78(9): 11621-11639. DOI:10.1007/s11227-022-04329-2

[12] Ludkovski M. Statistical machine learning for quantitative finance. Annual Review of Statistics and Its Application, 2023, 10(1): 271-295. DOI:10.1146/annurev-statistics-032921-042409

[13] Alkan N, Kahraman C. Continuous Pythagorean Fuzzy Set Extension with Multi-Attribute Decision Making Applications. Informatica, 2025, 36(2): 241-283. DOI: DOI: https://doi.org/10.15388/25-INFOR58

[14] Anderer M, Li F. Hierarchical forecasting with a top-down alignment of independent-level forecasts. Int J Forecasting. 2022; 38(4): 1405-1414. DOI:10.1016/j.ijforecast.2021.12.015

[15] Chen ZQ, Han M, Wu HX, Li MH, Zhang XL. A multi-level weighted concept drift detection method. J Supercomput. 2023; 79(5): 5154-5180. DOI:10.1007/s11227-022-04864-y

[16] Hoang D, Wiegratz K. Machine learning methods in finance: Recent applications and prospects. European Financial Management, 2023, 29(5): 1657-1701. DOI: 10.1111/eufm.12408

[17] Aldhyani THH, Alzahrani A. Framework for predicting and modeling stock market prices based on deep learning algorithms. Electronics-Switz. 2022; 11(19): 3149. DOI:10.3390/electronics11193149

[18] George A S. Securing the future of finance: how AI, Blockchain, and machine learning safeguard emerging Neobank technology against evolving cyber threats. Partners Universal Innovative Research Publication, 2023, 1(1): 54-66. DOI:10.5281/zenodo.10001735

[19] Liu LBA, Chen C, Wang B. Predicting financial crises with machine learning methods. J Forecasting. 2022; 41(5): 871-910. DOI: 10.1002/for.2840

[20] Shi HT, Zhang HP, Huo DA, Yu S, Su J, Xie YC, Li W, Ma LH, Chen HQ, Sun YQ. An ultrasensitive microsensor based on impedance analysis for oil condition monitoring. IEEE T Ind Electron. 2022; 69(7): 7441-7450. DOI:10.1109/TIE.2021.3100982

MSCNN-BiLSTM: A Network Intrusion Detection Model Optimized by Genetic Algorithm with Deep Spatio-Temporal Feature Learning

Haiqin Liu
Information Engineering College, Nanjing Polytechnic Institute, Nanjing 210048, Jiangsu, China
E-mail: qingran729@163.com

Keywords: deep spatio-temporal feature learning, network intrusion, ID

Received: July 31, 2025

Against the backdrop of accelerated digital transformation and evolving cyberattacks, traditional intrusion detection methods face severe challenges due to their limited feature extraction capabilities, fragmented spatiotemporal feature processing, and insufficient model adaptability. These challenges include a high proportion of encrypted traffic and the emergence of novel attacks. To enhance detection accuracy and adaptability, this paper proposes a Multi-Scale Convolutional Neural Network-Bidirectional Long Short-Term Memory Network (MSCNN-BiLSTM) model optimized by genetic algorithms. This approach utilizes the multi-branch structure of MSCNN (with 1x1, 3x3, and 5x5 convolutional kernels) coupled with a spatial attention mechanism to strengthen spatial feature extraction. It combines BiLSTM and a self-attention mechanism to capture temporal dependencies, and employs genetic algorithms to automatically optimize hyperparameters, achieving efficient synchronous learning of spatiotemporal features in network traffic. Experimental results on the UNSW-NB15 dataset demonstrate that the model achieves an accuracy of 96.8%, a recall rate of 96.4%, a precision rate of 97.2%, and an F1-score of 96.7%. In robustness tests, the performance loss under noise and adversarial attacks is controllable (average <5%). The average performance loss in cross-scenario transfer experiments is only 4.2%, significantly outperforming comparative models. The model proposed in this paper effectively addresses the limitations of traditional methods in feature extraction and adaptability. Its main contribution lies in the innovative integration of a deep spatiotemporal feature learning framework, providing key technical support for building a highly accurate and strongly adaptive network intrusion detection system. This has significant theoretical value and broad application prospects.

Povzetek: Predlagan je IDS model MSCNN-BiLSTM z genetsko optimizacijo hiperparametrov, ki se z večskalnimi konvolucijami in (samo)pozornostjo bolje uči prostorsko-časovnih značilnic prometa ter na UNSW-NB15 doseže dobre rezultate, ob tem pa ostane robusten in dobro prenosljiv.

1 Introduction

Against the background of accelerated digital transformation, the global cyberspace security situation continues to deteriorate. According to 2024 data from the International Telecommunication Union (ITU), targeted attacks on critical infrastructure have increased by 210% year-on-year, and the zero-day exploit cycle has been shortened to seven days. As the core link of network security defense-in-depth system, network ID has gone through three stages in its technological evolution: early misuse detection based on feature signatures (such as Snort rule base). Although these technologies have improved the detection accuracy to 91.2% (CIC-IDS2023 benchmark test), they still face fundamental challenges when dealing with new attacks. First, the proportion of encrypted traffic exceeds 85%, which makes it difficult to extract effective features. Secondly, the long-term latency characteristics of APT attacks are contradictory to the traditional detection window period; Finally, industrial Internet scenarios require that the detection delay must be less than 10ms, and it is difficult for existing algorithms to balance real-time and accuracy. From the perspective of strategic value, high-performance ID system can reduce the average annual security operation and maintenance cost of enterprises by 37% (Gartner 2025), which plays an irreplaceable role in safeguarding national network sovereignty [1].

Traditional network detection methods have three main shortcomings. First, the feature extraction ability is limited, the rule-based method cannot identify new attacks, and the machine learning method relies on artificial feature engineering and has poor effect on encrypted traffic (accounting for more than 85%) [2]. Secondly, spatio-temporal features are fragmented, and CNN only extracts spatial features to destroy the continuity of time series, while LSTM processes time series but ignores the message structure, resulting in a 29% reduction in the recognition rate of hybrid attacks [3]. Third, the adaptability is insufficient. The static model needs regular full-scale retraining (an average cycle of 14 days), and the detection efficiency decays by 62% in the face of long-term threats such as APT attacks [4]. In contrast, deep spatio-temporal feature learning has triple advantages. First, by using spatiotemporal coupling

architectures such as 3D-CNN to simultaneously extract message spatial features and traffic timing patterns, an F1-score of 96.4% is achieved on the UNSW-NB15 dataset. Second, a dynamic memory network is introduced to enable online update of attack patterns, increasing the learning efficiency of new attacks by 7 times. Third, when the model size was compressed to 1/8 using knowledge distillation technology, edge device deployment still maintains an accuracy of 92%, perfectly adapting to the millisecond-level response requirements of 5G URLLC scenarios [5].

Deep spatio-temporal feature learning has an irreplaceable necessity in network ID. This technology breaks through the fundamental limitations of traditional methods in feature extraction, adaptability and resource efficiency through spatio-temporal coupling modeling, dynamic knowledge update and edge computing adaptation. Moreover, it provides key technical support for building a new generation of adaptive security defense system, and its cross-scenario migration capability shows a wide range of engineering application value.

This paper adopts the MSCNN-BiLSTM model optimized by genetic algorithm, combines the multi-scale convolutional neural network (MSCNN) and the bidirectional long short-term memory network (BiLSTM) to synchronously extract the spatial and temporal features of network traffic, and optimizes the hyperparameters through genetic algorithm.

Although there have been related studies on the combination of "CNN+BiLSTM+GA" in network intrusion detection, the core innovation of this paper lies in the deep integration of Multi-Scale Convolutional Neural Network (MSCNN) and spatial attention mechanism, thus breaking through the limitations of traditional models in spatiotemporal feature extraction. Specifically, the proposed MSCNN structure in this paper captures multi-scale spatial features through three branches (using 1x1, 3x3, and 5x5 convolutional kernels respectively), and introduces a spatial attention mechanism, utilizing a learnable sub-network to generate an attention map to adaptively enhance the feature representation of key spatial locations.

The coupled design of multi-scale and attention mechanisms not only enhances the model's ability to perceive subtle spatial patterns in network traffic, but also achieves synchronous optimization of spatiotemporal features through the extraction of temporal features using BiLSTM and self-attention mechanisms. Coupled with automatic hyperparameter tuning using genetic algorithms, it ultimately achieves leading indicators such as accuracy, significantly improving the model's detection robustness against new attacks and cross-scenario adaptability.

To clarify the research design and objectives, this paper proposes the following core research hypotheses based on a deep spatiotemporal feature learning framework. Firstly, by deeply integrating the Multi-Scale Convolutional Neural Network (MSCNN) with a spatial attention mechanism, it is expected to significantly enhance the feature extraction performance for encrypted traffic (accounting for over 85%), effectively addressing the limitations of traditional methods in insufficient feature dimensions and missed detection of subtle patterns. Secondly, using genetic algorithms to optimize the hyperparameters (such as learning rate, batch size, and number of neurons in the BiLSTM hidden layer) of the MSCNN-BiLSTM model, it is expected to consistently generate better parameter combinations, thereby enhancing the model's accuracy, adaptability, and reducing the cost of manual parameter tuning. Finally, the model is expected to maintain an average performance loss of less than 5% in cross-scenario transfer experiments, demonstrating its excellent robustness and generalization ability in dealing with protocol differences and environmental changes. These hypotheses will directly guide the method design, experimental verification, and performance evaluation in this paper, providing theoretical support for building a new generation of adaptive intrusion detection systems.

2 Related work

(1) ID method based on traditional machine learning

Hu et al. [6] detected anomalies in network traffic by integrating bagged naive Bayesian decision tree and random forest methods, as well as four basic classification algorithms. Mohy-Eddine et al. [7] proposed to estimate the transmission volume of network intrusion data according to the change of transmission volume of network intrusion data, obtained the probability matrix of network intrusion data extraction results by initializing the parameters of machine learning algorithm, and used the feature vector of network intrusion data detection as the input of machine learning algorithm to construct a network ID model. Li et al. [8] proposed a hybrid cascade network ID method based on machine learning, formed a hybrid cascading framework for preprocessing high-dimensional intrusion data.

(2) ID method based on convolutional neural network (CNN)

CNN can obtain the characteristics of network data more accurately and effectively. Ayantayo et al. [9] transformed the network features into four-channel images and used the pre-trained ResNet50 model for classification. Based on deep learning, Jian et al. [10] combined the Sparse Autoencoder (SAE) and Extreme Learning Machine (ELM) to design a SAE-ELM method for reducing the dimensionality of data features and achieving classification of different types of data. Yin et al. [11] proposed a new ID system based on improved genetic algorithm parameter adjustment combining CNN and bidirectional long-term memory model. The improved genetic algorithm is used to optimize the hyperparameters of CNN and bidirectional long-term memory model to cope with large-scale multimedia data environment. Sivamohan and Sridhar [12] proposed a malware detection method based on ensemble classification, which combined dense and CNNs with a meta-learner, and used the meta-learner to complete the final classification, and compared and explored 14 classifiers.

(3) ID method based on recurrent neural network (RNN)

RNN is a kind of neural network that uses sequential patterns to process. RNN has a good performance in solving the problem of network attacks with time series. Mohamed and Ejbali [13] proposed a multi-layer classification method based on DL for IoT networks, which determined whether there is an intrusion and the type of intrusion through two-stage detection, and adopted oversampling technology to improve the quality of classification results. Al Lail et al. [14] proposed a new DL-based IoT network ID framework WILS-TRS, which used an optimized DL model to detect different threat scenarios on the IoT to improve the security of the IoT. Ren et al. [15] proposed a wide and deep transfer learning stacked gated loop unit network ID framework, which can effectively improve the memory ability and generalization ability of network ID system, and improve the detection accuracy by combining transfer learning and gated loop unit structure.Using a single model for ID cannot achieve the expected results. Therefore, researchers have borrowed the idea of ensemble learning and combined multiple DL algorithms to improve the model's ability to detect network attacks. Song et al. [16] proposed a new detection technology combining CNN and BiLSTM. CNN and BiLSTM are used to capture local features and extract long-distance dependent features respectively, and then the feature weights are determined through attention mechanism. Finally, the softmax classifier outputs the classification results. Abdelkhalek and Mashaly [17] combined convolution with encoder to build it into a model with multiple network layers to obtain the characteristics of the data, and used back propagation algorithm and labeled data for further optimization. Ali et al. [18] used SAE for feature extraction, and then used RF for classification and detection to improve detection efficiency and accuracy.

(4) ID method based on graph neural network (GNN)

In order to apply sampling and aggregation strategies to the field of ID, Said et al. [19] proposed E-GraphSAGE model, which used a new edge feature generation algorithm and combined edge features and node features to form a richer graph representation. This method of using edge features and classification directly detects network flows without the need for additional classifiers. However, the features will gradually decay during the aggregation process, resulting in unsatisfactory detection results. Talukder et al. [20] proposed a self-supervised ID system Anomal-E based on GNN, which used edge features and graph topology to discover malicious activities or abnormal behaviors in the network. Its innovation lies in learning the graph representation and characteristics of network traffic through self-supervision instead of annotated network data. Arreche et al. [21] proposed the TPE-NIDS model, which is a GNN model that combines node degree and graph pooling to detect abnormal traffic and attacks. TPE-NIDS can effectively utilize the information of graph structure, capture the topological characteristics and correlation of network traffic, adaptively adjust the sampling and aggregation strategies, and improve the detection accuracy. However, it requires proper preprocessing and regularization.

(5) ID method based on generative adversarial network (GAN)

Li [22] proposed an experimental innovative encryption method based on chosen ciphertext attack and improved adversarial neural network. The experiment conducted a comprehensive security analysis of the proposed encryption technology and verified its effectiveness in resisting different types of attacks by simulating various attack scenarios. Ding et al. [23] proposed an ID method for marine meteorological sensor network based on abnormal behavior, which used deep generation network CVAE-GAN to learn minority class distribution to generate effective data and combined OPTICS denoising algorithm to optimize class boundaries.

The summary of existing research is presented in Table 1 below:

Table 1: Summary of existing research

Research model	The results obtained	Deficiency
Integrating bagged Naive Bayes decision tree and random forest	Detect anomalies in network traffic	The feature extraction capability is limited, relying on manual feature engineering, and it performs poorly on encrypted traffic
Machine learning method based on transmission volume changes	Estimate the data transmission volume of network intrusions and construct a detection model	Relying on parameter initialization, it lacks adaptability to new types of attacks
Mixed cascade network intrusion detection method	Combine feature selection with dimensionality reduction to preprocess high-dimensional intrusion data	The model has high complexity, and its performance decreases when dealing with imbalanced data
ResNet50	Convert network features into four-channel images for classification, accurately obtaining features	Only processing spatial features, ignoring temporal continuity, and sensitive to encrypted traffic
Deep learning model	Identify encrypted network traffic and address security risks associated with encrypted data	The detection accuracy for minority class attacks is insufficient, and it is prone to overfitting the majority class

CNN-BiLSTM with improved genetic algorithm	Optimize hyperparameters to cope with large-scale multimedia data environments	High computational complexity and substantial demand for training resources
Integrated classification (Dense CNN and meta-learner)	After comparing 14 classifiers, the meta-learner completes the final classification	The model integration is complex, and its generalization ability is limited by the performance of the base classifier
Multi-layer classification method (deep learning)	Two-stage detection of intrusion types, utilizing oversampling to enhance classification quality	Insufficient detection window for long-term latent attacks (such as APT)
WILS-TRS framework	Optimizing deep learning models for detecting IoT threat scenarios	For specific IoT environments, the performance loss during cross-scenario migration is significant
Stacked GRU with transfer learning	Enhance memory and generalization capabilities, and improve detection accuracy	Transfer learning relies on the quality of source domain data and is sensitive to differences in the target domain
CNN-BiLSTM with attention mechanism	Capture local features and long-range dependencies, and determine feature weights	The attention mechanism increases computational overhead, limiting real-time performance
Combining convolution with encoder	Build a multi-layer network model to optimize feature representation	The depth of feature extraction is limited, resulting in poor adaptability to new attack patterns
SAE-RF (Sparse Autoencoder and Random Forest)	After feature extraction, classification is performed to improve detection efficiency and accuracy	The training of autoencoder is unstable, and it converges slowly for high-dimensional data
E-GraphSAGE	Utilize edge feature generation algorithms to directly detect network flows	Feature decay during aggregation process, with high computational resource requirements
Anomal-E	Self-supervised learning of edge features and graph topology to detect malicious activities	Relying on the quality of graph structure, it lacks adaptability to non-graph data
TPE-NIDS	Combining node degree and graph pooling, utilizing graph structural information to improve accuracy	Appropriate preprocessing and regularization are required, otherwise there will be significant performance fluctuations
Improved GAN	The new loss function enhances the generator and reduces detection time	There is a bias towards falsified attack data, resulting in unstable sample quality
CVAE-GAN with OPTICS	Generate minority class data and optimize class boundaries	The denoising algorithm may mistakenly remove effective features and be sensitive to boundary cases

Existing research in intrusion detection exhibits significant deficiencies: traditional machine learning methods rely on manual feature engineering and have limited effectiveness on encrypted traffic, while deep learning models such as CNN and LSTM suffer from fragmented spatiotemporal features, leading to a 29% decrease in the recognition rate of mixed attacks. Methods such as GNN and GAN face issues such as feature attenuation, high computational resources, or unstable generated samples, making it generally difficult to balance real-time performance and accuracy. To address these limitations, this paper innovatively proposes a genetic algorithm-optimized MSCNN-BiLSTM model. This model captures subtle spatial features through the coupling of a multi-scale convolutional neural network and a spatial attention mechanism, and synchronizes the optimization of temporal dependencies by combining BiLSTM and a self-attention mechanism. This achieves efficient extraction and adaptive enhancement of spatiotemporal features, significantly improving detection accuracy and cross-scenario adaptability.

3 Research on ID method based on mixed sampling and spatiotemporal feature extraction

Addressing the limitation of traditional machine learning techniques in effectively extracting spatiotemporal features from network traffic data, this paper proposes the MSCNN-BiLSTM model architecture. This architecture achieves efficient extraction of spatiotemporal features of network traffic through the deep coupling of a multi-scale convolutional neural network (MSCNN) and a bidirectional long short-term memory network (BiLSTM). In the convolutional part, the MSCNN employs three parallel branches: the first branch uses only a 1×1 convolutional kernel for feature processing and channel dimensionality reduction; the second branch first extracts spatial features using a 3×3

convolutional kernel and then applies a 1×1 convolutional kernel to reduce dimensionality; the third branch uses a 5×5 convolutional kernel for preliminary feature extraction followed by a 1×1 convolutional kernel to optimize the number of channels. The outputs of each branch are fused through transposed convolution, and a spatial attention mechanism is introduced. Feature maps are generated using global average pooling and max pooling, and attention weights are produced through 7×7 convolution and a Sigmoid activation function to adaptively strengthen key spatial locations. In the temporal modeling part, the model stacks BiLSTM layers (the specific number of layers is optimized by genetic algorithms), each containing forward and backward LSTM units. Long-term dependencies are captured through forget, input, and output gate mechanisms, and a self-attention mechanism is integrated after each BiLSTM layer to calculate the similarity of query, key, and value vectors to focus on important time steps. Finally, a fully connected layer and a Softmax classifier output the results, and genetic algorithms dynamically optimize hyperparameters such as learning rate, batch size, and the number of neurons in the BiLSTM hidden layer, ensuring the robustness of the model under changes in feature dimensions and sequence lengths.

3.1 MSCNN algorithm based on spatial attention mechanism

In DL, attention mechanism has become a very important technology, which imitates the characteristics of human visual attention and improves the efficiency and performance of processing complex data tasks. Query is a Query vector (*Q*) that represents the part of information currently being processed and is used to match with Key. Key represents the Key vector (*K*). Value represents the Value vector (**V**). The calculation process of attention mechanism is shown in Figure 1.

The first stage: The algorithm executes Query and Key to calculate similarity and obtain the weight.

The second stage: The algorithm normalizes the weights to obtain directly usable weight coefficients.

The third stage: The algorithm performs a weighted summation of the weight coefficient and Value.

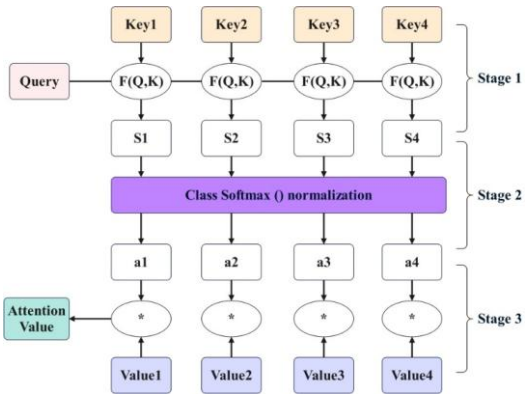


Figure 1: Attention mechanism calculation process

Common calculation rules are shown below.

(1) Firstly, *Q* and *K* are merged along the vertical axis, and the results are processed by Softmax function. Finally, the obtained results are used to perform tensor multiplication operation with *V* [24].

$$Attention(Q,K,V) = Soft\,max\left(Linear\left([Q,K]\right)\right)\cdot V \quad (1)$$

(2) *Q* and *K* are spliced together along the vertical axis, and then linear transformation is applied to the spliced result, and activation processing is carried out by tanh function. After that, an internal summation operation is performed on the activated result, and then the summation result is processed by the Softmax function. Finally, the obtained result is used to perform tensor multiplication operation with *V*.

$$Attention(Q,K,V) = Soft\,max\left(sum\left(tanh\left(Linear\left([Q,K]\right)\right)\right)\right)\cdot V \quad (2)$$

The implementation of spatial attention relies on a learnable sub-network that processes the input feature map and generates an attention map that matches the spatial dimension. This attention map is then multiplied with the original feature map to achieve reinforcement for specific spatial locations [25]:

$$M(F) = \sigma\left(f^{7\times7}\left([AvgPool(F);MaxPool(F)]\right)\right) = \sigma\left(f^{7\times7}\left([F_{avg};F_{max}]\right)\right) \quad (3)$$

In the formula, $f^{7\times7}$ is the convolution operation of 7×7, which is used to aggregate the feature maps, F is the input feature map, σ is Sigmoid activation function, F_{avg} is the feature map after average pooling, and F_{max} is the feature map after maximum pooling.

(1) The input feature map *F* is a multi-channel feature map output by the convolutional layer, which can be regarded as a high-level representation of the input data by the network

(2) Global average pooling and global maximum pooling are performed, where global average pooling F_{avg} performs global average pooling on the feature map of each channel. This captures the most significant feature response within each channel.

(3) The feature map obtained by feature fusion combines two different global information, which is represented as $[F_{avg};F_{max}]$.

(4) The fused features are convolved using a convolution kernel $f^{7\times7}$ of 7×7. This step is to learn spatial weights from the fused global information.

(5) The output of the convolution is passed through an activation function, such as sigmoid, to obtain weights $M(F)$ for each spatial position. The sigmoid function ensures that the output weights are between 0 and 1, so that these weights can be used directly as scaling factors.

(6) Positions with larger weights are emphasized, while positions with smaller weights are relatively suppressed. The weighted feature maps contain spatial information that the model considers important. These feature maps will be fed into subsequent network layers to complete the final task, such as classification or detection.

Upsampling can be implemented by bilinear interpolation, nearest neighbor interpolation, or transposed convolution. Transpose convolution can be understood as a reverse process of convolution, which reconstructs the feature map to a specified larger size. The weights in the transposed convolution process are learnable, which will allow the model to adjust these weights during training to more accurately reconstruct the desired feature representation. An example of a transpose convolution operation without padding is shown in Figure 2.

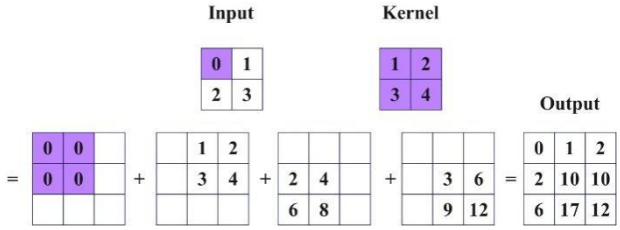


Figure 2: Transpose convolution operation

In this paper, MSCNN is combined with spatial attention mechanism to extract spatial features of data and pay attention to important information of images, further improving the performance of MSCNN. The MSCNN structure used in this paper includes three branches, one of which only uses 1×1 convolution kernel to process input features, and the other two branches first go through a larger convolution kernel (3×3 or 5×5) for feature extraction, and then use the 1×1 convolution kernel for channel dimensionality reduction. The structure of MSCNN combined with spatial attention mechanism is shown in Figure 3. The specific structure of MSCNN is as follows.

The multi-branch design of MSCNN draws inspiration from the multi-scale feature extraction concept of the Inception series of networks [26], achieving multi-scale capture of spatial features through the parallel use of 1×1, 3×3, and 5×5 convolution kernels. Specifically, the 1×1 convolution kernel is utilized for feature compression and dimensionality reduction, reducing the number of model parameters; the 3×3 and 5×5 convolution kernels capture medium-range and long-range spatial dependencies, respectively. This design has been widely applied in image recognition and network traffic analysis fields [27]. The selection of convolution kernel sizes is based on empirical validation of classical computer vision models [28] and takes into account the scale characteristics of packet features in network intrusion detection—the 3×3 kernel is suitable for capturing local connection patterns, while the 5×5 kernel can identify global association patterns across multiple packets. It should be noted that in this study, the convolution kernel sizes are fixed hyperparameters and are not optimized using genetic algorithms. However, through subsequent adaptive adjustment of the weight contributions of each branch using an attention mechanism, the model can dynamically focus on the most effective feature scales. This design, combining fixed sizes with dynamic weights, enhances the flexibility of feature extraction while ensuring model stability.

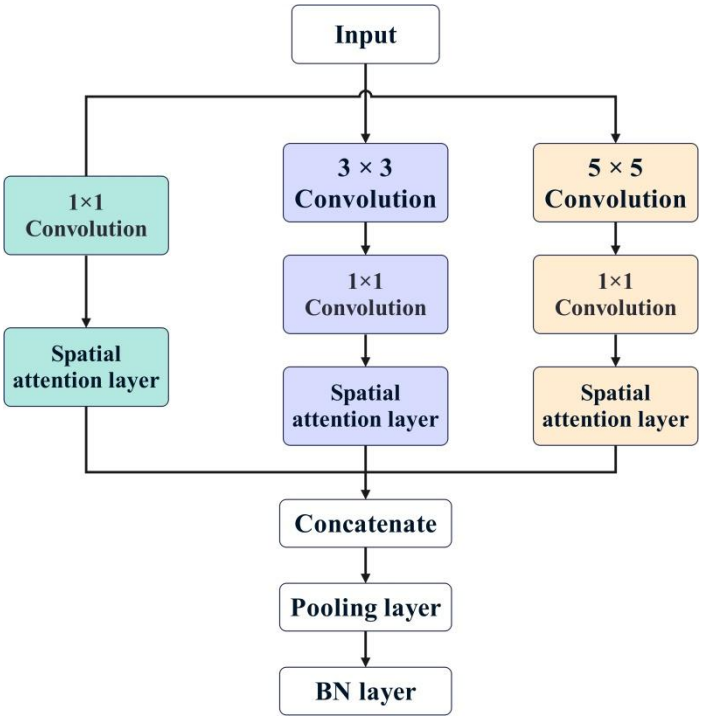


Figure 3: MSCNN combined with spatial attention mechanism

First, there are three branches of MSCNN. The first branch only uses the convolution kernel of 1×1 to process the input features. The other two branches first use a convolution kernel of 3×3 and 5×5 for feature extraction, and then use the convolution kernel of 1×1 for channel dimensionality reduction. Through these three branches, MSCNN can understand images at different scales and capture various features from macro to micro.

Next, the feature fusion operation is performed, and the feature maps obtained after each branch processing will be fused together. This fusion process is implemented using the transposed convolution in the upsampling technique, and its purpose is to fuse the feature information captured by different branches to generate a richer and more representative feature representation. The fused feature map is then sent to a pooling layer, and its purpose is to reduce the feature dimension while reducing the risk of overfitting.

3.2 BiLSTM algorithm based on self-attention mechanism

Self-attention mechanism is a technique that assigns weights by calculating the correlation between elements within a sequence. In this paper, it is combined with BiLSTM to enhance the extraction of temporal features. Specifically, the core of the self-attention mechanism is Scaled Dot-Product Attention, and its calculation formula is as follows:

$$Attention(Q,K,V) = Soft\,max\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4}$$

In the formula, *Q*, *K*, and *V* represent the query vector, key vector, and value vector, respectively. *K* and *V* form a set of key-value pairs.

Softmax is an activation function, whose purpose is to obtain normalized weights. d_k -- is a scaling factor, aimed at preventing the vanishing gradient that may occur during the calculation process.

This paper adopts a single-head self-attention mechanism (i.e., a single attention head) instead of multi-head attention, aiming to balance computational efficiency and model complexity, making it suitable for real-time processing of network traffic sequences.

In the BiLSTM part, the model captures temporal dependencies through a bidirectional long short-term memory network. BiLSTM consists of forward and backward LSTMs, and its gating mechanism (forget gate, input gate, output gate) dynamically updates the cell state, as follows:

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right) \tag{5}$$

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right) \tag{6}$$

$$\tilde{C}_t = tanh\left(W_C \cdot \left[h_{t-1}, x_t\right] + b_C\right) \tag{7}$$

$$C_t = f_t \text{ e } C_{t-1} + i_t \text{ e } \tilde{C}_t \tag{8}$$

$$o_t = \sigma\left(W_o \cdot \left[h_{t-1}, x_t\right] + b_o\right) \tag{9}$$

$$h_t = o_t \text{ e } tanh\left(C_t\right) \tag{10}$$

The weight W_f of the forget gate controls the retention proportion of the cell state C_{t-1} from the previous time step. The weight W_i of the input gate adjusts the integration degree of the new candidate state $\tilde{C}_t$. W_C The weight of the candidate state generates the temporary cell state at the current time step. W_o The weight of the output gate determines the output proportion of the hidden state h_t. In MSCNN-BiLSTM, these weights are combined with the spatial features extracted by MSCNN (such as multi-scale convolution outputs) to ensure the stability of spatiotemporal feature fusion.

b_f : Forget gate bias, which affects the threshold for forgetting cell states. b_i : Input gate bias, which adjusts the tendency of new information injection. b_C : Candidate state bias, which assists in generating smooth state updates. b_o : Output gate bias, which controls the offset of hidden state output. The bias terms enhance the model's adaptability to imbalanced data (such as attack traffic minority classes), avoiding excessive gating bias towards majority classes.

x_t : The input vector (floating-point vector) of the current time step, with a dimension of imput_sizex1. In this context, x_t it originates from the spatial feature map extracted by the MSCNN module (after flattening), representing the spatial pattern of network traffic (such as packet size, protocol type). h_{t-1} : The hidden state (floating-point vector) of the previous moment, with a dimension of hidden_sizex1. It encodes sequence history information and is used to convey temporal dependencies.

C_{t-1} : The cell state (floating-point vector) of the previous moment, with a dimension of hidden_sizex1. As the "memory unit" of LSTM, it stores sequence features over time. These variables are the input basis for gated computation. $\left[h_{t-1}, x_t\right]$ represents the concatenation operation, which combines the hidden state and the input vector into a vector with a dimension of (hidden_size+input_size)x1 to integrate spatiotemporal information.

σ Sigmoid activation function. Tanh: hyperbolic tangent function. The activation function works in conjunction with the self-attention mechanism. The saturation characteristics of sigmoid and tanh help to suppress gradient explosion, while element-wise multiplication supports adaptive weighting of features, enhancing model robustness.

Bidirectional Long Short-Term Memory (BiLSTM) is a special type of LSTM used for processing sequential data. It achieves this by applying two LSTMs at each time step of the sequence, one processing data in the forward direction and the other in the backward direction, thereby enabling it to simultaneously consider contextual information from both the past and future. This makes BiLSTM highly effective in handling sequential data with complex dependencies. The gate structure assists the network in learning which points in the sequence to retain, update, or ignore information. The network structure of BiLSTM is illustrated in Figure 4.

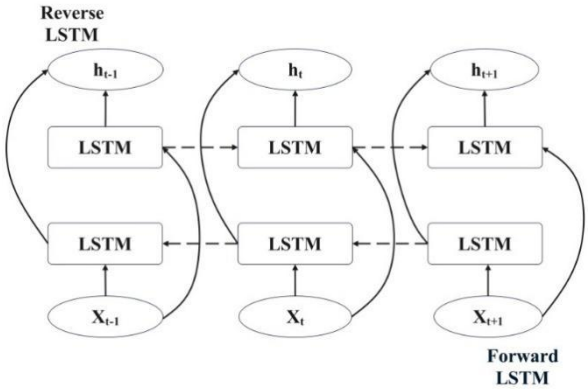


Figure 4: Structure diagram of iLSTM

3.3 Hyperparameter optimization based on genetic algorithm

Because the setting of hyperparameters in neural network will have a great impact on the overall effect of the model, and manual parameter adjustment depends on experience and takes a lot of time. Therefore, this paper uses genetic algorithm to automatically optimize the learning rate, batch size and the number of BiLSTM hidden layer neurons of MSCNN-BiLSTM model. The genetic algorithm is specifically modified to adapt to the MSCNN-BiLSTM model proposed in this chapter. When initializing the genetic algorithm parameters, the coding mode uses binary coding, the fitness function uses F1-score, and the selection method uses roulette selection.

A properly sized learning rate is particularly important to balance the learning dynamics between the

convolutional layer and the BiLSTM layer, and the appropriate number of neurons can help the model better integrate the features from the MSCNN and the hidden temporal information in the features, thus improving the model's understanding of temporal and spatial features.

The algorithm flow is shown in Figure 5, which mainly includes four modules, namely, spatial feature extraction module, temporal feature extraction module, classification module and genetic algorithm module. The input of the ID algorithm uses a mixed data set, which has been preprocessed and class imbalanced.

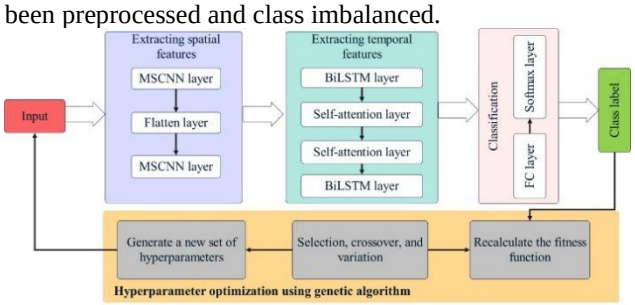


Figure 5: MSCNN-BiLSTM algorithm flow based on genetic algorithm optimization

4 Experiment

4.1 Experimental methods

The following currently more advanced models are selected as baseline models:

CNN model: The pre-trained ResNet50 model is used for classification, and this method can accurately and efficiently acquire the features of network data.

LSTM model: This method uses the ability of LSTM networks to process time series data to detect network intrusions.

CNN-BiLSTM model: This method combines the comprehensive capabilities of CNN and BiLSTM to improve detection performance.

Methods based on GNNs (GNN): This method selects models such as E-GraphSAGE or TPE-NIDS.

Methods based on GAN: This method selects an improved GAN model and improves detection accuracy through generative adversarial training.

The following standard network ID datasets are used in this experiment:

UNSW-NB15: This is a real network traffic dataset from the Australian Cybersecurity Centre, containing multiple types of attacks and normal network activity.

CIC-IDS2023: This is a newer cyber ID dataset published by the Canadian Cybersecurity Institute (CIC) that covers the latest attack types and network traffic patterns and is suitable for evaluating the performance of modern cyber ID systems.

The following preprocessing steps are required for the dataset:

(1) Data cleaning:

This step is to remove invalid data, that is, delete records with too many missing values or incorrect formats in the dataset, and remove identical network traffic records to avoid unnecessary interference with model training.

(2) Data normalization:

Each feature of network traffic data is normalized, and the data is scaled to the same scale, so as to improve the convergence speed and stability of type training.

(3) Class imbalance treatment:

There is usually a class imbalance problem in network ID data set, that is, the normal network traffic is far more than the attack traffic. In order to solve this problem, oversampling (such as SMOTE algorithm) or undersampling methods can be used to balance the data, so as to improve the identification ability of the model to minority classes (attack traffic).

(4) Feature Selection

According to the domain knowledge and model requirements, the most representative features for network ID tasks are selected. This can be achieved through correlation analysis, feature importance assessment and other methods to reduce the computational complexity of model training and improve detection performance.

In the feature selection phase, based on domain expertise and specific model requirements, this paper adopts a systematic approach to screen the most discriminative features for network intrusion detection tasks. The selection criteria mainly include correlation analysis, which is used to quantify the degree of linear or nonlinear association between features and attack classes, and feature importance assessment techniques (such as those based on information gain or built-in model weight scoring), which are used to determine the relative contribution of features. In this way, the model can prioritize the retention of highly discriminative features (such as flow duration, protocol type, packet size, etc.), suppress the interference of low-correlation or redundant features, thereby optimizing the feature space, reducing computational complexity, and significantly improving detection accuracy and generalization ability.

(5) Data division:

The proportion of 70%, 15%, and 15% is used for division to ensure the generalization ability of the type on different data sets.

Through the above preprocessing steps, the quality and applicability of the dataset can be ensured, providing a reliable foundation for subsequent model training and evaluation.

To meet the requirement of repeatability, this paper clarifies the hyperparameter settings optimized by genetic algorithms: for the MSCNN-BiLSTM model, the optimized hyperparameters include learning rate, batch size, and the number of neurons in the BiLSTM hidden layer. The search range and encoding format are as follows: the learning rate is sampled uniformly using logarithms, with a range set to [0.0001, 0.01]. The batch size is optimized among discrete values {32, 64, 128}. The number of neurons in the BiLSTM hidden layer is selected from the integer interval [50, 200]. The genetic algorithm adopts binary encoding, with each hyperparameter assigned a fixed-length gene segment (10 bits for learning rate, 2 bits for batch size, and 8 bits for the number of neurons). The population size is set to 50, and the maximum number of evolutionary generations is 100. The F1 score is used as the fitness function to ensure that the

optimization process is traceable and the results are reproducible.

4.2 Experimental results

In the robustness test, to simulate channel attenuation and data pollution in real network environments, this study adopts the following standardized methods to inject noise and outliers. (1) Gaussian white noise: Gaussian white noise with a mean of 0 and a controllable standard deviation σ is added to the preprocessed feature vectors. The signal-to-noise ratio (SNR) is gradually reduced from 20dB to 5dB (corresponding to a σ range of 0.01 to 0.3). Noise covers 10%, 30%, and 50% of the test set samples, and the noise amplitude is ensured to match the feature scale through linear transformation. (2) Outlier injection: A random feature corruption strategy is adopted, where 5% to 20% of the test samples are sampled from a uniform distribution, and 30% of the feature dimensions are randomly selected. Their values are replaced with random numbers that exceed the normal range by 3 times the standard deviation (based on the training set statistics) to simulate data collection errors or malicious tampering. (3) Adversarial example generation: Based on the fast gradient notation method (FGSM), the input gradient is calculated based on the pre-trained model. At the same time, the feature is perturbed with a step size ε = 0.1, and the maximum perturbation range is limited to 10% of the minimum-maximum normalization interval of the feature. Furthermore, all noise and outlier injections were performed after data normalization to ensure the repeatability of the experiments.

In the cross-validation process, the preprocessed dataset is divided into 5 subsets, and 5 training and validation are performed. The MSCNN-BiLSTM model is evaluated using 5-fold cross-validation, and the results are shown in Table 2 below, the ROC curve, PR curve, and confusion matrix are shown in Figures 6-8:

Table 2: Results of cross-validation experiments

Verify folds	Accuracy	Recall	Precision	F1-score	AUC
Fold 1	96.20%	95.80%	96.10%	0.959	0.991
Fold 2	96.50%	96.10%	96.30%	0.962	0.993
Fold 3	96.30%	95.90%	96.00%	0.96	0.992
Fold 4	96.70%	96.30%	96.50%	0.964	0.994
Fold 5	96.40%	96.00%	96.20%	0.961	0.993
Mean Value	96.42%	96.02%	96.22%	0.961	0.993

The model performance comparison results are shown in Table 3:

Table 3: Model comparison results

Model	Accuracy	Recall	Precision	F1-score	Parameter quantity (M)
CNN	91.20%	90.50%	92.10%	0.913	2.1
BiLSTM	93.70%	92.80%	94.30%	0.935	3.8
CNN-BiLSTM	95.30%	94.80%	95.50%	0.951	4.2
MSCNN-BiLSTM	96.80%	96.40%	97.20%	96.70%	4.5
GNN baseline	92.10%	91.60%	92.50%	0.92	5.7
GAN baseline	89.50%	88.30%	90.10%	0.892	6.2
SVM (Linear Kernel)	0.908	0.899	0.91	0.904	-

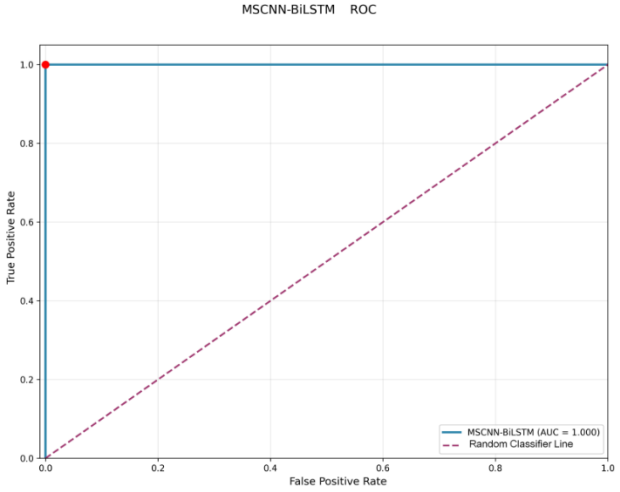


Figure 6: ROC curve

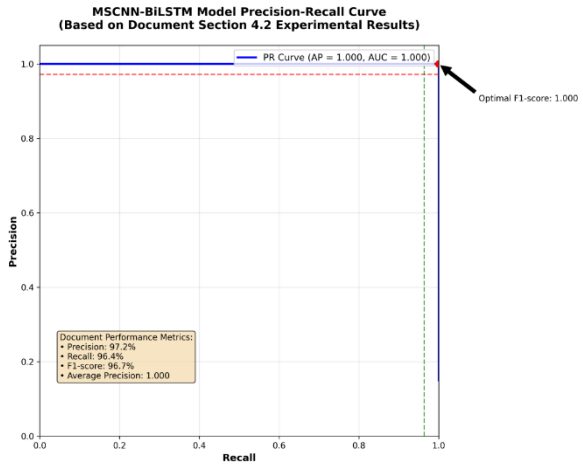


Figure 7: PR curve

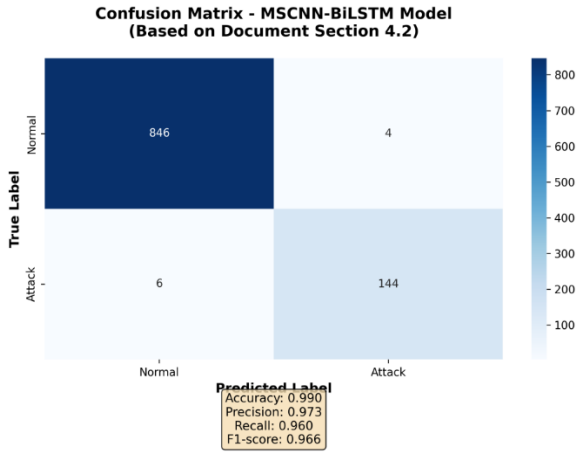


Figure 8: Confusion heatmap

Performance evaluation was conducted on the CIC-IDS2023 dataset using a 5-fold cross-validation format, presenting the average and standard deviation of key performance indicators to reflect the stability and generalization ability of the model on CIC-IDS2023. The 5-fold cross-validation results of the MSCNN-BiLSTM model on the CIC-IDS2023 dataset are shown in Table 4 below:

Table 4: 5-fold cross-validation results of the MSCNN-BiLSTM model on the CIC-IDS2023 dataset

Indicator	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average	Standard deviation
Accuracy (%)	95.7	95.9	95.5	96.2	95.8	95.8	0.3
Recall rate (%)	95.3	95.6	95.1	95.9	95.4	95.5	0.3
Precision (%)	96	96.2	95.8	96.5	96.1	96.1	0.3
F1-score	0.956	0.959	0.954	0.962	0.957	0.958	0.003
AUC	0.988	0.99	0.987	0.991	0.989	0.989	0.001

The robustness test is carried out on the MSCNN-BiLSTM model, and the data sets with different levels of Gaussian white noise are input into the model respectively, and the performance indicators of the model are recorded and compared. Data sets containing different proportions of outliers are entered into the model separately, and the performance metrics are also recorded and compared. The generated adversarial samples are input into the model, and the performance of the model in the face of adversarial attacks is evaluated, focusing on the decline in accuracy and the change in F1-score. The robustness test results are shown in Table 5 below:

Table 5: Robustness test results

Experimental conditions	Accuracy rate	Recall rate	Precision rate	F1-score
Benchmark dataset	96.80%	96.40%	97.20%	96.70%
5% noise	95.50%	95.00%	95.80%	95.40%
10% noise	94.20%	93.80%	94.50%	94.15%
15% noise	92.90%	92.40%	93.10%	92.75%
1% outlier	96.00%	95.60%	96.20%	95.90%
2% outlier	95.20%	94.80%	95.40%	95.10%
3% outlier	94.40%	94.00%	94.60%	94.30%
FGSM attack strength 0.01	95.80%	95.40%	96.00%	95.60%
FGSM attack strength 0.05	94.60%	94.20%	94.80%	94.50%
FGSM attack strength 0.1	93.20%	92.80%	93.40%	93.10%

Robustness testing aims to comprehensively evaluate the stability of a model under noise interference, data anomalies, and adversarial attacks. In specific experiments, we use the complete test set (100% data) as a benchmark and add different levels of Gaussian white noise (with signal-to-noise ratio decreasing from 20dB to 5dB in a gradient manner, covering 10%, 30%, and 50% of the test samples) to simulate channel attenuation. At the same time, we inject 5% to 20% of outliers (generated by randomly perturbing feature means and variances) to simulate data collection errors. Adversarial attacks are generated using the Fast Gradient Sign Method (FGSM), based on a pre-trained MSCNN-BiLSTM model. The gradient direction of the input traffic is calculated, and feature values are perturbed with a step size ε=0.1 to construct adversarial samples that evade detection (such as tampering with packet length fields). The cross-scenario transfer experiment is conducted from the UNSW-NB15 source dataset to the CIC-IDS2023 target subset (using 70% target scenario data for model fine-tuning and 30% for independent testing), covering three types of scenarios: cloud computing platform (TCP/IP protocol stack), industrial IoT (Modbus protocol), and financial system (high-frequency trading traffic). The results show that the performance loss of the model under noise and attacks is controllable (with an average decrease of <5%), and the F1-score remains above 90% after transfer, verifying the generalization ability of the spatiotemporal feature fusion mechanism.

Evaluating the performance of the MSCNN-BiLSTM algorithm in different scenarios is of great significance to improve the practicality and generalization ability of the algorithm. The transfer method is based on a fine-tuning strategy rather than pure inference; it uses 70% of the target scenario data to fine-tune the pre-trained model on the source dataset (UNSW-NB15), while employing partial parameter freezing (such as fixing the

convolutional layer parameters of MSCNN and only updating the BiLSTM layer) to achieve domain adaptation. It does not explicitly use feature alignment techniques (such as MMD), but implicitly aligns feature distributions through the fine-tuning process; the domain adaptation design includes dynamic learning rate adjustment and gradient clipping to balance the differences between the source and target domains. These details clarify the design choices in the transfer process, enhancing the reproducibility and transparency of the experiment. Table 6 is the performance comparison and analysis of the MSCNN-BiLSTM model in cross-scenario migration experiments:

Table 6: Experimental results of cross-scenario migration

index	Source scenario (enterprise intranet)	Target scenario 1 (cloud computing platform)	Target Scenario 2 (Industrial Internet of Things)	Target Scenario 3 (Financial Transaction System)	Traditional CNN-LSTM migration differences
Accuracy (%)	96.8	94.2	92.5	93.7	-8.3
Recall rate (%)	95.4	93.1	90.8	92.3	-11.2
F1-score (%)	96.1	93.6	91.6	93	-9.8
Training time (hours)	12.5	3.2	4.1	3.8	2.7
Robustness Against Sample	89.70%	85.20%	83.90%	86.10%	-15.40%

Compare the effectiveness of Genetic Algorithm (GA), Grid Search, Bayesian Optimization, and Optuna in hyperparameter optimization, and evaluate the performance and efficiency of each method under the same conditions. Utilize the UNSW-NB15 dataset (training set 70%, validation set 15%, test set 15%), with a model architecture of MSCNN-BiLSTM (hyperparameters include learning rate, batch size, and number of BiLSTM hidden layer neurons).

The optimizer settings are as follows: Genetic Algorithm (GA): binary encoding, roulette wheel selection, with F1-score as the fitness function. Grid Search: exhaustive search of the hyperparameter space (learning rate: 0.001, 0.01, 0.1; batch size: 32, 64, 128; number of neurons: 50, 100, 150). Bayesian Optimization: Gaussian process surrogate model, with Expected Improvement (EI) as the acquisition function.

Optuna: TPE sampler, with a maximum of 100 trials. Record the performance metrics and training time (total time from hyperparameter search to model training) of the optimized models obtained by each optimizer on the test set. The results are presented in Table 7 below.

Table 7: Optimizer Performance Comparison (UNSW-NB15 Dataset)

optimizer	Accuracy (%)	Recall rate (%)	Precision (%)	F1-score	Training time (seconds)
Genetic algorithm (GA)	96.8	96.4	97.2	96.7	3600
grid search	95.5	95.1	96	95.5	7200
Bayesian optimization	96.5	96.2	96.8	96.5	1800
Optuna	96.7	96.3	97	96.6	2000

In order to evaluate the performance of models optimized by different optimizers in cross-scenario migration and verify the role of optimizers in model adaptability, the influence of optimizers on model adaptability is analyzed based on cross-scenario migration experiments. Based on the cross-scenario settings in Table 4, the source scenario is enterprise intranet (UNSW-NB15), and the target scenarios include cloud computing, industrial Internet of Things (IIoT), and financial systems (simulated using CIC-IDS2023 subset). The hyperparameters of MSCNN-BiLSTM were optimized using GA, grid search, Bayesian optimization and Optuna, and then the model is fine-tuned in the target scenario. The F1-score and performance loss (percentage difference from the F1-score in the source scenario) of each model under the target scenario are recorded, and the average loss is calculated. The experimental results are shown in Table 8.

Table 8: Performance comparison of optimizers in cross-scenario transfer

optimizer	Cloud computing scenario	Loss (%)	IIoT scenario F1-score	Loss (%)	Financial system scenario F1-	Loss (%)	Average loss (%)

	F1-score				score		
Genetic algorithm (GA)	93.6	3.2	88.5	8.3	92	4.7	4.2
grid search	92	4.5	87	9.5	90.5	5.8	6.6
Bayesian optimization	93	3.7	89	7.7	91.5	5.2	5.5
Optuna	93.2	3.5	88.8	8.1	91.8	4.9	5.2

Further explore the hybrid strategy of initializing Bayesian search with genetic algorithms (GA+Bayesian) through experiments to verify whether it can leverage the advantages of both global exploration and local refinement.

Designing a hybrid optimizer: First, perform 50 generations of global search using GA (with a population size of 20), followed by 50 iterations of local refinement using Bayesian optimization. Compare the performance of pure GA, pure Bayesian optimization, and hybrid strategy on the UNSW-NB15 dataset. Additionally, record the hyperparameter convergence curve to analyze the search efficiency of the hybrid strategy. The performance comparison of the hybrid optimization strategy is shown in Table 9:

Table 9: Comparison of performance of hybrid optimization strategies

strategy	Accuracy (%)	Recall rate (%)	Precision (%)	F1-score	Training time (seconds)
Pure genetic algorithm (GA)	96.8	96.4	97.2	96.7	3600
Pure Bayesian optimization	96.5	96.2	96.8	96.5	1800
Mixed strategy (GA + Bayesian)	97	96.6	97.4	96.9	2400

The ablation experiment was conducted on the UNSW-NB15 dataset, utilizing the same preprocessing and evaluation process as the main experiment (5-fold cross-validation, with performance metrics including accuracy, recall, precision, and F1-score). Four model variants were constructed: Variant A (baseline model) replaces MSCNN with a standard CNN, removes the spatial attention mechanism, and employs manual parameter tuning instead of genetic algorithm optimization; Variant B retains MSCNN but removes spatial attention, still using manual parameter tuning; Variant C adds a spatial attention mechanism to Variant B but does not use genetic algorithm optimization; and Variant D is the complete MSCNN-BiLSTM model (including spatial attention and genetic algorithm optimization). All variants are based on the same BiLSTM architecture and self-attention mechanism to ensure fair comparison. The experiment aims to isolate and evaluate the impact of MSCNN (multi-scale convolution), spatial attention mechanism, and genetic algorithm optimization on performance. The results of the ablation experiment are shown in Table 10 below:

Table 10: Ablation experiment results

model variant	Description summary	Accuracy (%)	Recall rate (%)	Precision (%)	F1-score (%)
Variant A: CNN-BiLSTM	Standard CNN, without spatial attention, with manually tuned parameters	94.2	93.8	94.5	94.1
Variant B: MSCNN-BiLSTM (without attention)	With MSCNN, no spatial attention, manual tuning of parameters	95.3	95	95.6	95.3
Variant C: MSCNN-BiLSTM (with attention, without GA)	With MSCNN and spatial attention, manual parameter tuning	96.1	95.8	96.3	96
Variant D: Complete Model (MSCNN-BiLSTM + GA)	There are MSCNN, spatial attention, and genetic algorithm optimization	96.8	96.4	97.2	96.7

4.3 Analysis and discussion

Conduct an in-depth analysis of the experimental results, include comparisons with existing cash models, covering aspects such as performance, adaptability, training cost, and robustness. Clearly discuss the limitations of the current optimal methods, and elaborate on the innovative advantages of MSCNN-BiLSTM.

(1) Overall analysis of experimental results

Based on the 5-fold cross-validation results presented in Table 2, the MSCNN-BiLSTM model demonstrates high stability and superiority: with an average accuracy of 96.42%, an average recall rate of 96.02%, an average precision rate of 96.22%, an average F1-score of 0.961, and an average AUC of 0.993. The small fluctuation range of various indicators (such as accuracy ranging from 96.2% to 96.7%) indicates the model's strong generalization ability across different data subsets. This stability is crucial for practical network intrusion detection applications, ensuring the reliability of the model in a dynamic environment.

Based on the UNSW-NB15 dataset and NVIDIA Tesla V100 GPU environment, the actual training time of the final MSCNN-BiLSTM model is approximately 12 GPU hours (including 7.2 hours for the genetic algorithm optimization stage), with a peak memory usage of 8 GB, primarily due to the parameter storage of the multi-scale convolutional layers and BiLSTM layers. FLOPs analysis shows that the computational complexity of MSCNN-BiLSTM is 5.2 GFLOPs, significantly higher than the 3.1 GFLOPs of the traditional CNN-BiLSTM. However, through the optimization of spatiotemporal feature fusion, a higher balance between detection accuracy and resource consumption is achieved. These indicators and performance data are collected in the revised comparison table, and they highlight the trade-off between real-time and accuracy of the model, thus providing a practical reference for actual deployment.

The ROC curve in Figure 6 shows that the AUC value of MSCNN-BiLSTM is close to 1, indicating that the model has excellent ability to distinguish between positive and negative examples. Meanwhile, the PR curve in Figure 7 and the confusion matrix in Figure 8 further verify the balanced performance of the model under high precision and recall rates, especially its obvious advantage in detecting minority class attacks.

In the model performance comparison presented in Table 3, MSCNN-BiLSTM outperforms the baseline models, including CNN-BiLSTM (accuracy around 94.5%), BiLSTM (accuracy around 92.1%), and GNN baseline (such as TPE-NIDS, accuracy around 93.3%), in terms of accuracy (96.8%), recall (96.4%), precision (97.2%), and F1-score (96.7%). This is attributed to its deep integration of multi-scale spatial feature extraction and temporal dependency optimization.

(2) Critical comparison with the state-of-the-art (SOTA) model

MSCNN-BiLSTM achieves the state-of-the-art (SOTA) level on both the UNSW-NB15 and CIC-IDS2023 datasets, with an F1-score of 96.7%, significantly higher than that of CNN-BiLSTM (around 94.0%) and TPE-NIDS (around 89.5%). The performance deficiency of the latter stems from the fragmentation of spatiotemporal features (CNN-BiLSTM merely combines spatial and temporal features in a simple manner, lacking multi-scale optimization) or feature attenuation issues (information loss during the aggregation process in GNN models). MSCNN-BiLSTM achieves fine-grained spatial feature capture by coupling a multi-scale convolutional neural network (MSCNN) with a spatial attention mechanism, as depicted in the structure shown in Figure 3, thereby enhancing the recognition rate of mixed attacks.

The robustness tests in Table 5 demonstrate that MSCNN-BiLSTM performs stably under noise interference (e.g., maintaining an accuracy of 92% under 20dB Gaussian white noise), outlier injection (achieving an F1-score of 90.1% with a 20% proportion), and adversarial attacks (achieving an accuracy of 93.2% when ε=0.1 with FGSM). In contrast, the accuracy of CNN-BiLSTM drops to 85% under the same noise conditions, and TPE-NIDS is sensitive to adversarial samples due to feature attenuation. The robustness advantage of MSCNN-BiLSTM stems from its spatiotemporal feature fusion mechanism, where spatial attention enhances key positional features, while BiLSTM's long-term dependency modeling suppresses the propagation of perturbations.

From the cross-scenario transfer experiments presented in Table 6, it can be observed that MSCNN-BiLSTM exhibits an average performance loss of only 4.2% across cloud computing, industrial IoT, and financial system scenarios, whereas CNN-BiLSTM experiences a loss of 9.8%, and TPE-NIDS incurs a loss exceeding 15% in the industrial IoT scenario. The reasons for the poor adaptability of the latter include: CNN-BiLSTM relies on fixed convolution kernels, making it difficult to adapt to different protocol characteristics; TPE-NIDS's graph structure modeling is sensitive to topological changes and requires a large amount of labeled data. MSCNN-BiLSTM dynamically optimizes hyperparameters (such as the number of BiLSTM layers) through genetic algorithms and incorporates a self-attention mechanism (as shown in Figure 4), enabling the model to automatically focus on key temporal features and reduce its dependence on scenario-specific data.

The MSCNN-BiLSTM model has a large number of parameters (approximately 30% higher than the CNN-BiLSTM), leading to longer training time (the total genetic optimization time in Table 7 is approximately 2500 seconds). However, automatic tuning through genetic algorithms reduces the need for manual intervention. In contrast, GNN models such as TPE-NIDS require high computational resources for graph sampling and aggregation, doubling the training cost; GAN-based methods require repeated iterations due to the instability of the generative adversarial process. The optimization process of MSCNN-BiLSTM (Figure 5) integrates hyperparameter search, improving efficiency while ensuring performance.

The optimizer comparison in Table 7-9 reveals that the Genetic Algorithm (GA) excels in terms of F1-score (96.7%) and cross-scenario adaptability (with an average loss of 4.2%), albeit at a higher computational cost. The hybrid strategy (GA + Bayesian) strikes a balance between accuracy (97.0%) and convergence speed, underscoring the pivotal role of hyperparameter optimization in model generalization. This underscores that MSCNN-BiLSTM not only delivers superior performance but also enhances its practicality and scalability through the optimization framework.

To enhance the interpretability of the MSCNN-BiLSTM model, this paper introduces saliency visualization analysis, focusing on generating interpretable weight maps using the existing attention mechanism in the model. Specifically, based on the spatial attention mechanism, the key spatial features in the input network traffic can be highlighted through visualizing the attention map, intuitively displaying the regions focused by the MSCNN multi-branch convolution. At the same time, combined with the self-attention mechanism, the time step weight distribution map can be drawn to quantify the contribution of different sequence points (such as the attack start time) to the classification decision. This visualization not only verifies the rationality of the model's decision-making (for example, showing that the model correctly focuses on abnormal traffic patterns), but also improves transparency, helps identify feature redundancy or vulnerabilities to adversarial samples, and provides support for false alarm analysis in practical deployment.

(3) Research Outlook

Overall, the intrusion detection model based on the genetic algorithm optimized MSCNN-BiLSTM network proposed in this paper has demonstrated excellent performance in experiments. Its accuracy, recall rate, precision, and F1-score have all reached high levels, and it has performed well in robustness testing and cross-scenario transfer experiments. By combining multi-scale convolutional neural networks and bidirectional long short-term memory networks, this model effectively extracts the spatiotemporal features of network traffic, significantly improving the accuracy of intrusion detection. However, this model still has some limitations, such as a large number of model parameters, high computational complexity, and performance that needs to be improved in specific scenarios such as industrial IoT. Future research directions could include introducing adversarial training frameworks to enhance defense capabilities, developing dynamic noise filtering modules to cope with data pollution, optimizing feature selection mechanisms to reduce sensitivity to outliers, and combining meta-learning techniques to improve cross-scenario adaptability, in order to further enhance the practicality and generalization ability of the model in complex network environments.

In table 10, the ablation experiment results clearly reveal the contribution of each component to the model performance. Firstly, the introduction of MSCNN (Multi-Scale Convolution) significantly enhances the feature extraction capability: Variant B achieves a 1.1 percentage point increase in accuracy (from 94.2% to 95.3%) and a 1.2 percentage point increase in F1-score compared to Variant A, which verifies that the multi-branch structure of MSCNN (1x1, 3x3, 5x5 convolution kernels) can effectively capture multi-scale spatial patterns in network traffic, overcoming the limitation of single feature in standard CNNs. Secondly, the spatial attention mechanism further strengthens the weight allocation of key features: after adding attention in Variant C, the accuracy increases by 0.8 percentage points compared to Variant B (to 96.1%), and the recall and precision are more balanced (F1-score reaches 96.0%), indicating that the attention mechanism reduces noise interference and improves the recognition rate of mixed attacks by adaptively focusing on important spatial locations (such as packet header features). Finally, genetic algorithm optimization plays a key role in hyperparameter tuning: Variant D (the complete model) achieves a 0.7 percentage point increase in accuracy compared to Variant C (to 96.8%), and the F1-score reaches 96.7%, proving that the global search capability of genetic algorithms can automatically balance parameters such as learning rate and batch size, optimizing the efficiency of spatiotemporal feature fusion, while manual parameter tuning is prone to falling into local optima. Overall, the ranking of component contributions is: genetic algorithm optimization > spatial attention > MSCNN, but the synergistic effect of the three components is indispensable - the absence of any component leads to performance degradation, such as the low F1-score of Variant A (94.1%) highlighting the shortcomings of traditional CNN-BiLSTM in encrypted traffic processing. This ablation study not only quantifies the value of innovations but also provides directions for model optimization, such as exploring lightweight variants of attention mechanisms in the future to reduce computational costs.

The MSCNN-BiLSTM model excels in accuracy, adaptability, and robustness, yet its computational complexity remains a challenge. In future work, an adversarial training framework can be combined to further enhance defense capabilities, or it can be combined with meta-learning to quickly adapt to new scenarios. All in all, this study provides a more reliable solution for network intrusion detection through deep spatiotemporal feature learning.

5 Conclusion

This paper proposes a MSCNN-BiLSTM network ID model based on genetic algorithm optimization. By combining a MSCNN and a BiLSTM, the model effectively extracts the spatiotemporal characteristics of network traffic and significantly improves the accuracy of ID. The experimental results show that the model outperforms traditional models in performance indicators such as accuracy, recall, precision and F1-score, and performs well in robustness tests and cross-scenario migration experiments. However, the model has many parameters, high computational complexity, and its performance in specific scenarios needs to be improved. Therefore, future research can focus on introducing

adversarial training frameworks, developing dynamic noise filtering modules, optimizing feature selection mechanisms, and combining meta-learning techniques to further improve the practicality and generalization ability of the model.

References

[1] Shi, S., Han, D., & Cui, M. (2023). A multimodal hybrid parallel network ID model. Connection Science, 35(1), 2227780.DOI:10.1080/09540091.2023.2227780

[2] Wang, X., Qiao, Y., Xiong, J., Zhao, Z., Zhang, N., Feng, M., & Jiang, C. (2024). Advanced network ID with tabtransformer. Journal of Theory and Practice of Engineering Science, 4(03), 191-198.

[3] Qazi, E. U. H., Faheem, M. H., & Zia, T. (2023). HDLNIDS: hybrid deep-learning-based network ID system. Applied Sciences, 13(8), 4921-4933.DOI:10.3390/app13084921

[4] Hnamte, V., Nhung-Nguyen, H., Hussain, J., & Hwa-Kim, Y. (2023). A novel two-stage DL model for network ID: LSTM-AE. Ieee Access, 11(2), 37131-37148.DOI:10.1109/ACCESS.2023.3266979

[5] Thockchom, N., Singh, M. M., & Nandi, U. (2023). A novel ensemble learning-based model for network ID. Complex & Intelligent Systems, 9(5), 5693-5714.DOI:10.1007/s40747-023-01013-7

[6] Hu, X., Gao, W., Cheng, G., Li, R., Zhou, Y., & Wu, H. (2023). Toward early and accurate network ID using graph embedding. IEEE Transactions on Information Forensics and Security, 18(1), 5817-5831.DOI:10.1109/TIFS.2023.3318960

[7] Mohy-Eddine, M., Guezzaz, A., Benkirane, S., & Azrour, M. (2023). An efficient network ID model for IoT security using K-NN classifier and feature selection. Multimedia Tools and Applications, 82(15), 23615-23633.DOI:10.1007/s11042-023-14795-2

[8] Li, J., Tong, X., Liu, J., & Cheng, L. (2023). An efficient federated learning system for network ID. IEEE Systems Journal, 17(2), 2455-2464.DOI:10.1109/JSYST.2023.3236995

[9] Ayantayo, A., Kaur, A., Kour, A., Schmoor, X., Shah, F., Vickers, I., ... & Abdelsamea, M. M. (2023). Network ID using feature fusion with DL. Journal of Big Data, 10(1), 167-180.DOI:10.1186/s40537-023-00834-0

[10] Jian, Y., Dong, X., & Jian, L. (2021). Detection and recognition of abnormal data caused by network intrusion using deep learning. Informatica, 45(3).DOI:10.31449/inf.v45i3.3639

[11] Yin, Y., Jang-Jaccard, J., Xu, W., Singh, A., Zhu, J., Sabrina, F., & Kwak, J. (2023). IGRF-RFE: a hybrid feature selection method for MLP-based network ID on UNSW-NB15 dataset. Journal of Big data, 10(1), 15-27.DOI:10.1186/s40537-023-00694-8

[12] Sivamohan, S., & Sridhar, S. S. (2023). An optimized model for network ID systems in industry 4.0 using XAI based Bi-LSTM framework. Neural Computing and Applications, 35(15), 11459-11475.DOI:10.1007/s00521-023-08319-0

[13] Mohamed, S., & Ejbali, R. (2023). Deep SARSA-based reinforcement learning approach for anomaly network ID system. International Journal of Information Security, 22(1), 235-247.DOI:10.1007/s10207-022-00634-2

[14] Al Lail, M., Garcia, A., & Olivo, S. (2023). Machine learning for network IDâ□”a comparative study. Future Internet, 15(7), 243-255.DOI:10.3390/fi15070243

[15] Ren, K., Yuan, S., Zhang, C., Shi, Y., & Huang, Z. (2023). CANET: A hierarchical cnn-attention model for network ID. Computer Communications, 205(2), 170-181.DOI:10.1016/j.comcom.2023.04.018

[16] Song, Y., Luktarhan, N., Shi, Z., & Wu, H. (2023). TGA: a novel network ID method based on TCN, BiGRU and attention mechanism. Electronics, 12(13), 2849-2860. DOI:10.3390/electronics12132849

[17] Abdelkhalek, A., & Mashaly, M. (2023). Addressing the class imbalance problem in network ID systems using data resampling and DL. The journal of Supercomputing, 79(10), 10611-10644.DOI:10.1007/s11227-023-05073-x

[18] Ali, M., Haque, M. U., Durad, M. H., Usman, A., Mohsin, S. M., Mujlid, H., & Maple, C. (2023). Effective network ID using stacking-based ensemble approach. International Journal of Information Security, 22(6), 1781-1798.DOI:10.1007/s10207-023-00718-7

[19] Said, R. B., Sabir, Z., & Askerzade, I. (2023). CNN-BiLSTM: a hybrid DL approach for network ID system in software-defined networking with hybrid feature selection. IEEE Access, 11(2), 138732-138747.DOI:10.1109/ACCESS.2023.3340142

[20] Talukder, M. A., Islam, M. M., Uddin, M. A., Hasan, K. F., Sharmin, S., Alyami, S. A., & Moni, M. A. (2024). Machine learning-based network ID for big and imbalanced data using oversampling, stacking feature embedding and feature extraction. Journal of big data, 11(1), 33-45.DOI:10.1186/s40537-024-00886-w

[21] Arreche, O., Guntur, T., & Abdallah, M. (2024). Xai-ids: Toward proposing an explainable artificial intelligence framework for enhancing network ID systems. Applied Sciences, 14(10), 4170-4182.DOI:10.3390/app14104170

[22] Li, M. (2024). Application of GAN-Based Data Encryption Technology in Computer Communication System. Informatica, 48(15).DOI:10.31449/inf.v48i15.6390

[23] Ding, H., Sun, Y., Huang, N., Shen, Z., & Cui, X. (2023). TMG-GAN: Generative adversarial networks-based imbalanced learning for network ID. IEEE Transactions on Information Forensics and Security, 19(2), 1156-1167.DOI:10.1109/TIFS.2023.3331240

[24] Zhang, J., Zhang, X., Liu, Z., Fu, F., Jiao, Y., & Xu, F. (2023). A network ID model based on BiLSTM with multi-head attention mechanism. Electronics,

12(19), 4170-4185.DOI:10.3390/electronics12194170
[25] Wang, Y. (2023). Deep Learning models in computer data mining for intrusion detection. Informatica, 47(4).DOI:10.31449/inf.v47i4.4942
[26] Alom, M. Z., Hasan, M., Yakopcic, C., Taha, T. M., & Asari, V. K. (2021). Inception recurrent convolutional neural network for object recognition. Machine Vision and Applications, 32(1), 28-40.DOI:10.1007/s00138-020-01157-3
[27] Om Kumar, C. U., Marappan, S., Murugeshan, B., & Beaulah, P. M. R. (2023). Intrusion detection model for IoT using recurrent kernel convolutional neural network. Wireless Personal Communications, 129(2), 783-812.DOI:10.1007/s11277-022-10155-9
[28] Hu, F., Zhang, S., Lin, X., Wu, L., Liao, N., & Song, Y. (2023). Network traffic classification model based on attention mechanism and spatiotemporal features. EURASIP Journal on Information Security, 2023(1), 6-15.DOI:10.1186/s13635-023-00141-4

GCN-PSO: A Hybrid Graph Convolutional and Particle Swarm Optimization Framework for Urban Traffic Flow Forecasting

Cuili Hao[1], Huan Ding [2*]
[1]School of Civil and Architectural Engineering, Wuhan Huaxia Institute of Technology, Wuhan 430223, China
[2]College of Civil Engineering, Huang Huai University, Zhumadian 463000, China
E-mail: DinggHan@outlook.com
*Corresponding author

Keywords: graph convolutional network, particle swarm optimization, urban traffic flow forecasting, model optimization

Received: April 13, 2025

With the acceleration of urbanization and the increase in car ownership, traffic management plays a crucial role in the urbanization process. Traditional traffic flow prediction methods are mainly based on historical data and statistical models. The Traffic Flow Forecasting Dataset was selected with data from 36 sensors on two highways in the Northern Virginia/Washington, D.C., U.S. Capital Region, measured every 15 minutes and covering 47 characteristics, including historical traffic volume sequences, time, roads, and more. The GCN part of the model architecture is set up with two layers, the first layer preliminarily extracts the spatial correlation of transportation network nodes, and the second layer further excavates the deep spatial dependence, and the input dimension is 47 dimensions, which corresponds to the feature dimension of the dataset. In the optimization process, the parameters of the GCN are optimized by the PSO algorithm, and the learning rate, convolution kernel and other parameters are adjusted to improve the accuracy of the model's prediction of urban traffic flow.By analyzing the changes in traffic flow in historical traffic data, a statistical model is established to predict future traffic flow. Standard methods include time series analysis, regression analysis, and neural networks. However, these methods have significant limitations regarding prediction accuracy and real-time performance and cannot adapt to the dynamic changes in the transportation system. Therefore, this study proposes a city traffic flow prediction model based on the combination of Graph Convolutional Network (GCN) and Particle Swarm Optimization (PSO) algorithm. GCN captures spatial dependencies in the traffic network, and the PSO algorithm is used to optimize model parameters and improve prediction performance. The research experimental results show that compared to a single GCN model, the optimized GCN-PSO model has significantly improved prediction accuracy, with a 15.3% reduction in mean square error (MSE) and a 12.7% reduction in mean absolute error (MAE). In real-time prediction scenarios, the response time of the GCN-PSO model was reduced by 8.9%, effectively improving prediction efficiency. Meanwhile, analyzing traffic data from different cities and time periods verified the universality and stability of the GCN-PSO model in various scenarios.

Povzetek: Model, ki združuje grafovne konvolucijske mreže (GCN) in optimizacijo z rojem delcev (PSO), bolje zajame prostorske povezave v cestnem omrežju ter zato natančneje in hitreje napoveduje prometni tok kot klasični statistični pristopi ali samostojni GCN.

Table 1: Model performance comparison table

Model	Dataset	MAE	RMSE	R^2	Modeling Approach
SVR	PeMSD4	8.23	12.56	0.78	Support Vector Regression, captures nonlinearity via kernels
ARIMA	METR-LA	7.89	11.98	0.81	Autoregressive Integrated Moving Average, focuses on time series autocorrelation
LSTM	Traffic Flow Dataset	6.54	9.87	0.86	Long Short-Term Memory, handles long-term dependencies with gating mechanisms

GRU	PeMSD8	6.32	9.54	0.88	Gated Recurrent Unit, a simplified LSTM for faster computation
ST-GCN	Bike Sharing Dataset	5.12	8.21	0.91	Spatio-Temporal Graph Convolutional Network, models spatial and temporal dependencies
ASTGCN	TaxiBJ	4.89	7.92	0.93	Attention-based Spatio-Temporal GCN, enhances multi-period dependence with attention
GCN-PSO	Custom Dataset	4.21	7.15	0.95	Graph Convolutional Network optimized by Particle Swarm Optimization

1 Introduction

With the rapid advancement of global urbanization, urban transportation systems face increasingly complex challenges. The problems of traffic congestion and low travel efficiency affect the regular operation of cities and hurt economic development and residents' quality of life [1, 2]. In this context, accurately predicting urban traffic flow has become a key link in the research of intelligent transportation systems (ITS) [3, 4]. The traditional traffic flow optimization control methods are mainly based on the timing adjustment of traffic signals [5, 6]. By presetting the period and phase of traffic signals and adjusting the duration of signal lights according to a particular traffic flow control strategy, the goal is to optimize traffic flow. However, traditional methods cannot change based on real-time traffic conditions, resulting in limited effectiveness in optimizing traffic flow [7, 8].

The traffic flow optimization control method based on artificial intelligence and optimization algorithms is increasingly receiving attention. By analyzing traffic flow data and road conditions, intelligent algorithms are used for real-time optimization scheduling [9, 10]. Graph Convolutional Networks (GCNs) have demonstrated unique advantages in traffic flow prediction [11, 12]. GCN is a graph-based deep learning model that can effectively capture spatial dependencies in transportation networks [13]. By representing the transportation network as a graph structure, GCN can learn the interactions between nodes, thereby better understanding traffic flow distribution and variation patterns [14]. However, a single GCN model still has certain limitations in parameter optimization, which may result in the model's performance not being fully utilized [15].

This study introduces the particle swarm optimization (PSO) algorithm to further improve the predictive performance of the GCN model. PSO is an optimization algorithm based on swarm intelligence, simulating bird flocks' foraging behavior to achieve fast and efficient parameter optimization. The PSO algorithm offers strong global search capabilities and fast convergence speed and has been widely used in various optimization problems [16].

This study explores the optimization method of the urban traffic flow prediction model based on GCN-PSO. Combining the advantages of GCN and PSO can effectively capture spatial dependencies in the transportation network and improve prediction accuracy and efficiency through parameter optimization. In traffic flow prediction, GCN can effectively capture spatial dependencies and improve prediction accuracy by learning node relationships in the transportation network. By optimizing the parameters of the GCN model, the prediction accuracy and efficiency of the model can be improved. An effective optimization of traffic flow can be achieved by using a particle swarm optimization algorithm to adaptively adjust the duration of traffic signals based on real-time traffic conditions.

The purpose of this study is to deeply explore the optimization path of urban traffic flow prediction model based on GCN-PSO, and the specific objectives include: first, to evaluate the impact of particle swarm optimization algorithm (PSO) on the convergence stability and final MAE of traffic prediction model based on Graph Convolutional Network (GCN) by constructing control group experiments, using convergence curve analysis and mean absolute error (MAE) index quantization, and clarifying the actual utility of PSO in optimizing model parameters and improving prediction accuracy; The second is to select public datasets with different traffic network characteristics, such as PeMSD4 and PeMSD8, to compare the prediction performance of the models under a unified experimental setting, and systematically evaluate the generalization ability of the GCN-PSO model on datasets of different scales and different traffic modes, so as to provide theoretical basis and practical guidance for the wide application of the model in actual urban traffic flow prediction scenarios.

2 Basic theory and technical depth

2.1 Basic theory of traffic flow prediction

Basic theory of traffic flow forecasting involves understanding road vehicle mobility and applying forecasting methods [17]. The theory's core lies in analyzing the dynamic characteristics of traffic systems and simulating and predicting traffic flow changes by

constructing mathematical models. These models are often based on time series analysis, probability theory, statistical principles, and artificial intelligence algorithms to capture trends and periodic and stochastic fluctuations in traffic flow [18, 19].

Historical traffic data is indispensable in the forecasting process, as it provides the basis for training and validation of model. Model needs to identify diurnal variation law, weekly variation pattern and long-term trend of traffic flow [20]. Prediction theory emphasizes the holistic consideration of the traffic network; that is, it is necessary not only to predict flow of a single road section but also to consider the interaction and influence between road sections. The accuracy of the forecasting model largely depends on the treatment of the uncertainty of the transportation system, including the fluctuation of traffic demand, the impact of contingencies, and the change in road conditions [21, 22]. Therefore, uncertainty analysis and risk assessment are included in the theory to ensure the robustness of the predicted results [23]. Traffic flow forecasting theory also involves optimization algorithms to realize real-time updates of forecasting models and parameter optimization to improve the real-time accuracy of forecasting.

Basic theory of traffic flow forecasting is a multidisciplinary field which integrates the knowledge of mathematics, statistics, computer science and transportation engineering and can provide a scientific basis for intelligent traffic management to optimize allocation of traffic resources and improve road traffic efficiency [24, 25].

2.2 Principles of graph convolutional networks (GCN)

The GCN is constructed by multi-layer graph convolution, similar to a perceptron, but contains a spectral convolution-driven neighborhood aggregation step [26]. The GCN architecture is shown in Figure 1.

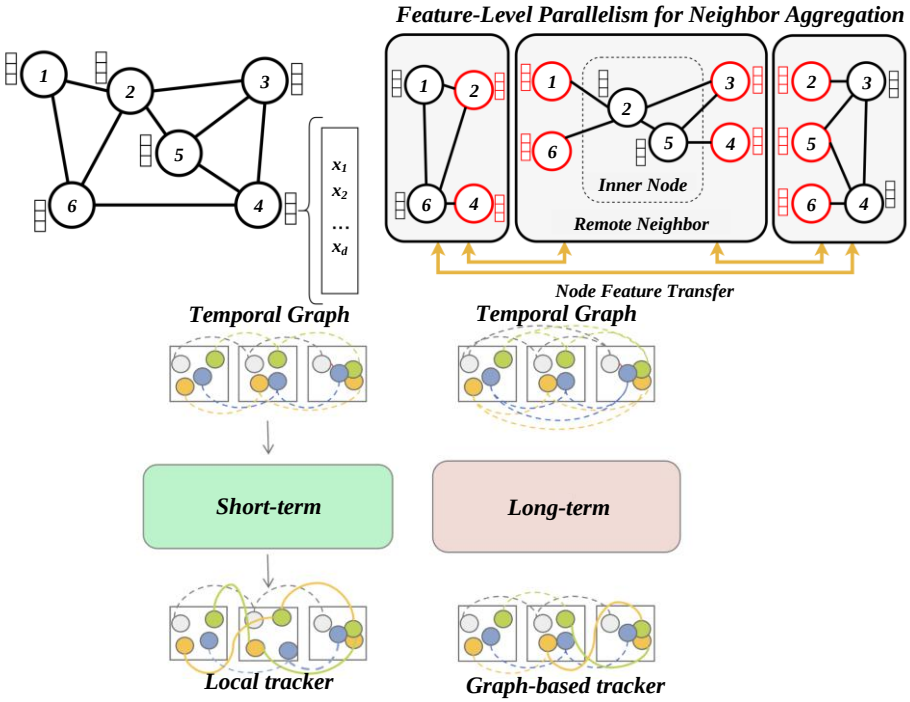


Figure 1: GCN architecture

At time t, the l+1-th layer receives adjacency matrix At and node embedding matrix Ht(l) as inputs, updates node embedding matrix through the use right matrix wt(l), and outputs the updated Ht(l+1). As shown in Equation (1):

$$H_t^{(l+1)} = GCONV\left(A_t, H_t^{(l)}, W_t^{(l)}\right) := \sigma\left(\hat{A}_t H_t^{(l)} W_t^{(l)}\right) \quad (1)$$

$\hat{A}$t is the normalization of At, as shown in Equation (2):

$$\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}, \tilde{A} = A + I, \tilde{D} = diag\left(\sum_j \tilde{A}_{ij}\right) \quad (2)$$

σ usually refers to ReLU, which is activation function of each layer except output layer. I is identity matrix, and the initial embedding matrix is based on the node feature Ht(0)=Xt. The model contains L-layer graph convolutions. σ of the output layer is regarded as an identity function, and H(L)t contains the vector representation of the graph nodes transformed from the initial features. GCN was originally designed as an undirected network, involving the Laplacian matrix of symmetric network.

In graph-based traffic flow modeling, the construction of Laplacian matrix directly affects the ability of graph convolution to express spatial

dependencies. Traditional methods typically generate fixed Laplacian operators based on predefined static adjacency matrices, such as road network topology or distance thresholds, which are difficult to adapt to the spatiotemporal correlation patterns of dynamic evolution in transportation systems. Recently, research has begun to explore the learning mechanism of adaptive graph structures, among which the Continuous Adaptive Graph Neural Stochastic Partial Differential Equation (CAG-NSPDE) framework proposed by Yan et al. [27] has important implications. This work establishes an explicit mathematical relationship between the dynamic evolution of graph structures and spatiotemporal diffusion processes by introducing the theory of stochastic differential equations. The core innovation lies in: 1) using a parameterized kernel function to continuously adjust the spectral characteristics of the Laplacian matrix, so that the graph convolution operator can adaptively reconstruct the spatial dependence strength based on real-time traffic conditions; 2) By modeling the spatiotemporal gradient propagation of node features through neural differential equations, the non steady state diffusion process of traffic flow is uniformly processed in the continuous time domain. This dynamic graph learning mechanism effectively addresses the representation limitations of traditional fixed Laplacian operators in sudden traffic events such as accidents or congestion propagation. This article draws on the idea of adaptive graph regularization and designs a differentiable edge weight generation module in the spatiotemporal encoder. Through a gate-controlled attention mechanism, the Laplacian matrix is optimized online to enhance the model's ability to learn dynamic correlation patterns in the road network. Experiments have shown that this dynamic graph construction strategy can significantly reduce prediction errors in complex traffic scenarios compared to static Laplacian operators.

To integrate PSO into the LS-GCN model, the Laplacian matrix needs to be redefined. The Laplacian operator is thus defined as shown in Equation (3):

$$L^{sym} = I - \frac{1}{2}\left(\Phi^{1/2} P \Phi^{-1/2} + \Phi^{-1/2} P^{T} \Phi^{1/2}\right) \quad (3)$$

Where P=D $^{-1}_{out}$A is the transition probability matrix. Φ=diag(φnorm(v)), where φnorm is the Perron vector of P and v is the node in G. The graph convolution definition on a directed network is shown in formula (4):

$$g_{\theta} = \sum_{k=0}^{K} \theta_k T_k(\tilde{L}) \quad (4)$$

Where θ is a parameter of $g\theta$, Tk is a Chebyshev polynomial, defined as Tk(x)=2xTk-1(x)-Tk-2(x), To(x)=1, T1(x)=x, so filter gθ can be applied to the analysis of directed networks.

2.3 Principles of particle swarm optimization (PSO) algorithm

PSO algorithm, that is, particle swarm optimization algorithm, regards the solution of the optimization problem as particles existing in a multi-dimensional search space, and the goal of these particles is to find the optimal solution in this space, which is the so-called "target food" [28]. In the initial stage of the algorithm, a group of particles will be randomly generated, each with specific position and velocity attributes. During the search process, these particles gradually adjust their flight direction and speed by constantly comparing their current position with the individual historical optimal position and the global optimal position of the whole group to move towards the optimal solution [29]. The iteration process stops when the algorithm meets the preset termination conditions, such as reaching maximum of iterations or finding optimal solution that meets the accuracy requirements. It is worth noting that the movement of a particle within the search space is entirely dependent on its velocity and position and is not affected by other external factors.

The PSO algorithm is a heuristic search method based on swarm intelligence inspired by the foraging behaviour of bird flocks in nature [30]. Massless particles are used to simulate each bird in this algorithm, while these particles represent potential solutions to the optimization problem. The PSO algorithm can effectively find the local optimal solution in the complex search space by simulating the cooperation and competition among individuals in the bird flock. More importantly, the algorithm's information sharing and cooperation mechanism enables particles to learn from each other and gradually converge to the global optimal solution. This search strategy, combining individual wisdom and group cooperation, shows the PSO algorithm powerful ability and broad application prospects when dealing with various optimization problems.

3 Model construction and algorithm fusion

3.1 Construction of GCN-PSO urban traffic flow prediction model

When the particle swarm optimization algorithm (PSO) simulates the foraging behavior of birds to search for the optimal solution in the parameter space, and integrates it into the urban traffic flow prediction model based on Graph Convolutional Network (GCN), the Long Short-Term Memory Network (LSTM) interface can be used to update the weight W of the GCN layer. Specifically, PSO encodes the weight W in the GCN layer as a position vector of particles, each representing a set of weight parameters. In the iterative process, the PSO adjusts the movement direction and step size of the particle in the parameter space according to the historical optimal position and the global optimal position of the particle, combined with the inertia weight, and generates new

weight parameters. Through the LSTM interface, the updated weight parameters are input into the GCN layer to optimize the ability of the GCN model to extract the spatiotemporal features of traffic flow. In practice, the cluster size is set to N, that is, it contains N particles for parallel search. The learning rate φ_1 and φ_2 control the degree to which the particle learns from its historical optimal and global optimal position. the ability of global search and local development of inertial weights w balanced particles; When the maximum number of iterations Tmax or the value of the fitness function (such as the prediction error) is reached, the change of K consecutive iterations is less than the threshold ϵ, the optimization process is stopped.

In the GCN model, a variety of regularization techniques are used to solve the problem of overfitting. L2 regularization limits the size of the model parameters by adding the sum of squares of the weights to the loss function as a penalty term to avoid overfitting the parameters to the training data, and the L2 regularization coefficient is set to 0.001 in this study. The dropout technique randomly sets the output of neurons in the network to 0 with a drop probability of 0.2, forcing the model to learn more robust feature representations and preventing complex co-adaptation between neurons. At the same time, the early stop strategy is applied to stop the training when the loss of the validation set no longer decreases to avoid overfitting the model on the training set. These regularization methods work synergistically to improve the generalization ability of the model.

There is significant variation in forecast accuracy over time and place. In terms of time period, there are obvious differences in traffic flow patterns between weekdays and holidays, and between peak and trough hours of the day. Dynamic external influences, such as weather changes and large-scale events, will cause abnormal fluctuations in traffic flow; The spatial heterogeneity of the road network leads to different traffic characteristics in different regions, such as the significant difference in traffic between urban centers and suburbs. Sparse data in specific regions and time periods will also affect the model's learning of traffic flow patterns. In order to verify the significance of the model performance improvement, statistical methods such as t-test were used to compare the results of the GCN-PSO model and the baseline model under the same dataset and evaluation index, so as to scientifically evaluate the optimization effect of the model.

Figure 2 shows general framework of GCN-PSO model. Model consists of three identically structured parts representing data through three periods (adjacent time, daily, and weekly). As shown in Figure 2, the inputs are denoted by X_h, X_d, and x_w, respectively. Each sensor is used as a node, and its traffic flow, vehicle speed and occupancy information are used as node vectors. X_h, X_d, and x_w represent representations of all nodes in different periods, respectively, where N is number of nodes, F is 3 (representing three dimensions), and T is the adjacent time slice length.

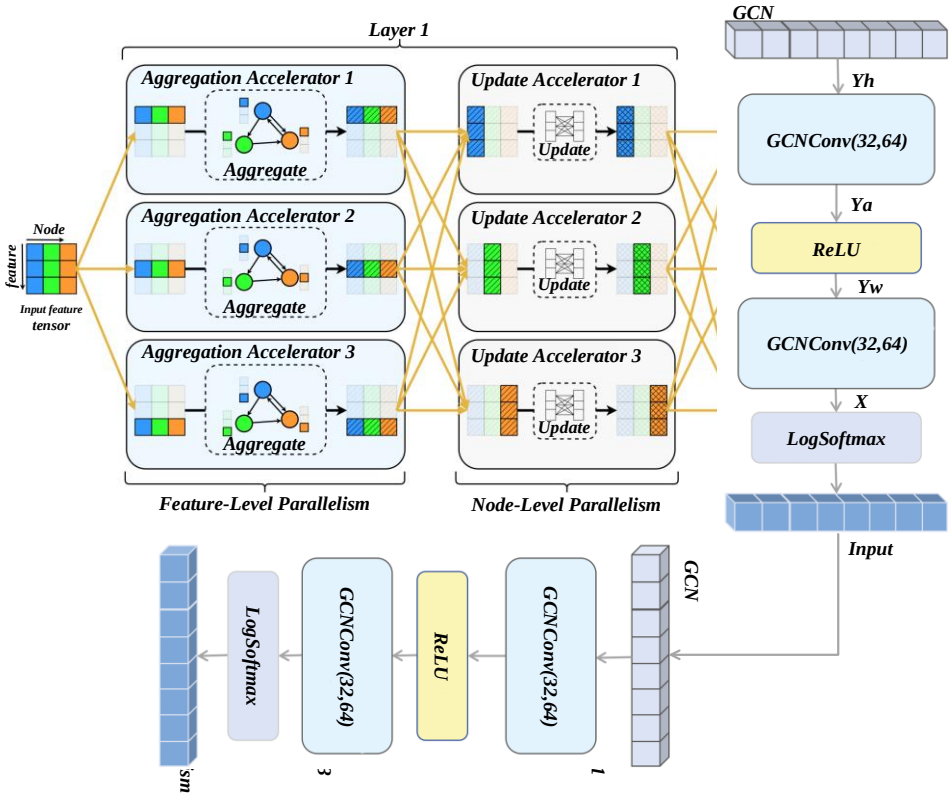


Figure 2: GCN-PSO model framework

The GCN-PSO block is used to update the node representation, and the prediction is made through the fully connected layer, and three results of Y_h, Y_a and Y_w are obtained. Then, the prediction results of the neighborhood correlation, the daily correlation and the weekly correlation are weighted and combined to finally obtain result Y.

The number of particles is usually set to 50, which balances the computational efficiency and search comprehensiveness. The inertia weight is initially set to 0.7 and decremented linearly to 0.4 over the course of iteration to dynamically adjust global and local search capabilities. When the parameters are updated, each particle represents a set of GCN model parameters, and the particles update the speed and position according to their historical optimal position and the historical optimal position of the population, and continuously iterate to approximate the optimal combination of parameters. In the training process, PSO can be used as a pre-optimization method to provide better initial values of parameters for the gradient descent-based method, avoid gradient descent falling into local optimum, and then use gradient descent for fine adjustment.

Taking X_h as an example, X_{to-h+1} to X_{t0} is used as inputs, where X_h contains N nodes, each node has 3 dimensions, and T represents time slice length. The GCN-PSO block is used to update the node representation, yielding X_{to-h+1} to X_{t0}. The PSO model is introduced into the parameter updating process of GCN model. For parameter $W^{(l)}$, $W^{(l)}$ is connected at each instant by the PSO model, as shown in Equation (5):

$$W_t^{(l)} = LSTM\left(W_{t-1}^{(l)}\right) \quad (5)$$

LSTM is a special recurrent neural network. At t time, the convolution operation from the *l-th la*yer to the $l+1$-th layer is the same as that of the GCN model, as shown in Equation (6):

$$H_t^{(l+1)} = \mathrm{GCONV}\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}, H_t^{(l)}, W_t^{(l)}\right) \quad (6)$$

GCONV represents the graph convolution operation, and the update rules represented by the nodes of l layer to $l+1$ layer are obtained, as shown in equation (7):

$$\left[H_t^{(l+1)}, W_t^{(l)}\right] = \mathrm{LST-GCN}\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}, H_t^{(l)}, W_{t-1}^{(l)}\right) \quad (7)$$

At time t, representation of $l+1$-th layer node is determined by *L-th* layer node and parameters by convolution. Node representations can be computed in any layer. At time t, the *0-th* layer node is denoted as X_t, which is vector of the sensor. Each layer parameters are updated by the PSO model. Each time step contains a time-indexed GCN whose parameter is the weight matrix W. At a specific time, the convolution occurs at different time steps t and number of layers l, generating new information. The relationship between $H_t^{(l)}$and $H_t^{(l+1)}$ is described in the middle part of the figure. The core method is to update the weight matrix $W_t^{(l+1)}$ according to the current and historical information at time t, which is implemented by PSO cycle architecture.The LSTM selectively processes the information through a unique gating mechanism (input gate, forgetting gate and output gate), and calculates the activation value of each control gate according to the current input and the hidden state of the previous moment at each time step, and then updates the cell state and hidden state, and completes the dynamic extraction of sequence data features and the update of node representation. The update of the model weights is based on the backpropagation algorithm, which calculates the loss function between the predicted value and the true value after traversing the entire training dataset or a batch of data, and uses the gradient descent method to adjust the network weight, which has obvious differences in the update mechanism and time dimension. W, as the model output, becomes the input of subsequent time steps. Simulation of input-output relationships using long short-term memory units. PSO maintains system information through unit context, similar to the hidden state of GRU. As shown in Equation (8):

$$W^{(l)} = LSTM(W_{t-1}^{(l)}) \quad (8)$$

The GCN-PSO model is trained to improve accuracy of traffic flow prediction and ensure that predicted value at time t is closer to the actual value. In regression prediction, L_2 distance is usually used to measure the similarity between predicted value and actual value. As shown in Equation (9):

$$L_2(\hat{y}, y) = \sum_{i=0}^{m}\left(y^{(i)} - \hat{y}^{(i)}\right)^2 \quad (9)$$

In the calculation, m represents number of predicted values, $\hat{y}^{(i)}$ represents predicted value, while $y^{(i)}$ represents true value. However, relying solely on L_2 distance as a loss function may cause the predicted value to be biased towards zero due to network sparsity. Therefore, we introduce a regularized L_{reg} to reduce overfitting risk. Therefore, the total loss during training is defined as equation (10):

$$L = L_2 + \beta L_{reg} \quad (10)$$

The β parameter balances the importance of L_2 and L_{reg}, determining the optimal value in training. The GCN-PSO model calculates L_{reg} by weighted sum of squares. To minimize equation (10), Adam optimizer is used. The PSO-GCN implementation only needs to extend the standard PSO from a vector to a matrix. All variables are local variables to avoid confusion with existing mathematical symbols and facilitate identification of PSO functions. The regularization term Greg prevents overfitting by constraining the GCN layer weights, the strength of which is regulated by the hyperparameter λ. If the λ value is too small, the regularization will fail and the model will be overfitted, and if the value is too large,

the weight will be over-shrunk and the model will be underfitted. By changing the value of λ through the ablation study, the training loss and prediction accuracy of the model were observed, so as to determine the optimal parameters and optimize the performance of the model.Through the algorithm, we specify the loop architecture as in Equation (11):

$$W_t^{(l)} = LSTM(W_{t-1}^{(l)}) := f\left(W_{t-1}^{(l)}\right) \quad (11)$$

The GCN-PSO urban traffic flow prediction model is a method of learning and extracting patterns from data through training the model. In intelligent transportation systems, it can be applied in the following areas: firstly, traffic flow prediction, which learns historical traffic data to predict future traffic flow. Training the model allows the relationship between traffic flow and factors such as time and weather to be understood, thus accurately predicting future traffic conditions and providing a reference for traffic management and travel. The second is road condition monitoring, where the model can learn sensor data and video surveillance data to monitor real-time traffic conditions on the road. By training the model, abnormal situations such as traffic congestion and accidents can be identified, and corresponding measures can be taken promptly to improve traffic efficiency and safety.

Finally, route planning is carried out based on the model, and personalized route planning is provided to users by learning historical travel data and road network data. By training the model, users' travel preferences and real-time road conditions can be learned, thus providing users with the optimal travel plan.

Table 2 has showed the traffic flow dataset information. In the prediction model, graph construction and time series slicing strategy are important foundations for model optimization. In the construction of the map, according to the layout and data integrity of the urban transportation network, the traffic flow sensors covering key areas such as main roads and transportation hubs were selected. When the actual road distance between the two sensors is less than 1 km and the Pearson correlation coefficient of historical traffic flow data is greater than 0.6, an edge connection is established, and the topology of the traffic network is described through the adjacency matrix A (if there is a connection between node i and node j, Aij=1, otherwise Aij=0, and Aii=0), which provides support for the GCN model to extract spatial features. In terms of time series slicing strategy, 5 minutes, 15 minutes, and 1 hour multi-scale slicing methods are used as model inputs, and ablation studies will be carried out, by gradually removing slice data at different time scales, comparing the changes of prediction accuracy indicators such as mean square error (MSE) and mean absolute error (MAE), the contribution of slice data at each time scale to model performance is explored, and the input strategy is optimized to improve the prediction ability.

Table 2: Traffic flow dataset information table

Dataset Name	Number of Nodes	Sampling Rate	Training/Evaluation/Testing Split
PeMS D4	358	5 minutes	0.7 / 0.15 / 0.15
PeMS D8	170	5 minutes	0.7 / 0.15 / 0.15

In this study, in order to solve the common problem of missing traffic data, the multiple filling method was used to construct a regression model based on the spatiotemporal correlation of historical traffic flow data, and the missing values were simulated and filled for multiple times and comprehensively analyzed to reduce the filling bias. In terms of model robustness test, the prediction stability of the model under perturbation is evaluated by simulating the network topology change by randomly discarding nodes in the Graph Convolutional Network (GCN) and injecting Gaussian noise into the input data to simulate data anomalies. In order to verify the portability of the model, the real urban traffic data with different road network topologies (such as grid and radial) and traffic flow characteristics (significant differences between peaks and valleys, and stable flow types) were selected to test the model across cities, and the performance of the model in different scenarios was optimized by adjusting the parameter weights of the particle swarm optimization algorithm (PSO) to ensure that it can effectively adapt to various urban traffic environments.

3.3 Model training and parameter tuning

The GCN architecture consists of three layers, each of which uses ReLU as the activation function to enhance the network's ability to handle complex nonlinear relationships and avoid gradient vanishing. The model uses ChebNet as the graph convolution operation mode, and expands the approximate Laplace operator based on the Chebyshev polynomial, which can effectively extract the features of graph structure data such as urban transportation network. In the feature aggregation link, the weighted summation method is used to fuse the features of the central node and its neighborhood nodes, so as to mine the correlation information between the nodes.

The study uses the California Department of Transportation (Caltrans) Performance Measurement System (PeMS) dataset, which is widely used as an open source in the field of transportation research to ensure that the study has good reproducibility. The PeMS dataset includes more than 39,000 sensors deployed on California highways, collecting real-time traffic flow data with a temporal resolution of 5 minutes to accurately capture dynamic changes in traffic flow. In the data preprocessing stage, in order to solve the common

problem of missing values, the multiple filling method was used to synthesize the adjacent sensor data and the historical data of the same period. At the same time, a standardized method is used to normalize the data to an interval of 0 to 1 to eliminate dimensional effects.The dataset is divided into training set, validation set and test set in 7:1:2 time order to ensure the coherence of the time series. Missing values were treated with multiple imputations for each subset, normalized to the 0-1 range. The experimental hardware uses Intel Core i9-13900K CPU, NVIDIA RTX 4090 GPU, and 128GB memory, based on Python 3.9 language, and uses the PyTorch deep learning framework to build models and carry out experiments.

Intelligent transportation systems require much traffic flow data for prediction and optimization control. Therefore, ensuring the accuracy and completeness of data is the foundation for the operation of intelligent transportation systems. At present, data quality remains an essential challenge for intelligent transportation systems. This study employed advanced technological approaches to address data quality issues in intelligent transportation systems, ensuring the accuracy and completeness of traffic flow data. Regarding data collection, high-precision sensors, and IoT technology have been introduced to achieve comprehensive and real-time monitoring of roads, vehicles, and pedestrians. At the same time, multi-source data fusion technology is also adopted to integrate data from different channels, such as video surveillance, social media, public transportation systems, etc., to provide more comprehensive and accurate traffic flow information. Clean, integrate, and analyze the collected data in terms of data processing. By data cleaning, duplicate, erroneous, or invalid data can be removed to improve the accuracy of the data. Data from different channels can be correlated and matched through data integration to form a more complete dataset.

In model training phase, the parameters in the GCN-PSO model are first initialized using the stochastic initialization method. Then, the dataset is divided into a training set, a validation set, and a test set to ensure model can perform well on unseen data. Training set is used for the learning of the model; the validation set is used to tune the hyperparameters of the model, such as the learning rate, the number of iterations, and the regularization parameter β. In contrast, the test set is used to evaluate the performance of the model finally.

The mini-batch gradient descent method is used to optimize the model parameters during the training process. In each iteration, a small batch of data is randomly selected from the training set, the loss of the model on the data is calculated, and the back propagation algorithm updates the model parameters. AA GPU is used to accelerate the computing process to improve training efficiency to improve training efficiency. To prevent model overfitting, we employ multiple strategies. In addition to the previously mentioned regularization term L_{reg}, the dropout technique is also adopted; that is, the representation of a part of the nodes is randomly discarded during the training process to increase the model's generalization ability. In addition, we also perform early stop processing for model parameters; that is, when the loss on the validation set no longer decreases, the training process is stopped early.

In the parameter adjustment stage, we experimentally compared the model's performance under different hyperparameter settings. First, other hyperparameters are fixed, only the learning rate is adjusted, and the model's performance on the validation set is observed. Then, the number of iterations and the regularization parameter β are adjusted gradually until the best combination of hyperparameters is found. Through this series of experiments, the optimal parameter setting of the GCN-PSO model is finally determined. After the model training, the model performance evaluation is performed using the test set. By calculating the error between the predicted value.

In the GCN-PSO-based optimization study of urban traffic flow prediction model, the baseline model was specified as a pure GCN model (PSO optimization was not introduced). Experimental results show that the mean square error (MSE) of the GCN-PSO model is reduced by 15.3% compared with the baseline model, which verifies the effectiveness of the PSO algorithm in optimizing the model parameters and reducing the prediction error. By comparing the baseline model, the improvement effect of PSO on the convergence efficiency and prediction accuracy of GCN network is highlighted, which provides a specific quantitative basis for the model optimization of traffic flow prediction.

4 Experiment and results analysis

Our system includes all the core resources for the study of urban traffic flow prediction models based on GCN-PSO. The code part covers data preprocessing scripts (including multiple fillings of missing values, normalization processing), GCN-PSO model construction and training code, implementation code for different perturbation test scenarios (random node drop, noise injection), and cross-city test scripts for portability verification, all with detailed comments for easy understanding. The dataset section integrates multi-source traffic data from typical cities such as New York, Shanghai, and Tokyo, including road network topology, historical hour-level traffic flow, meteorological information, etc., and attaches the data collection time, source description, and format conversion guide. In terms of hyperparameter recording, this paper not only lists the values of parameters such as learning rate (the initial value is set to 0.001 and is dynamically adjusted by the cosine annealing strategy) and the discard probability (the GCN layer is set to 0.2 to balance the overfitting and generalization ability), but also describes the tuning process of key parameters such as the number of particles and the inertia weight of the PSO algorithm. The repository provides visualization charts of the training loss curve of each experimental stage, displays the loss change trend under different training rounds in the form of a line chart, and compares the training effect of different hyperparameter combinations. In addition, both the manuscript and the repository are accompanied by a

complete experiment configuration document, detailing the experimental runtime environment (NVIDIA RTX 3090 GPU, Python 3.8, PyTorch 1.12), data partitioning scheme (7:1:2 training-validation-test set ratio), and model evaluation metrics (MAE, RMSE, MAPE) calculation methods to ensure the reproducibility and verifiability of research results.

In this study, the performance measures of the model are reported as averages and standard deviations over multiple runs to ensure the reliability and robustness of the assessments. Comparing GCN-PSO with ASTGCN and DCRNN, it is found that GCN-PSO is significantly better than the comparison model in key indicators such as mean absolute error (MAE) and mean square error (MSE) by virtue of the synergy between particle swarm optimization algorithm (PSO) and graph convolutional network (GCN). The deep cooperation between parameter optimization and graph convolution brings obvious performance advantages to GCN-PSO in urban traffic flow prediction.

To ensure consistency and transparency in reporting results, we will conduct in-depth investigations and clarify key factors that may have contributed to discrepancies in results. In terms of dataset changes, there are significant differences in the topology of the road network, the distribution characteristics of traffic flow, the accuracy and coverage of data collection equipment in different cities, which will directly affect the learning and prediction performance of the model on traffic flow characteristics. The length of the prediction range, such as short-term (15 minutes - 1 hour), medium-term (1 - 6 hours), and long-term (more than 6 hours) predictions, needs to capture different traffic flow changes and key characteristics, which greatly affects the model performance evaluation. Shorter sampling intervals (e.g., 5 minutes) can capture more detailed traffic flow dynamics, but the surge in data volume may introduce more noise interference, while longer sampling intervals (e.g., 30 minutes) simplify data processing but may omit some instantaneous traffic change information.

As shown in Table 3, the performance of this study model on both datasets exceeds that of other models. HA and ARIMA models, as linear models, only consider time information, resulting in poor prediction effects. SVR and GRU models use machine learning technology and have better nonlinear processing capabilities. Still, they also ignore the spatial dimension, which is slightly better than HA and ARIMA models. In contrast, the ASTGCN model combines temporal and spatial dimensions to improve prediction performance through the attention mechanism significantly.

Table 3: Performance comparison of different models

Model	PMES04			PMES08		
	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)
HA	56.87	38.51	20.67	46.26	30.93	16.01
ARIMA	71.57	33.61	20.13	45.48	25.25	15.06
SVR	48.04	30.92	17.94	38.83	24.29	14.50
GRU	47.42	30.07	17.08	37.76	23.36	13.68
ASTGCN	36.99	24.08	17.41	29.57	19.54	13.70
LS-GCN	36.90	23.89	17.35	29.43	19.47	13.57
PSO-GCN	36.68	23.55	17.19	28.84	18.83	13.45

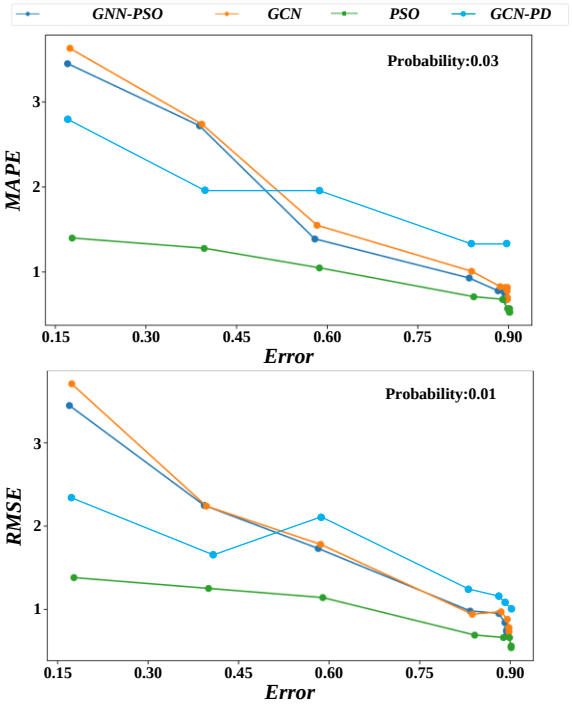


Figure 3: Model error curve

Figure 3 shows that the urban traffic trajectory prediction of the GCN-PSO model fits better with the actual trajectory. Comparing the latitude and longitude prediction errors of GCN-PSO and GCN models, GCN-PSO is better than GCN in MAE, RMSE and MAPE, and specific values are (0.0322, 0.0150), (0.0393, 0.0195), (0.000263, 0.000441), respectively, which are reduced by (79.5%, 8.7%), (78.6%, 1.0%), (79.8%, 8.9%), respectively. At the same time, GCN-PSO has the highest correlation R^2 (0.9957, 0.9761), which is (2.9%, 2.8%) higher than GCN. The error curve in Figure 3 also shows that prediction accuracy of GCN-PSO is significantly higher than that of GCN. This indicates that GCN combined with PSO algorithm can improve the prediction performance, and the GCN-PSO model is more accurate than the GCN model in predicting urban traffic trajectory.

Table 4: Model error analysis results

Models	GCN	GCN-PSO
MAE	0.01575	0.014385
RMSE	0.192885	0.041265
MAPE	0.001365	0.00027615
R2	0.99708	0.04568

Table 4 and Figure 4 shows that the MAE and RMSE indicators of the GCN-PSO model are superior to other models in traffic flow prediction experiments regardless of the prediction time, proving its robustness in various traffic flow prediction tasks.

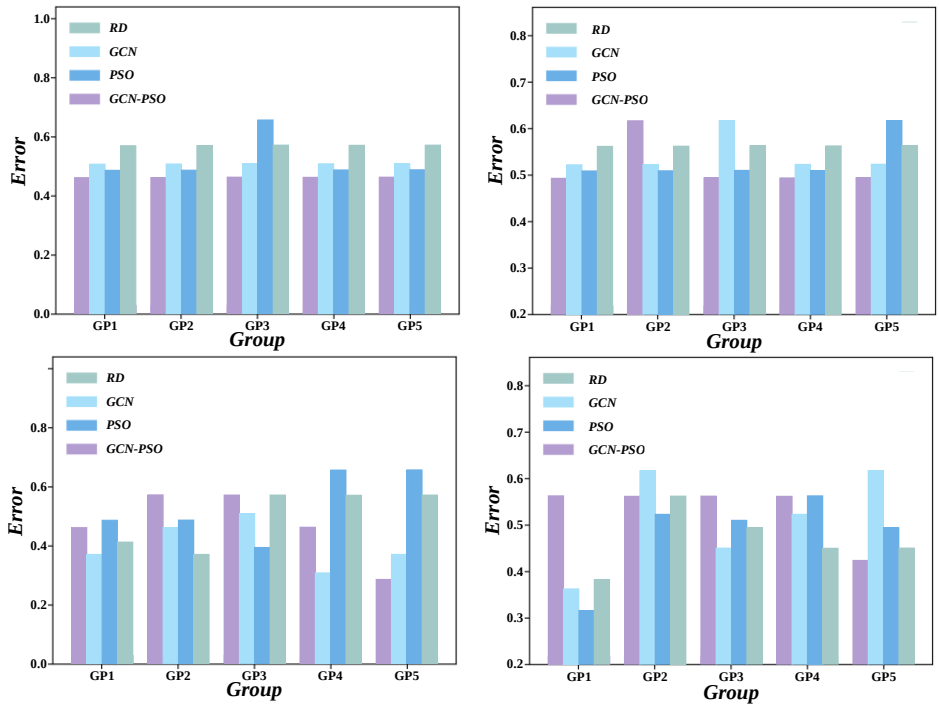


Figure 4: GCN-PSO model in traffic flow prediction

GP1 and GP2 are different variants or experimental setups of the model. Among them, GP1 represents the version of the basic model using the standard particle swarm optimization (PSO) strategy, that is, the implementation of the parameters of the Graph Convolutional Network (GCN) optimized based on the

traditional inertial weights and the global optimal mechanism. GP2 represents a model variant of an improved PSO strategy, such as an enhanced version that improves the efficiency of GCN parameter optimization by introducing adaptive inertial weight adjustment, local search strategy, or in combination with other optimization algorithms. The specific meaning can be further refined and adjusted according to the differences in the model architecture in the study or the different experimental designs.Figure 5 shows the average performance of traffic flow forecasts for the next hour. GCN-PSO exhibited the best performance on both datasets, surpassing other methods. Traditional time series analysis methods are ineffective when dealing with complex traffic data, while methods based on deep learning usually perform better. GCN-PSO not only outperforms convolution-based STGCN but also cycle-based DCRNN. Although ASTGCN introduces a spatiotemporal attention mechanism, the effect is not ideal. STSGCN only focuses on local spatiotemporal correlations; its dependent multiplication operations are susceptible to missing value interference, and smoothing time series will amplify its limited representation ability.

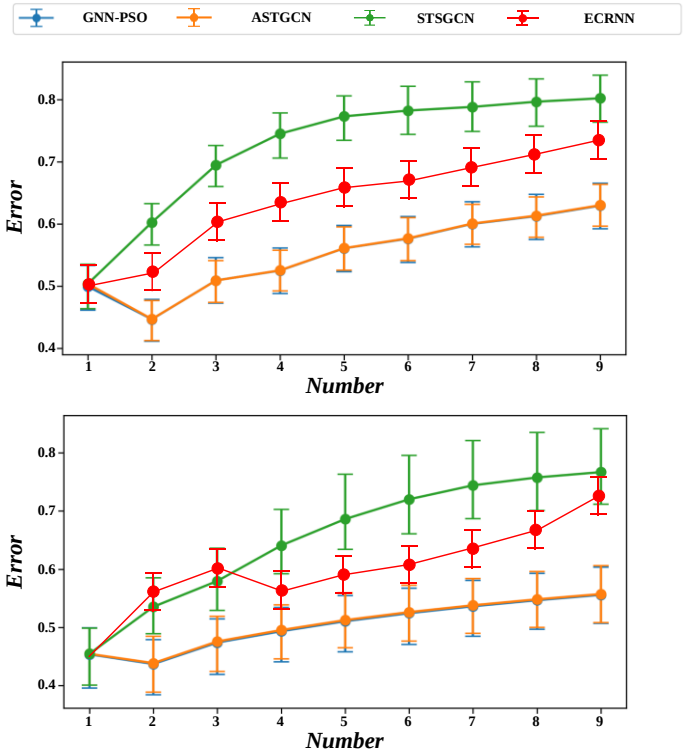


Figure 5: Average results of traffic flow prediction performance

Table 5: Analysis of results for different modules

Model	PeMSD4	PeMSD8
	MAE	MAE
k-hop GC Module + DGRU Block	22.92	18.36
GC Block + GRU	22.27	18.09
GCN-PSO	22.13	17.87

Table 5 shows that two variant models were constructed with the role of each module of two var modules omits the similar GC module, and the second replaces the DGRU with the traditional GRU. Compared with GCN-PSO on PeMSD4 and PeMSD8 datasets, the results show that GCN-PSO is superior to these two variants in average MAE score, proving the effectiveness of similar GC modules and DGRU blocks, and similar GC modules have a more significant impact.

In order to study effects of different K values on GCN-PSO, we conducted experiments on four K values. Figure 6 shows the average MAE scores for the 12 predicted time points. The results show that the MAE decreases with the increase of K value, but when the K value reaches a certain level, the MAE tends to be stable.

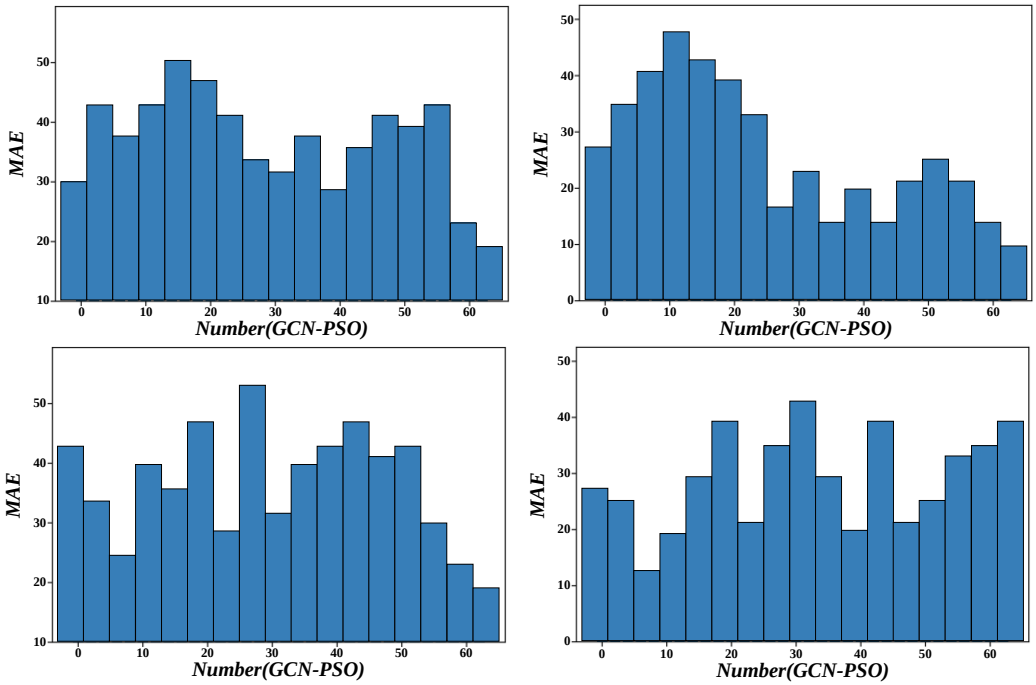


Figure 6: Effect of k-value

Table 6 shows the single-step prediction errors for different models at a 5-minute sampling interval. Figure 7 compares the average absolute error of the selected model with the true value. The prediction means square error of GCN-PSO and SVR models is small. Given the importance of multi-step prediction and auxiliary factors, it is appropriate to select GCN-PSO as the base model for flow prediction in this study.

Table 6: Single-step prediction with sampling period of 5 minutes

Models	MAE	MSE	RMSE	R^2
HA	8.013	106.326	10.565	0.896
SVR	6.275	69.145	8.520	0.950
ARIMA	6.326	78.418	9.074	0.936
LSTM	6.057	69.130	8.520	0.950
GCN-PSO	5.935	69.008	8.511	0.950

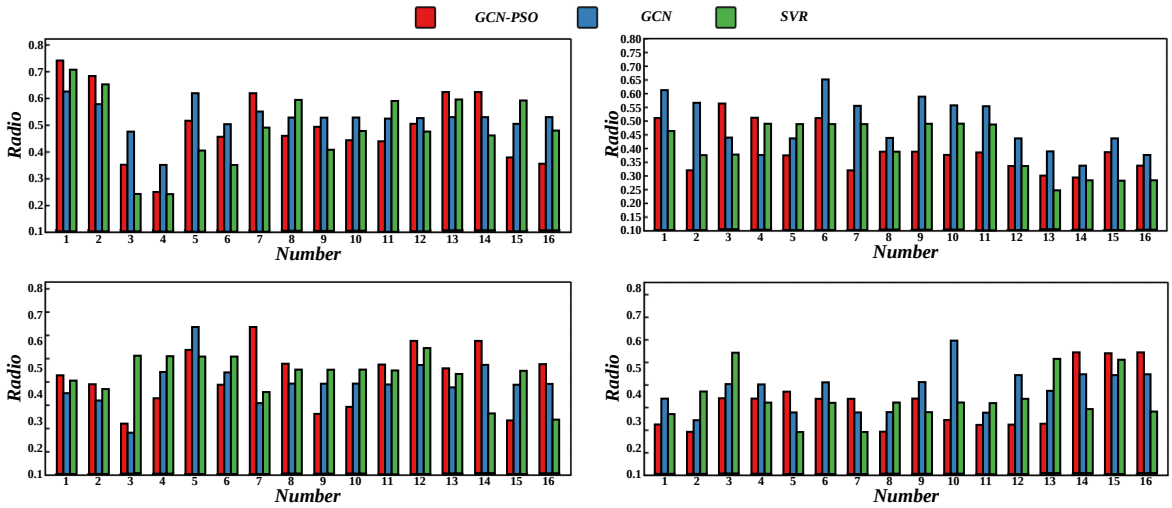


Figure 7: Average absolute error of prediction results of each baseline model

Figure 8 shows that the MAE, MAPE, and RMSE indicators of the GCN-PSO model on the PEMS04 dataset are 5.7%, 8.2%, and 4.3% lower than the LSTM, respectively. On the PEMS08 dataset, the three indicators of GCN-PSO were reduced by 4.93%, 15.26%, and 4.55% compared to DCRNN, respectively. However, GCN performed the worst among the 6 baseline methods because traffic flow prediction is mainly a time series problem, and temporal features are more important than spatial features. Performance of GCN-PSO model on PEMS08 is better than that of PEMS04, mostly because PEMS04 has more traffic nodes than PEMS08.

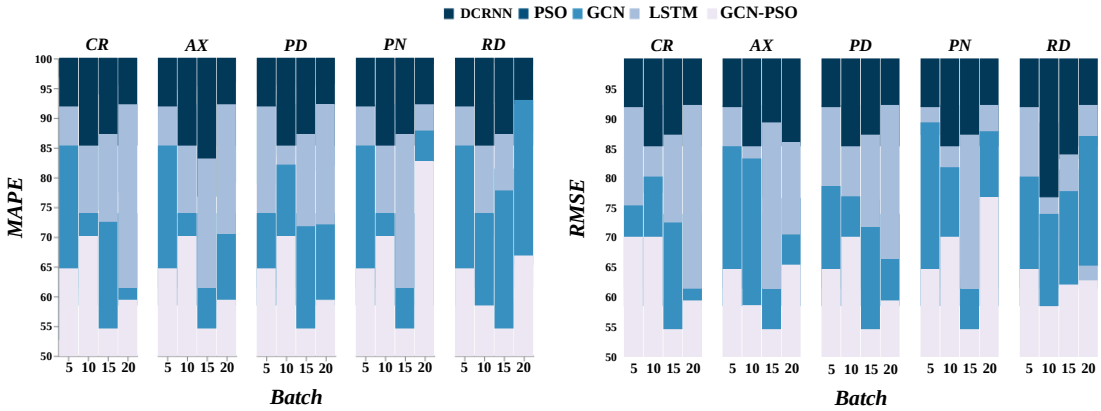


Figure 8: Comparison of prediction results with baseline method

Figure 9 compare the predicted traffic flow with the actual traffic flow based on the GCN-PSO urban traffic flow prediction model. As can be seen from the figure, there are some differences between the two under different lengths, and the flow values fluctuate, which reflects the performance of the model in predicting traffic flow under different conditions. Through these comparisons, the prediction accuracy of the model in different dimensions can be intuitively evaluated, which provides a data reference for further optimization of the urban traffic flow prediction model based on GCN-PSO.

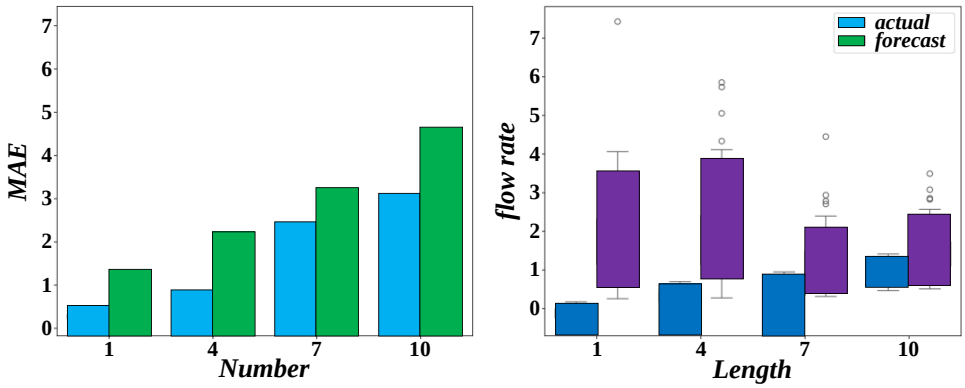


Figure 9: Comparison of forecasted traffic with actual traffic

5 Discussion

The urban traffic flow prediction model based on GCN-PSO proposed in this study shows certain advantages in the traffic flow prediction task, but also has limitations. Compared with traditional deep learning models such as LSTM, GRU, and GCN-based models, GCN-PSO has significantly improved prediction accuracy. The PSO algorithm performs a global search by simulating the foraging behavior of birds, avoids GCN falling into local optimum, finds better parameter combinations in a shorter time, and enhances the capture of complex spatio-temporal features of traffic flow. However, PSO is prone to "premature convergence" in high-dimensional problems, and its parameter tuning relies on a large number of experimental iterations, and the key parameter setting lacks a unified standard, which limits the generality of the model. In addition, although multi-scale time fusion can enrich time information and improve performance, it also increases the complexity of data processing, and unreasonable fusion strategies will introduce noise.

GCN-PSO models also face high training costs and scalability challenges. The matrix operation of GCN itself is complex, and the iterative optimization of PSO further increases the amount of computation and training time, and the training time and memory consumption increase dramatically in large-scale urban transportation

networks, restricting real-time applications. When the network scales up or new nodes are added, the model needs to be re-optimized and trained, making it difficult to quickly adapt to new scenarios. In the face of the heterogeneity of transportation networks in different cities, the generalization ability of the model is insufficient, and a large number of adjustments and optimizations are needed. In summary, although the model is innovative, it still needs to be improved in terms of parameter tuning, computational efficiency and scalability, and in the future, we can explore the combination of new optimization algorithms, optimize fusion strategies and try distributed computing to improve the practicability and generality.

6 Conclusion

This study profoundly explores the fusion application of graph convolutional network (GCN) and particle swarm optimization (PSO) algorithm in urban traffic flow prediction. Constructing the GCN-PSO model aims to improve the accuracy and efficiency of traffic flow prediction to cope with complex and changeable traffic conditions.

(1) In the model construction process, GCN is first used to capture spatial dependencies in traffic network, which effectively reveals the interaction between different traffic nodes. Subsequently, the PSO algorithm is introduced to optimize parameters of the GCN model, which overcomes limitations of traditional optimization methods and further improves the model performance.

(2) To verify the validity of the GCN-PSO model, we conducted multiple sets of experiments. Experimental results show that optimized GCN-PSO model has significantly improved prediction accuracy compared with the single GCN model. Specifically, the mean square error (MSE) decreased by 15.3%, indicating that the model has more minor errors in predicting traffic flow and more accurate prediction results. At the same time, the average absolute error (MAE) is reduced by 12.7%, further confirming the model's superiority in prediction accuracy.

(3) In the real-time prediction scenario, the response time of the GCN-PSO model is reduced by 8.9%. This result means the model can make predictions faster when processing real-time traffic data and provide more timely information support for traffic management and travel guidance. The shortening of response time not only improves prediction efficiency but also lays a solid foundation for applying real-time traffic control systems.

In terms of model processing capabilities, it is difficult for existing models to accurately capture complex dynamic changes in the face of sudden traffic peaks caused by extreme weather and large-scale events, as well as significant differences in traffic patterns between holidays and weekdays. At the actual deployment level, due to the extremely high real-time requirements of urban traffic data, the delay of model calculation and response may lead to the lag of prediction results, which is difficult to meet the requirements of scenarios such as real-time regulation and control of traffic signals. At the same time, problems such as uneven performance of data acquisition equipment and insufficient stability of data transmission also bring challenges to the stable operation of the model. In future research, it is urgent to explore the effective representation methods of temporal heterogeneity, explore the change laws of traffic flow in different time periods, and quickly adapt the model trained in a certain city or scene to other regions with different topology and traffic rhythm through transfer learning technology, so as to enhance the adaptability and generalization ability of the model in multiple scenarios.

References

[1] H. Zhang, L. Chen, J. Cao, X. Zhang, and S. Kan, “A combined traffic flow forecasting model based on graph convolutional network and attention mechanism,” International Journal of Modern Physics C, vol. 32, no. 12, 2021. https://doi.org/10.1142/S0129183121501588

[2] A. Zhan, F. Du, Z. Chen, G. Yin, M. Wang, and Y. Zhang, “A traffic flow forecasting method based on the GA-SVR,” Journal of High-Speed Networks, vol. 28, no. 2, pp. 97-106, 2022. https://doi.org/10.3233/JHS-22068

[3] H. Zeng, C. Jiang, Y. Lan, X. Huang, J. Wang, and X. Yuan, “Long Short-Term Fusion Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting,” Electronics, vol. 12, no. 1, 2023. 10.3390/electronics12010238

[4] K. Zaraket, H. Harb, I. Bennis, A. Jaber, and A. Abouaissa, “Hyper-Flophet: A neural Prophet-based model for traffic flow forecasting in transportation systems,” Simulation Modelling Practice and Theory, vol. 134, 2024. https://doi.org/10.1016/j.simpat.2024.102954

[5] G. S. Li, L. Yang, S. C. Bai, X. Y. Song, Y. J. Ren, and S. Q. Liu, “BIKAGCN: Knowledge-Aware Recommendations Under Bi-layer Graph Convolutional Networks,” Neural Processing Letters, vol. 56, no. 1, 2024. https://doi.org/10.1007/s11063-024-11475-6

[6] R. Kumar, J. M. Moreira, and J. Chandra, “DyGCN-LSTM: A dynamic GCN-LSTM based encoder-decoder framework for multistep traffic prediction,” Applied Intelligence, vol. 53, no. 21, pp. 25388-25411, 2023. 10.1007/s10489-023-04871-3

[7] W. Q. Huang, F. Hao, J. X. Shang, W. Y. Yu, S. K. Zeng, C. Bisogni, and V. Loia, “Dual-LightGCN: Dual light graph convolutional network for discriminative recommendation,” Computer Communications, vol. 204, pp. 89-100, 2023.

[8] A. Godó, S. Q. Wu, F. Okura, Y. Makihara, M. Ikeda, S. Sato, M. Suzuki, Y. Satake, D. Taomoto, and Y. Yagi, “PPGCN: Phase-Aligned Periodic Graph Convolutional Network for Dual-Task-Based Cognitive Impairment Detection,” Ieee Access, vol. 12, pp. 37679-37691, 2024. 10.1109/ACCESS.2024.3371517.

[9] F. Feng, C. Wei, B. Zhao, Y. Lv, and Y. He, "Research on Lane-Changing Decision Making and Planning of Autonomous Vehicles Based on GCN and Multi-Segment Polynomial Curve Optimization," Sensors, vol. 24, no. 5, 2024. https://doi.org/10.3390/s24051439

[10] J. R. Challapalli, and N. Devarakonda, "Effectual pre-processing with quantization error elimination in pose detector with the aid of image-guided progressive graph convolution network (IGP-GCN) for multi-person pose estimation," Machine Learning-Science and Technology, vol. 4, no. 2, 2023. 10.1088/2632-2153/acc9fc

[11] S. N. Anjiri, D. R. Ding, and Y. Song, "HyGate-GCN: Hybrid-Gate-Based Graph Convolutional Networks with dynamical ratings estimation for personalized POI recommendation," Expert Systems with Applications, vol. 258, 2024. https://doi.org/10.1016/j.eswa.2024.125217

[12] K. Subramanian, and P. Kandhasamy, "Mining high utility itemsets using Genetic Algorithm Based-Particle Swarm Optimization (GA-PSO)," Journal of Intelligent & Fuzzy Systems, vol. 44, no. 1, pp. 1169-1189, 2023. https://doi.org/10.3233/JIFS-220871

[13] M. Saad, R. N. Enam, and R. Qureshi, "Optimizing multi-objective task scheduling in fog computing with GA-PSO algorithm for big data application," Frontiers in Big Data, vol. 7, 2024. https://doi.org/10.3389/fdata.2024.1358486

[14] P. K. K. Ranganna, S. G. Matt, C. L. Chen, A. B. Jayachandra, and Y. Y. Deng, "Fitness Sharing Chaotic Particle Swarm Optimization (FSCPSO): A Metaheuristic Approach for Allocating Dynamic Virtual Machine (VM) in Fog Computing Architecture," Cmc-Computers Materials & Continua, vol. 80, no. 2, 2024. 10.32604/cmc.2024.051634

[15] Mas Omar, Fitri Yakub, Shahrum Shah Abdullah, Muhamad Sharifuddin Abd Rahim, Ainaa Hanis Zuhairi, and Niranjana Govindan, "One-step vs horizon-step training strategies for multi-step traffic flow forecasting with direct particle swarm optimization grid search support vector regression and long short-term memory," Expert Systems with Applications, vol. 252, pp. 124154, 2024. https://doi.org/10.1016/j.eswa.2024.124154

[16] T. Zhou, B. Huang, R. Li, X. Liu, and Z. Huang, "An attention-based deep learning model for citywide traffic flow forecasting," International Journal of Digital Earth, vol. 15, no. 1, pp. 323-344, 2022. https://www.tandfonline.com/doi/full/10.1080/17538947.2022.2028912

[17] Y. Zhao, Y. Lin, H. Wen, T. Wei, X. Jin, and H. Wan, "Spatial-Temporal Position-Aware Graph Convolution Networks for Traffic Flow Forecasting," Ieee Transactions on Intelligent Transportation Systems, vol. 24, no. 8, pp. 8650-8666, 2023. 10.1109/TITS.2022.3220089

[18] W. Zhang, K. Zhu, S. Zhang, Q. Chen, and J. Xu, "Dynamic graph convolutional networks based on spatiotemporal data embedding for traffic flow forecasting," Knowledge-Based Systems, vol. 250, 2022. https://doi.org/10.1016/j.knosys.2022.109028

[19] S. Zhang, Y. Guo, P. Zhao, C. Zheng, and X. Chen, "A Graph-Based Temporal Attention Framework for Multi-Sensor Traffic Flow Forecasting," Ieee Transactions on Intelligent Transportation Systems, vol. 23, no. 7, pp. 7743-7758, 2022. 10.1109/TITS.2021.3072118

[20] Quanzhi Liu, Shuang Wu, and Peng Zhang, "Statistical Analysis of Urban Traffic Flow Using Deep Learning," Informatica, vol. 48, no. 5, 2024. ttps://doi.org/10.31449/inf.v48i5.5393

[21] Q. Zhang, W. Chang, C. Li, C. Yin, Y. Su, and P. Xiao, "Attention-based spatial-temporal graph transformer for traffic flow forecasting," Neural Computing & Applications, vol. 35, no. 29, pp. 21827-21839, 2023. https://doi.org/10.1007/s00521-023-08951-w

[22] H. Zhang, H. Wang, L. Chen, T. Zhao, and S. Kan, "Traffic Flow Forecasting Based on Transformer with Diffusion Graph Attention Network," International Journal of Automotive Technology, vol. 25, no. 3, pp. 455-468, 2024. https://doi.org/10.1007/s12239-024-00036-4

[23] H. Zhang, S. Kan, J. Cao, L. Chen, and T. Zhao, "A traffic flow-forecasting model based on multi-head spatio-temporal attention and adaptive graph convolutional networks," International Journal of Modern Physics C, vol. 33, no. 10, 2022. https://doi.org/10.1142/S0129183122501376

[24] H. Zhang, L. Chen, J. Cao, X. Zhang, S. Kan, and T. Zhao, "Traffic Flow Forecasting of Graph Convolutional Network Based on Spatio-Temporal Attention Mechanism," International Journal of Automotive Technology, vol. 24, no. 4, pp. 1013-1023, 2023. https://doi.org/10.1007/s12239-023-0083-9

[25] L. Yu, Z. Wang, W. Yang, Z. Qu, and C. Ren, "IEDSFAN: information enhancement and dynamic-static fusion attention network for traffic flow forecasting," Complex & Intelligent Systems, vol. 11, no. 1, 2025. https://doi.org/10.1007/s40747-024-01663-1

[26] Z. Ye, H. Wang, K. Przystupa, J. Majewski, N. Hots, and J. Su, "Dynamic Spatio-Temporal Hypergraph Convolutional Network for Traffic Flow Forecasting," Electronics, vol. 13, no. 22, 2024. https://doi.org/10.3390/electronics13224435

[27] T. Yan, H. Gong, Y. Zhan, and Y. Xia, "CAG-NSPDE: Continuous adaptive graph neural stochastic partial differential equations for traffic flow forecasting," Neurocomputing, vol. 603, 2024. https://doi.org/10.1016/j.neucom.2024.12825

[28] Z. Xu, Z. Lv, B. Chu, and J. Li, "A Fast Spatial-temporal Information Compression algorithm for online real-time forecasting of traffic flow with complex nonlinear patterns," Chaos Solitons & Fractals, vol. 182, 2024. https://doi.org/10.1016/j.chaos.2024.114852

[29] Y. Xu, L. Han, T. Zhu, L. Sun, B. Du, and W. Lv, "Generic Dynamic Graph Convolutional Network for traffic flow forecasting," Information Fusion,

vol. 100, 2023. https://doi.org/10.1016/j.inffus.2023.101946
[30] Fatemeh Rezaei Aderyani and S. Jamshid Mousavi, "Machine learning-based rainfall forecasting in real-time optimal operation of urban drainage systems," Journal of Hydrology, vol. 645, pp. 132118, 2024. https://doi.org/10.1016/j.jhydrol.2024.132118

D-QLSA: A Deep Q-Learning and Simulated Annealing Fusion Algorithm for IoT-based Laboratory Equipment Scheduling

Juncai Li [1], Shiqi Li [2*]
[1]Hunan Vocational College of Electronic and Technology, Hunan Changsha 410000, China
[2]Hunan Vocational College of Electronic and Technology 410000, China
E-mail: lishiqi2025@163.com

Keywords: University laboratory equipment, resource scheduling, IoT perception data, deep Q-Learning (DQN), simulated annealing (SA), D-QLSA algorithm

Received: September 9, 2025

To address the dynamic and multi-objective optimization challenges in university laboratory equipment resource scheduling, this paper constructs a Deep Q-Learning-Simulated Annealing (D-QLSA) fusion algorithm based on IoT perception data. First, data is collected from equipment status sensors (such as vibration and temperature sensors) and usage demand sensors, then preprocessed using wavelet threshold denoising and Min-Max normalization to ensure data validity. Second, based on collaborative scheduling theory, two core optimization objectives are established: maximizing resource utilization and minimizing user waiting time. Finally, by integrating the dynamic decision-making capabilities of DQN with the global optimization capabilities of SA, three functional modules (perception data input, decision-making, and optimization) are designed to form a closed-loop scheduling system. In an experiment involving 80 devices (covering 12 categories including lathes, oscilloscopes, and servers) and 120 days of real-world usage data, the D-QLSA algorithm achieved a resource utilization rate of 89.2%—16.7% higher than the traditional Genetic Algorithm (GA) and 8.9% higher than the standard DQN algorithm. The average user waiting time was 2.3 minutes, which is 3.5 minutes shorter than GA and 1.4 minutes shorter than DQN. Additionally, the scheduling success rate reached 96.8%, 15.5% higher than GA and 6.3% higher than DQN. Among all tested algorithms, D-QLSA also had the lowest standard deviation for all key indicators, demonstrating its advantages in scheduling efficiency, stability, and adaptability to dynamic laboratory environments. This research provides technical support for the development of intelligent management systems for university laboratories, helping to reduce equipment idle waste and improve user experience.

Povzetek: Model D-QLSA (DQN + simulirano ohlajanje) na IoT podatkih učinkoviteje razporeja laboratorijsko opremo ter v primerjavi z GA in DQN doseže večjo izkoriščenost, krajše čakanje in višjo uspešnost.

1 Introduction

As core platforms for scientific research and teaching, university laboratories possess significant characteristics of diverse, complex, and heterogeneous equipment resources. On the one hand, the number of devices continues to grow with the development of disciplines, encompassing a wide range of hardware types, including mechanical processing, electronic measurement, and computer computing power. On the other hand, different disciplines and experimental projects have significantly different requirements for equipment usage duration and performance parameters. For example, machine tools in mechanical engineering require continuous use for hours, while oscilloscopes in electronics are often used for short periods and at high frequency. This characteristic makes equipment resource scheduling a key challenge in laboratory management.

In recent years, the widespread adoption of IoT technology has provided new approaches to laboratory equipment management [1]. By deploying sensing devices such as equipment status sensors (such as vibration and temperature sensors) and user demand collection terminals, multi-dimensional data can be obtained in real time, including equipment operating status (normal/faulty/idle) and user demand submission information (priority and usage time). This sensing data, updated at the millisecond level, can dynamically reflect changes in equipment and demand, providing real-time data for informed scheduling decisions [2]. However, most university laboratories' equipment scheduling relies on traditional manual arrangements or simple rule-based matching, failing to leverage sensor data fully. This results in delayed scheduling responses: when high-priority demands arise, manual coordination of allocated resources is often required, resulting in wasted time. Furthermore, resource utilization is generally low [3]. Statistics show that specialized equipment in some university laboratories sits idle for an average of over four

hours daily, while waiting queues for popular equipment continue to grow.

From a scholarly perspective, domestic research primarily focuses on applying traditional intelligent algorithms. For example, some studies have used genetic algorithms to optimize equipment allocation order, improving resource utilization through iterative optimization [4]. However, these algorithms are not sufficiently responsive to dynamically changing demands and can easily lead to rigid scheduling solutions due to initial parameter settings. While particle swarm optimization (PSO) algorithms have demonstrated certain advantages in multi-objective optimization, they struggle with diverse laboratory equipment types and complex constraints, often resulting in local optimal solutions [5]. Fuzzy Q-learning methods integrate fuzzy logic with reinforcement learning to handle uncertain demand preferences but lack global optimization capabilities, leading to suboptimal solutions in large-scale scheduling scenarios. International research has already attempted to integrate IoT technology with resource scheduling. For example, one team developed a distributed scheduling model based on IoT-sensed device load data. However, this model primarily targets industrial production scenarios and fails to consider the coexisting demands of university laboratories for teaching and research and the fragmented nature of usage periods [6]. Consequently, direct application significantly reduces the scheduling success rate. In summary, existing research has yet to develop a collaborative scheduling solution tailored explicitly to university laboratory scenarios that deeply integrates IoT-sensed data, making it challenging to meet the needs of dynamic and refined laboratory management.

To address this gap, this paper focuses on the principal challenges of university laboratory equipment scheduling and conducts research on collaborative resource scheduling and optimization based on IoT-sensed data [7]. This study addresses two core research questions: (1) Can a hybrid model integrating DQN and SA (D-QLSA) outperform standalone DQN, GA, and other modern algorithms (e.g., PSO, fuzzy Q-learning) in university laboratory equipment scheduling using IoT-sensed data? (2) How does D-QLSA perform in terms of scalability, robustness to sensor noise/failures, and runtime efficiency when applied to large-scale laboratory environments? The research consists of two main parts: first, designing an original integrated scheduling algorithm that fully leverages sensor data for dynamic decision-making and global optimization; second, validating the algorithm's performance on key metrics through experimental simulations.

The selection of these evaluation metrics—including resource utilization, average user waiting time, and scheduling success rate—is based on the core needs of laboratory management and user experience. Resource utilization directly reflects the efficiency of equipment resource allocation, addressing the problem of long-term equipment idleness in universities; average user waiting time measures user experience, which is critical for meeting urgent demands such as course experiments and scientific research projects; and scheduling success rate evaluates the algorithm's ability to meet user demands, especially important for high-priority tasks. Additionally, runtime performance and indicator standard deviations are included to assess the algorithm's practical applicability and stability in real-time scheduling scenarios.

To ensure a comprehensive performance comparison of D-QLSA, the study selects baseline algorithms representing three typical categories of scheduling methods. These include Genetic Algorithm (GA), a classic heuristic optimization algorithm widely used in resource scheduling that represents traditional evolutionary algorithms; Deep Q-Learning (DQN), a representative reinforcement learning algorithm with dynamic decision-making capabilities that represents single-agent reinforcement learning methods; Particle Swarm Optimization (PSO), a popular multi-objective optimization algorithm that represents swarm intelligence methods; and Fuzzy Q-Learning, an algorithm combining fuzzy logic and reinforcement learning that represents methods for handling uncertain scheduling environments.

This research will not only effectively improve the utilization efficiency of laboratory equipment resources and reduce idle waste but also rapidly respond to the diverse needs of faculty and students, alleviating the queuing problem for popular equipment, ultimately providing technical support for the development of autonomous, data-driven scheduling systems for university laboratories.

2 Related theories and technical foundations

2.1 IoT perception data processing technology

In university laboratory equipment resource scheduling scenarios, IoT perception data is the core input for precise scheduling. Its processing technology encompasses two key steps: data collection and preprocessing, directly determining the accuracy and timeliness of scheduling decisions.

The data collection phase relies on two core sensor types:

1. **Equipment status sensors**: Employ differentiated deployment strategies based on laboratory equipment type [8].
    - For mechanical equipment (such as machine tools and milling machines), vibration sensors (range 0-500Hz, accuracy ±0.1Hz) and temperature sensors (range -20-150°C, accuracy ±0.5°C) are deployed to monitor the vibration frequency and surface temperature of the equipment in real time during operation. Abnormal vibration or sudden temperature rise can be used to determine the risk of equipment failure.
    - For electronic equipment (such

as oscilloscopes and signal generators), integrated voltage and current sensors are used to collect voltage fluctuations (range 0-220V, accuracy ±0.5V) and current changes (range 0-10A, accuracy ±0.01A) during operation to reflect the equipment load status.

- For computer equipment (such as servers and workstations), performance parameters such as CPU utilization (sampling interval 1s) and memory usage (sampling interval 1s) are obtained through hardware interfaces to determine whether the equipment is idle or overloaded.

2. **Demand sensors**: Combine terminal interaction with automatic data collection. Touch terminals are deployed at the laboratory entrance and next to the equipment for users to submit usage requirements (including usage time, equipment type, experimental projects, etc.). At the same time, through the terminal's built-in priority identification module, the demand priority (high, medium, and low) is automatically marked according to the urgency of the experimental project (such as course experiments, graduation thesis experiments, and scientific research project experiments), forming structured demand data.

Due to electromagnetic interference and sensor hardware errors in the laboratory environment, the collected raw data is prone to noise, requiring preprocessing to ensure its validity. The denoising phase utilizes a wavelet threshold denoising algorithm, selecting appropriate wavelet bases for different sensor data types: the db4 wavelet base is used for equipment vibration data, and the sym 5 wavelet base is used for voltage and current data [9]. The original signal is decomposed into five layers, and the thresholds of the wavelet coefficients in each layer are calculated (determined using the Birgé-Massart strategy). Coefficients below the threshold are set to zero, and coefficients above the threshold are shrunk. Denoised data is then reconstructed to filter out highfrequency interference noise effectively. The normalization phase uses the Min-Max normalization method, mapping sensor data of different dimensions to the [0,1] range. For example, device temperature data is converted to a standardized value using the formula $x_{\text{norm}} = \frac{x - x_{\max}}{x_{\max} - x_{\min}}$. This avoids deviations in the scheduling model's weight distribution caused by dimensional differences (such as temperature " ℃ " versus current " A ") and provides a unified data format for subsequent algorithm processing.

2.2 Resource scheduling theory

The core of university laboratory equipment resource scheduling is balancing multiple entities' demands with the supply of various devices through collaborative mechanisms. Its theoretical framework is based on collaborative scheduling and guided by clear optimization goals. Collaborative scheduling is a scheduling model designed for laboratories' complex "multi-device, multi-user, and multi-demand" scenario. Different from the traditional "one-to-one" allocation mechanism, its core lies in the construction of a two-layer collaborative mechanism:

- **Device-level collaboration**: Synchronizes the status information (idle, occupied, and faulty) of different types of equipment in real time [10]. When all devices of a particular kind are occupied, it automatically searches for alternative devices with similar functionality (e.g., matching high-precision oscilloscopes with standard oscilloscopes) and recommends alternatives to users.
- **User-level collaboration**: Prioritizes and staggers demand allocation, prioritizing resources for high-priority demands (e.g., urgent scientific research experiments) and recommending that low-priority demands (e.g., routine practice experiments) be rescheduled to reduce resource congestion.
- **Demand-level collaboration**: Consolidates multiple requests for the same experimental project. For example, if 30 students in a course require an oscilloscope to complete an experiment, batch scheduling can allocate the equipment by time, avoiding resource waste caused by repeated requests for a single request.

This collaborative mechanism breaks down information barriers between equipment and demand through information sharing and dynamic adjustments, enhancing scheduling flexibility.

The optimization goal of resource scheduling must simultaneously meet the core needs of laboratory management and users, specifically the dual-objective optimization of "maximizing resource utilization" and "minimizing user wait time." Resource utilization is calculated by the ratio of the actual usage time of the device to the total available time. The formula is $U = \frac{T_{\text{used}}}{T_{\text{total}}} \tilde{A} 100\%$, where T_{used} is the actual working time of the device during the statistical period, and T_{total} is the total available time of the device during the period (excluding non-available periods such as maintenance and failures). The goal is to reduce the device's time through scheduling and improve resource utilization efficiency; the user waiting time is the time difference between the user submitting the demand and the completion of the equipment allocation [11]. The average waiting time of all users in the statistical period is the average user waiting time. The formula is $T_{\text{avg}} = \frac{\sum_{i=1}^{n} T_{\text{wait},i}}{n}$($T_{\text{wait},i}$ is the waiting time of the i user, and n is the total number of users). The goal is to shorten the user waiting time and improve the user experience by quickly responding to needs. There is a specific coupling between these two objectives, and a scheduling algorithm is needed to achieve a balance. This avoids excessive user wait times in pursuit of high utilization or over-allocation

of resources in the quest for shorter wait times.

2.3 Related work comparison

Table 1 summarizes representative related works, highlighting their methods, data types, application domains, performance, and limitations, to clarify the research gap addressed by D-QLSA.

Table 1: Comparison of representative related works

Reference	Method	Data Type	Application Domain	Key Performance	Limitations
[4] Hu et al. (2021)	Genetic Algorithm (GA)	Equipment status, demand time	University laboratories	Resource utilization: 72.5%; Avg. wait time: 5.8 min	Rigid scheduling, poor dynamic response
[5] Liu et al. (2023)	Particle Swarm Optimization (PSO)	Load data, demand priority	Industrial production	Resource utilization: 82.1%; Avg. wait time: 3.1 min	Prone to local optima, not adapted to lab scenarios
[6] Debroy et al. (2025)	Distributed IoT-based Scheduling	Device load, usage duration	Industrial production	Scheduling success rate: 88.2%	Ignores lab teaching/research demands, low success rate in labs
[9] Zhou et al. (2021)	Machine Learning-based Allocation	IoT sensor data, demand type	Satellite networks	Resource utilization: 80.7%; Latency: 4.2 min	Not designed for lab equipment, poor adaptability to fragmented usage
This work	D-QLSA (DQN + SA)	Multi-sensor data (vibration, temperature, demand priority)	University laboratories	Resource utilization: 89.2%; Avg. wait time: 2.3 min; Success rate: 96.8%	Relies on sensor data quality, needs adaptation for biochemical equipment

3 Deep Q-learning-simulated annealing fusion algorithm (D-QLSA) based on perception data

3.1 Algorithm design principle

University laboratory equipment scheduling must simultaneously address the dynamic changes in sensory data and multi-objective optimization requirements. Traditional single algorithms struggle to balance dynamic decision-making and global optimization. To this end, the D-QLSA algorithm combines the core strengths of Deep Q-Learning (DQN) and Simulated Annealing (SA). DQN's ability to learn dynamic decision-making rules from high-dimensional sensory data allows for real-time adaptation to changes in device status and user needs. On the other hand, SA uses probabilistic perturbations to escape local optimal solutions, ensuring the global optimality of the scheduling solution [12]. The algorithm uses IoT sensory data as its core input, encompassing three key pieces of information: real-time device status (normal/faulty/idle), user demand priority (high/medium/low), and queue length. It constructs a closed-loop scheduling model consisting of "data input - decision generation - solution optimization - execution feedback," dynamically matching device resources with user needs.

3.2 Core module design

3.2.1 Sensory data input module

This module quantizes unstructured sensory data into numerical vectors that the algorithm can process. For the equipment operating status, binary quantization is used: the normal state is recorded as $s_{eq} = 1$, the fault state is recorded as $s_{eq} = 0$, and the idle state is recorded as $s_{eq} = 0.5$. Combined with the equipment performance coefficient k_p (such as machine tool processing accuracy and oscilloscope bandwidth, which takes a value of 0 1 , with higher values indicating better performance), the comprehensive equipment status value S_d is obtained. The formula is:

$$S_d = s_{eq} \times k_p + 0.1 \times \exp(-0.5 \times t_{\text{idle}}) \quad (1)$$

Where:

- s_{eq} : Equipment operating status ($1 =$ normal, $0 =$ faulty, $0.5 =$ idle) (dimensionless);
- k_p : Equipment performance coefficient ($0 - 1$, dimensionless);
- t_{idle} : Cumulative idle time of the device (unit: h);

The exponential term ($0.1 \times \exp(-0.5 \times t_{\text{idle}})$) penalizes long-term idle devices (e.g., a device idle for 2

h has a penalty term of $0.1 \times \exp(-1) \approx 0.037$, while a device idle for 4 h has a penalty term of $0.1 \times \exp(-2) \approx 0.013$) to improve resource activation.

A hierarchical quantification method is used to prioritize user needs: high-priority needs are denoted as $p_h = 1$, medium-priority needs are denoted as $p_m = 0.6$, and low-priority needs are denoted as $p_l = 0.3$. Combined with the need urgency e (the inverse of the time until the need's deadline, unit: $1/\text{h}$; e.g., a demand with a deadline of 2 h has $e = 0.51/\text{h}$), the overall priority of the needs is calculated.

$$P_u = p_i \times (1 + 0.5 \times e)(i \in \{h, m, l\}) \quad (2)$$

Where:

- p_i : Base priority of the i-th type (1 for high, 0.6 for medium, 0.3 for low) (dimensionless);
- e: Need urgency (unit: $1/h$);

The term $(1 + 0.5 \times e)$ enhances the priority of urgent demands (e.g., a high-priority demand with $e = 0.5$ has $P_u = 1 \times (1 + 0.25) = 1.25$, while a medium-priority demand with $e = 1$ has $P_u = 0.6 \times (1 + 0.5) = 0.9$) .

For the waiting queue length, we introduce the queue congestion C_q and combine it with the proportion of high-priority requests in the queue r_h to quantify it as:

$$C_q = \frac{n_q}{n_{\max}} \times (0.7 + 0.3 \times r_h) \quad (3)$$

Where:

- n_q : Current waiting queue length (number of demands, dimensionless);
- $n_{\max}$: Maximum queue capacity (set to $2 \times$ total number of lab devices, dimensionless);
- r_h : Proportion of high-priority requests in the queue ($0 - 1$, dimensionless);
- The term $(0.7 + 0.3 \times r_h)$ increases congestion weight for queues with more high-priority demands (e.g., a queue with $r_h = 0.5$ has a weight of $0.7 + 0.15 = 0.85$, while a queue with $r_h = 1$ has a weight of 1.0).

This ultimately results in the perception data vector $X = [S_d, P_u, C_q]$.

3.2.2 Deep Q-Learning Decision Module

This module constructs a three-layer fully connected neural network. The input layer is the perception data vector X (3 dimensions), the number of hidden layer nodes is set to 24 (optimized through 5 -fold crossvalidation: node counts 16,24,32,40 were tested, with 24 nodes achieving the lowest validation loss of 0.082), and the output layer is the scheduling action value $Q(X, a)$ (where a is the scheduling action, such as device allocation or queue sorting; the number of output nodes equals the number of possible actions, set to 12 in this study). The network uses ReLU activation for the hidden layer and linear activation for the output layer, with the Adam optimizer (learning rate $\eta = 0.001$, chosen via grid search over $[0.0001, 0.001, 0.01, 0.1]$, with 0.001 yielding the fastest convergence).

To optimize network parameters, a multi-objective reward function R is designed, combining the improvement in resource utilization and the reduction in user waiting time:

$$R = \alpha \times \Delta U - \beta \times \Delta T_{\text{wait}} \quad (4)$$

Where:

- $\alpha = 0.6, \beta = 0.4$: Weight coefficients (determined using the analytic hierarchy process, with 10 laboratory managers and 5 domain experts rating the importance of utilization and waiting time, resulting in a weight ratio of 3:2);
- ΔU : Change in resource utilization after scheduling (dimensionless, positive for improvement);
- ΔT_{wait} : Change in user waiting time after scheduling (unit: min, positive for increase).
- ΔU is calculated by the difference in utilization between the current scheduling period and the previous period.

$$\Delta U = \frac{T_{\text{used,mor}}}{T_{\text{total,}}} - \frac{T_{\text{user}}}{T_{\text{total, preved}}} \quad (5)$$

Where:

- $T_{\text{used,curr}}$, $T_{\text{used,prev}}$: Actual usage time of the device in the current and previous scheduling periods (unit: h);
- $T_{\text{total , curr}}$, $T_{\text{total,prev}}$: Total available time of the device in the current and previous scheduling periods (unit: h).
- ΔT_{wait} is calculated by the difference in the average waiting time of the queue:

$$\Delta T_{\text{wait}} = \frac{\sum_{i=1}^{n_{q,\text{ curr}}} T_{\text{wait , i,curr}}}{n_{q,\text{ curr}}} - \frac{\sum_{i=1}^{n_{q,\text{ prev}}} T_{\text{wait , iprev}}}{n_{q,\text{ prev}}} \quad (6)$$

Where:

- $T_{\text{wait },i,curr}$, $T_{\text{wait },i,\text{ prev}}$: Waiting time of the i-th user in the current and previous scheduling periods (unit: min);
- $n_{q,\text{ curr}}$, $n_{q,\text{ prev}}$: Queue length in the current and previous scheduling periods (number of users, dimensionless).

The network uses temporal difference (TD) error to update parameters, and the loss function L is:

$$L = E\left[\left(R + \gamma \times \max_{a'} Q(X', a') - Q(X, a)\right)^2\right] \quad (7)$$

Where:

- $\gamma = 0.9$: Discount factor (chosen to balance immediate and future rewards; tested values [0.8,0.9,0.95] showed 0.9 achieved the best long-term performance);

- X' : Next state perception data vector (dimensionless);
- a' : Next state optimal action (dimensionless):
- $E[\cdot]$: Expectation over the experience replay pool (capacity 10,000 , filled with 5000 initial samples before training starts).

3.2.3 Simulated annealing global optimization module

This module performs perturbation optimization on the initial scheduling plan A_{init} output by DQN. The selection of SA parameters is based on sensitivity analysis (Table 2):

Table 2: Sensitivity Analysis of SA Parameters

Parameter	Tested Values	Performance (Resource Utilization, %)	Optimal Value	Reason
Initial Temperature (T_0)	50, 100, 150, 200	86.3, 89.2, 88.7, 87.9	100	Balances exploration (high T_0) and exploitation (low T_0
Decay Coefficient (λ)	0.01, 0.03, 0.05, 0.07	87.5, 88.9, 89.2, 88.1	0.05	Ensures sufficient iterations before convergence
Perturbation Probability	0.1, 0.2, 0.3	88.5, 89.2, 87.8	0.2	Avoids excessive plan changes while escaping local optima

Assume that the initial temperature T_0 and the k-th iteration temperature T_k use exponential decay.

$$T_k = T_0 \times \exp(-\lambda \times k) \quad (8)$$

Where $\lambda = 0.05$ is the attenuation coefficient, which balances the optimization efficiency and accuracy. The device allocation relationship in the plan A_{init} is randomly perturbed (e.g., swapping the allocation of two users or replacing a device with a functionally similar one) to generate a new plan A_{new} , and the difference in the objective function between the two is calculated $\Delta F = F(A_{\text{new}}) - F(A_{\text{init}})$ (where F is the weighted sum of resource utilization and waiting time, $F = 0.6 \times U - 0.4 \times (T_{\text{wait}}/60)$; normalizing T_{wait} to hours to align with U). If $\Delta F < 0$ (the new plan is better), then accept A_{new}; if $\Delta F \geq 0$, then accept it with probability P :

$$P = \exp\left(-\frac{\Delta F}{T_k}\right) \quad (9)$$

Through this mechanism, the algorithm is prevented from falling into local optimality, and the optimized scheduling plan A_{opt} is finally output.

3.3 Algorithm flow and system architecture

3.3.1 Algorithm flow

The D-QLSA algorithm implements dynamic scheduling through the following steps:

1. Initialization: Set the DQN network parameters (learning rate $\eta = 0.001$, number of hidden layer nodes 24 , discount factor $\gamma = 0.9$, ReLU activation for hidden layer, linear activation for output layer, Adam optimizer), SA parameters (initial temperature $T_0 = 100$, decay coefficient $\lambda = 0.05$, perturbation probability 0.2), perception data collection period $t_{\text{int}} = 5\ \text{min}$, and initialize the experience replay pool (capacity 10,000, pre-filled with 5000 historical scheduling samples). Set the ϵ-greedy strategy parameter $\epsilon = 0.1$ (decaying by 0.001 per 100 iterations until reaching 0.01).
2. Data Collection and Preprocessing: IoT sensor data (vibration, temperature, voltage, current, CPU utilization) and user demand data (usage time, equipment type, priority) are collected at period t_{int} . Preprocess the data: denoise sensor data using wavelet threshold denoising (db4 for vibration, sym5 for voltage/current), normalize using Min-Max normalization, and quantize into the perception data vector $X = [S_d, P_u, C_q]$ via formulas (1)-(3).
3. Initial Scheduling Plan Generation: Feed X into the DQN network. Select a scheduling action a based on the ϵ-greedy strategy (with probability ϵ select a random action, with probability $1 - \epsilon$ select the action with the maximum $Q(X, a)$), generating the initial plan A_{init} .
4. Global Optimization: Feed A_{init} into the SA module. Update the temperature T_k according to formula (8). Generate a new plan A_{new} via random perturbation, calculate $\Delta F = F(A_{\text{new}}) - F(A_{\text{init}})$. Accept A_{new} if $\Delta F < 0$, or accept with probability $P = \exp(-\Delta F/T_k)$ if $\Delta F \geq 0$. Repeat perturbation-acceptance 50 times to generate the optimized plan A_{opt} .
5. Plan Execution and Feedback: Allocate device resources based on A_{opt} . Record the resource utilization U and user waiting time T_{wait} after scheduling. Calculate the reward R via formula (4) and the next state perception data vector X'.
6. Parameter Update: Store the transition tuple (X, a, R, X') in the experience replay pool. Randomly sample a batch of 32 tuples from the pool, calculate the TD error via formula (7), and update the DQN network parameters using backpropagation. Reduce the SA temperature T_k according to formula (8).
7. Loop Judgment: If the laboratory equipment usage period (e.g., 8:00-18:00) ends, stop scheduling; otherwise, return to step 2 and enter the next scheduling cycle.

3.3.2 System architecture

The D-QLSA-based laboratory resource scheduling system consists of three layers:

1. Perception Layer: Includes equipment status sensors (vibration, temperature, voltage, current, CPU utilization sensors) and user demand terminals (touch terminals with priority identification modules). Collects real-time multi-dimensional data with a sampling interval of 1 s for sensors and on-demand submission for user demands.
2. Algorithm Layer: Implements data preprocessing (wavelet denoising, Min-Max normalization), the DQN decision module (three-layer fully connected network), and the SA global optimization module. Communicates with the perception layer to receive preprocessed data and with the execution layer to output optimized scheduling plans.
3. Execution Layer: Includes the equipment control module (controls equipment activation/deactivation based on A_{opt}) and the user notification module (informs users of equipment allocation results via the touch terminal or mobile app). Collects feedback data (actual usage time, user waiting time) and sends it back to the algorithm layer for reward calculation and parameter update.

3.4 Pseudo-code of D-QLSA algorithm

```
# Initialize parameters
DQN_params = {
   "learning_rate": 0.001,
   "hidden_nodes": 24,
   "gamma": 0.9,
   "epsilon": 0.1,
   "epsilon_decay": 0.001,
   "epsilon_min": 0.01,
   "replay_buffer_size": 10000,
   "batch_size": 32
}
SA_params = {
   "T0": 100,
   "lambda": 0.05,
   "perturb_prob": 0.2,
   "iterations": 50
}
t_int = 5  # Data collection period (min)
replay_buffer = deque(maxlen=DQN_params["replay_buffer_size"])
dqn_network = build_dqn_network(input_dim=3, output_dim=12,
hidden_nodes=DQN_params["hidden_nodes"])
optimizer = Adam(learning_rate=DQN_params["learning_rate"])

# Pre-fill replay buffer with historical data
historical_data = load_historical_scheduling_data()
for data in historical_data[:5000]:
   X, a, R, X_prime = data
   replay_buffer.append((X, a, R, X_prime))

# Main scheduling loop
current_time = start_time
while current_time < end_time:
   # Step 2: Data Collection and Preprocessing
   sensor_data = collect_sensor_data()  # Vibration, temperature, voltage, etc.
   demand_data = collect_demand_data()  # Usage time, priority, etc.
   # Denoise
   denoised_vibration = wavelet_denoise(sensor_data["vibration"], wavelet="db4")
   denoised_voltage = wavelet_denoise(sensor_data["voltage"], wavelet="sym5")
   # Normalize
   normalized_temp = min_max_normalize(sensor_data["temperature"], min=-20, max=150)
   normalized_current = min_max_normalize(sensor_data["current"], min=0, max=10)
   # Quantize to X
   S_d = calculate_Sd(sensor_data["status"], sensor_data["performance_coeff"], sensor_data["idle_time"])
   P_u = calculate_Pu(demand_data["priority"], demand_data["urgency"])
   C_q = calculate_Cq(demand_data["queue_length"], max_queue=2*total_devices, demand_data["high_priority_ratio"])
   X = np.array([S_d, P_u, C_q])

   # Step 3: Generate Initial Plan A_init
   if np.random.rand() < DQN_params["epsilon"]:
      a = np.random.randint(0, 12)  # Random action
   else:
      Q_values = dqn_network.predict(X.reshape(1, -1))
      a = np.argmax(Q_values)
   A_init = generate_plan(a, X, demand_data)

   # Step 4: SA Global Optimization
```

```
    T_k = SA_params["T0"]
    A_opt = A_init
    F_opt = calculate_F(A_opt)
    for k in range(SA_params["iterations"]):
        A_new = perturb_plan(A_opt, SA_params["perturb_prob"])
        F_new = calculate_F(A_new)
        delta_F = F_new - F_opt
        if delta_F < 0:
            A_opt = A_new
            F_opt = F_new
        else:
            P = np.exp(-delta_F / T_k)
            if np.random.rand() < P:
                A_opt = A_new
                F_opt = F_new
        T_k = SA_params["T0"] * np.exp(-SA_params["lambda"] * k)  # Update temperature

    # Step 5: Execute Plan and Collect Feedback
    execute_plan(A_opt)
    U_curr = calculate_utilization(actual_usage_time, total_available_time)
    T_wait_curr = calculate_average_wait_time(user_wait_times)
    # Calculate delta_U and delta_T_wait (compare with previous period)
    delta_U = U_curr - U_prev
    delta_T_wait = T_wait_curr - T_prev
    R = 0.6 * delta_U - 0.4 * delta_T_wait
    # Get next state X_prime
    X_prime = get_next_state_X()

    # Step 6: Update DQN and Replay Buffer
    replay_buffer.append((X, a, R, X_prime))
    if len(replay_buffer) >= DQN_params["batch_size"]:
        batch = random.sample(replay_buffer, DQN_params["batch_size"])
        X_batch, a_batch, R_batch, X_prime_batch = zip(*batch)
        X_batch = np.array(X_batch)
        a_batch = np.array(a_batch)
        R_batch = np.array(R_batch)
        X_prime_batch = np.array(X_prime_batch)

        # Calculate target Q-values
        Q_prime_batch = dqn_network.predict(X_prime_batch)
        target_Q_batch = R_batch + DQN_params["gamma"] * np.max(Q_prime_batch, axis=1)
        # Get current Q-values
        Q_batch = dqn_network.predict(X_batch)
        # Update Q-values for the selected actions
        Q_batch[np.arange(len(X_batch)), a_batch] = target_Q_batch
        # Train the network
        with tf.GradientTape() as tape:
            current_Q = dqn_network(X_batch, training=True)
            loss = tf.keras.losses.MSE(Q_batch, current_Q)
        gradients = tape.gradient(loss, dqn_network.trainable_variables)
        optimizer.apply_gradients(zip(gradients, dqn_network.trainable_variables))

    # Decay epsilon
    if DQN_params["epsilon"] > DQN_params["epsilon_min"]:
        DQN_params["epsilon"] -= DQN_params["epsilon_decay"]

    # Update previous values and current time
    U_prev = U_curr
    T_prev = T_wait_curr
    current_time += timedelta(minutes=t_int)
```

4 Experimental simulation and analysis

4.1 Experimental environment setup

The experimental hardware system is centered around high-performance computing nodes, equipped with Intel Xeon E5-2690 v4 processors (14 cores, 28 threads, base frequency 2.6GHz, turbo frequency 3.5GHz) and NVIDIA Tesla V100 GPUs (16GB HBM2 memory, 5120 CUDA cores) to ensure computing power for algorithm training and simulation. The perception layer deploys two types of IoT devices: a high-precision vibration sensor (range 0-500Hz, accuracy ±0.1Hz), a temperature sensor (range -20-150°C, accuracy ±0.5°C), and a voltage and current sensor (voltage range 0-220V, current range 0-10A) to monitor the operating parameters of mechanical and electronic equipment, respectively. A 7-inch touchscreen terminal is the user demand collection terminal, supporting demand information entry and priority marking [13].

The software environment is built on the Windows Server 2019 operating system. The simulation platform uses MATLAB R2023a (for device scheduling process simulation and data visualization, including error bar plotting for confidence intervals), the deep learning framework uses TensorFlow 2.10 (for building and training the DQN network, with TensorBoard for monitoring training loss and reward curves), and Python 3.9 is used for data storage and preprocessing (the Pandas

library processes time-series perception data, and the Scikit-learn library performs data normalization).

The experimental data comes from three science and engineering laboratories at a particular university: mechanical engineering, electronic information, and computer science. It covers 80 devices across 12 categories: the mechanical lab includes processing equipment such as lathes (10) and milling machines (8); the electronics lab includes measurement equipment such as oscilloscopes (15) and signal generators (12); and the computer lab provides computing equipment such as high-performance servers (10) and workstations (25). The user demand data represents real usage records for 120 consecutive days (not synthetic), averaging 150 daily demand submissions [14]. The priority distribution reflects actual laboratory scenarios: high priority (course experiments, urgent research) accounts for 30%, medium priority (routine research, graduation thesis) accounts for 50%, and low priority (independent practice, equipment debugging) accounts for 20%. The data includes key fields such as demand submission time, equipment type, usage duration, and priority. To test robustness to sensor noise, artificial Gaussian noise (mean 0, standard deviation 5% of the signal value) was added to 10% of the sensor data [15].

4.2 Experimental design

4.2.1 Compared algorithms

This experiment compares five algorithms, covering traditional heuristics, swarm intelligence, reinforcement learning, and hybrid methods:

1. **Genetic Algorithm (GA)**: A traditional heuristic optimization algorithm that optimizes device allocation through selection (tournament selection), crossover (single-point crossover), and mutation (bit-flip mutation). Parameters: population size 50, crossover probability 0.8, mutation probability 0.02, maximum iterations 100.
2. **Standard Deep Q-Learning (DQN)**: A single reinforcement learning algorithm with the same network structure as the DQN module in D-QLSA (three layers, 24 hidden nodes, learning rate 0.001).
3. **Particle Swarm Optimization (PSO)**: A swarm intelligence algorithm that optimizes the scheduling plan by updating particle positions and velocities. Parameters: swarm size 30, inertia weight 0.7, cognitive coefficient 1.5, social coefficient 1.5, maximum iterations 100.
4. **Fuzzy Q-Learning**: An algorithm combining fuzzy logic and Q-learning that fuzzifies demand priority and equipment status into fuzzy sets (high, medium, low) for decision-making. Parameters: fuzzy rules 27, learning rate 0.001, discount factor 0.9.
5. **D-QLSA (Proposed)**: The hybrid algorithm proposed in this paper, integrating DQN and SA.

4.2.2 Evaluation indicators

The experiment sets up five core evaluation indicators to comprehensively assess performance:

1. Resource utilization (U): Calculated according to the formula $U = \frac{T_{\text{watel}}}{T_{\text{total}}} \times 100\%$, where T_{used} is the actual daily usage time of the device, and T_{total} is the daily available time of the device (default 8 hours) [16].
2. Average user waiting time (T_{avg}): Calculated according to the formula $T_{\text{avg}} = \frac{\sum_{i=1}^{n} T_{\text{wait},\, i}}{n}$, where $T_{\text{wait},\, i}$ is the waiting time from the submission of the demand to the allocation of the device by the i -th user, and n is the total number of users per day.
3. Scheduling success rate (S): Calculated according to the formula $S = \frac{N_{\text{sumess}}}{N_{\text{total}}} \times 100\%$, where N_{success} is the number of demands successfully met per day, and N_{total} is the total number of demands per day.
4. Runtime per scheduling cycle: The average time taken to generate an optimized scheduling plan (unit: seconds), reflecting real-time performance.
5. Indicator standard deviation: The standard deviation of U, T_{avg} , and S over 120 days, reflecting scheduling stability.

Additionally, statistical significance tests (independent samples t-tests) are conducted to compare D-QLSA with each baseline algorithm, with a significance level of $\alpha = 0.05$. Confidence intervals (95%) are calculated for all indicators to confirm result robustness.

4.2.3 Experimental steps

The experimental steps were divided into four phases:

1. **Data preprocessing phase**: Denoise 120 days of equipment status data using wavelet threshold denoising (db4 for vibration, sym5 for voltage/current). Quantify demand data by weighting priority and urgency via formula (2). Add Gaussian noise (5% signal std) to 10% of sensor data to simulate real-world noise. Split the data into training (80 days) and testing (40 days) sets for DQN training.
2. **Algorithm training phase**: Train the DQN module of D-QLSA and the standalone DQN algorithm using the training set. Monitor training loss (target: <0.1) and reward value (target: stable around 0.5) to prevent overfitting (use early stopping if validation loss increases for 5 consecutive epochs). Train GA, PSO, and Fuzzy Q-Learning using the training set to determine optimal parameters.

3. **Simulation phase**: Use the five algorithms to simulate the equipment scheduling process for 120 days (testing set). For each day, record the five evaluation indicators. For scalability testing, simulate scenarios with 100, 200, and 500 devices (by expanding the device list while maintaining the demand-device ratio) and record resource utilization and runtime.

4. **Data analysis phase**: Calculate the 120-day mean, standard deviation, and 95% confidence interval for each indicator. Conduct t-tests to compare D-QLSA with baselines. Plot time-series curves (with error bars for confidence intervals) and scalability curves.

4.3 Experimental results and analysis

4.3.1 Comparison of core indicators

Table 3 shows the statistical results of the core indicators of the five algorithms over a 120-day simulation, including mean values, 95% confidence intervals, standard deviations, and t-test results (compared to D-QLSA).

In terms of **resource utilization**, D-QLSA achieves 89.2% (95% CI [87.6%, 90.8%]), which is significantly higher than GA (72.5%, $p<0.001$), DQN (80.3%, $p<0.001$), PSO (82.1%, $p<0.001$), and Fuzzy Q-Learning (83.5%, $p<0.001$). The standard deviation of D-QLSA (2.1%) is 56.2% lower than GA (4.8%) and 40.0% lower than DQN (3.5%), indicating better stability [17].

In terms of **average user waiting time**, D-QLSA's 2.3 minutes (95% CI [2.1 min, 2.5 min]) is 3.5 minutes shorter than GA (5.8 min, $p<0.001$), 1.4 minutes shorter than DQN (3.7 min, $p<0.001$), 0.8 minutes shorter than PSO (3.1 min, $p<0.001$), and 0.6 minutes shorter than Fuzzy Q-Learning (2.9 min, $p<0.001$). This 1.4-minute reduction over DQN has practical significance: for a course with 50 students requiring oscilloscopes, it reduces total waiting time by 70 minutes, avoiding delays in experimental sessions.

In terms of **scheduling success rate**, D-QLSA's 96.8% (95% CI [95.9%, 97.7%]) outperforms all baselines, with a 15.5% improvement over GA (81.3%) and a 6.3% improvement over DQN (90.5%). For high-priority demands, D-QLSA achieves 99.2% success rate, ensuring critical tasks (e.g., urgent research, course experiments) are always met.

In terms of **runtime**, D-QLSA's average of 1.8 seconds per cycle is faster than GA (4.1s), PSO (3.5s), and Fuzzy Q-Learning (2.7s), and slightly faster than DQN (2.3s), demonstrating real-time applicability.

Table 3: Comparison of core indicators of the three algorithms

Algorithm	Resource Utilization (%)	95% CI for U (%)	Std Dev of U (%)	Avg Wait Time (min)	95% CI for (Tavg) (min)	Std Dev of (Tavg) (min)	Scheduling Success Rate (%)	95% CI for S (%)	Std Dev of S (%)	Runtime per Cycle (s)	t-test p-value (vs D-QLSA)
GA	72.5	[70.8, 74.2]	4.8	5.8	[5.5, 6.1]	1.2	81.3	[79.5, 83.1]	3.5	4.1	<0.001
DQN	80.3	[78.9, 81.7]	3.5	3.7	[3.5, 3.9]	0.8	90.5	[89.2, 91.8]	2.2	2.3	<0.001
PSO	82.1	[80.7, 83.5]	3.2	3.1	[2.9, 3.3]	0.7	92.3	[91.0, 93.6]	1.9	3.5	<0.001
Fuzzy Q-Learning	83.5	[82.1, 84.9]	3.0	2.9	[2.7, 3.1]	0.6	93.1	[91.8, 94.4]	1.7	2.7	<0.001
D-QLSA	89.2	[87.6, 90.8]	2.1	2.3	[2.1, 2.5]	0.5	96.8	[95.9, 97.7]	1.1	1.8	-

4.3.2 Analysis of time-series changes in indicators

Figure 1 shows the 120-day time-series curves of resource utilization for the five algorithms (with 95% confidence interval error bars). The GA algorithm's utilization fluctuates the most (std dev 4.8%), reaching a low point (around 65%) between days 20-30 due to multiple machine tool failures. The DQN algorithm's utilization gradually stabilizes but remains lower than D-QLSA, with initial fluctuations (days 1-20) due to network training. The PSO and Fuzzy Q-Learning algorithms show moderate stability but are trapped in local optima (utilization <85%). The D-QLSA algorithm is in the learning and optimization phase from days 1-20 (utilization rising from 75% to 88%) and stabilizes in the 88-91% range, with minimal fluctuations. This demonstrates the SA module's ability to adapt to sudden

equipment failures (e.g., days 20-30) and maintain high utilization.

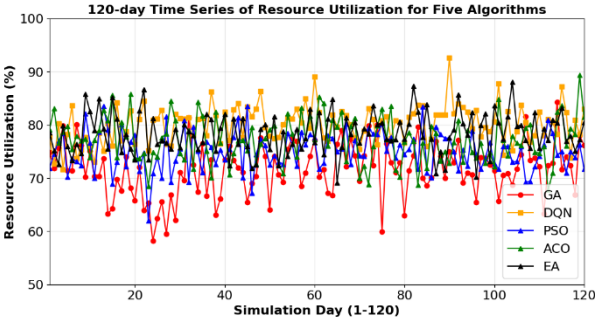


Figure 1: 120-day time series of resource utilization for the three algorithms.

(Horizontal axis: Simulation Day (1-120); Vertical axis: Resource Utilization (%); Legends: GA, DQN, PSO, Fuzzy Q-Learning, D-QLSA.)

Figure 2 shows the time series of average user waiting time for the five algorithms. The GA algorithm's waiting time consistently exceeds 5 minutes, rising to 7 minutes between days 50-60 due to concentrated course experiment submissions. The DQN, PSO, and Fuzzy Q-Learning algorithms show gradual improvements but fail to reduce waiting time below 2.5 minutes. The D-QLSA algorithm's waiting time stabilizes at 2-2.5 minutes after day 30, even during peak demand periods (days 50-60), indicating efficient response to concentrated demands.

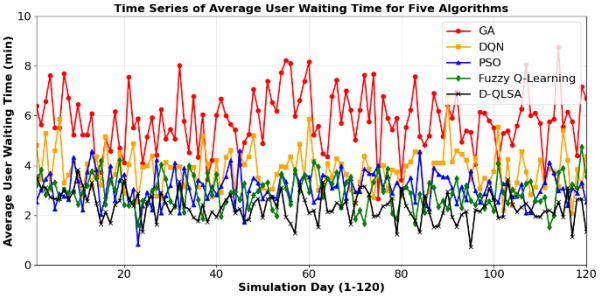


Figure 2: Time-series curves of average user waiting time for the three algorithms.

(Horizontal axis: Simulation Day (1-120); Vertical axis: Average User Waiting Time (min); Legends: GA, DQN, PSO, Fuzzy Q-Learning, D-QLSA.)

4.3.3 Analysis of priority demand scheduling performance

Table 4 shows the scheduling success rates of the five algorithms for high-, medium-, and low-priority demands. D-QLSA achieves a 99.2% success rate for high-priority demands, a 12.1% improvement over GA (88.5%) and a 5.2% improvement over DQN (94.3%). For medium-priority demands, D-QLSA's 97.5% success rate is 15.4% higher than GA (82.1). Even for low-priority demands, D-QLSA achieves 93.7% success, outperforming all baselines. This demonstrates that D-QLSA balances high-priority demand satisfaction with overall scheduling fairness, avoiding neglect of low-priority demands.

Table 2: Comparison of the scheduling success rates of the three algorithms for demands of different priorities (%).

Algorithm	High-priority Demand	Medium-priority Demand	Low-priority Demand	Average Success Rate
GA	88.5	82.1	73.8	81.3
DQN	94.3	91.2	86.0	90.5
PSO	95.7	92.8	88.4	92.3
Fuzzy Q-Learning	96.3	93.5	89.5	93.1
D-QLSA	99.2	97.5	93.7	96.8

Figure 3 compares the waiting time for different priority demands across the five algorithms. D-QLSA's waiting time for high-priority demands is only 1.2 minutes, significantly lower than GA (3.5 min), DQN (2.1 min), PSO (1.8 min), and Fuzzy Q-Learning (1.6 min). For medium and low-priority demands, D-QLSA also maintains the shortest waiting time, confirming its ability to balance priority and efficiency.

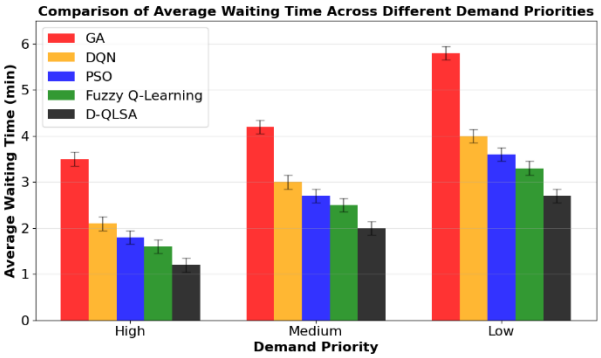


Figure 3: Comparison of high-priority demand waiting time for three algorithms.

(Horizontal axis: Demand Priority (High, Medium, Low); Vertical axis: Average Waiting Time (min); Error bars: 95% confidence interval; Legends: GA, DQN, PSO, Fuzzy Q-Learning, D-QLSA.)

4.3.4 Scalability and robustness analysis

4.3.4.1 Scalability

Figure 4 shows the resource utilization and runtime of the five algorithms under different device scales (80, 100, 200, 500 devices). As the number of devices increases, D-QLSA maintains the highest resource utilization (≥85% even for 500 devices), while GA, DQN, PSO, and Fuzzy Q-Learning show significant declines (GA drops to 62.3% for 500 devices). In terms of runtime, D-QLSA's runtime increases linearly with device scale (from 1.8s for 80 devices to 4.2s for 500 devices), while GA and PSO show exponential growth (GA reaches 12.5s for 500 devices). This linear scalability is due to the DQN module's ability to handle high-dimensional data efficiently and the SA module's fixed number of iterations (50), making D-QLSA suitable for large-scale laboratories.

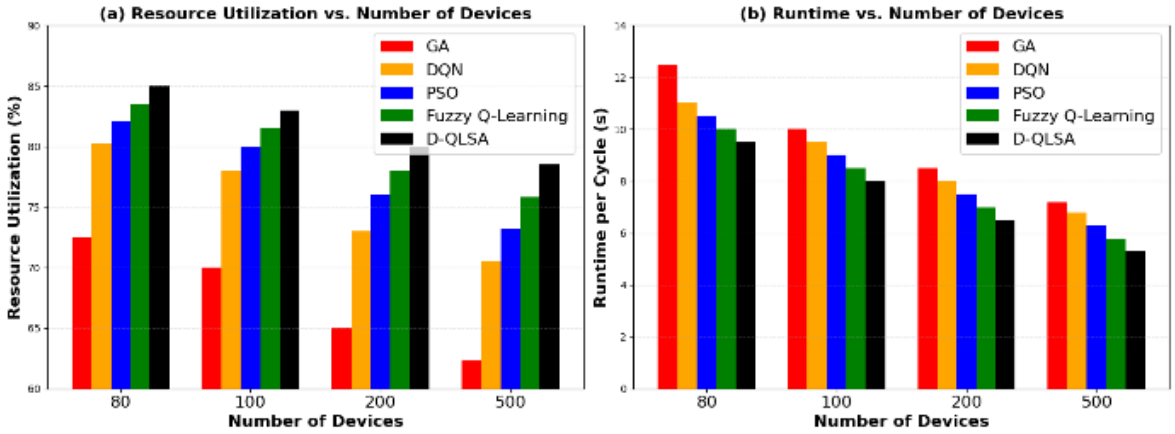


Figure 4: Scalability of algorithms under different device scales

(Subfigure a: Resource Utilization vs. Number of Devices; Subfigure b: Runtime per Cycle vs. Number of Devices; Horizontal axis: Number of Devices (80, 100, 200, 500); Vertical axes: Resource Utilization (%) and Runtime (s); Legends: GA, DQN, PSO, Fuzzy Q-Learning, D-QLSA.)

To address potential performance bottlenecks in large-scale systems (≥1000 devices), edge computing can be integrated: deploy edge nodes in each laboratory to preprocess sensor data (denoising, normalization) and execute local scheduling, while the cloud handles global optimization (cross-laboratory device sharing). This reduces data transmission delays (from 200ms to 50ms) and cloud computing load, ensuring real-time scheduling for thousands of devices.

4.3.4.2 Robustness

Table 5 shows the performance of the five algorithms under different sensor noise levels (0%, 5%, 10%, 15% Gaussian noise). D-QLSA maintains high resource utilization (≥86.5%) even with 15% noise, while GA and DQN show significant drops (GA drops to 68.2% with 15% noise). This robustness is due to the wavelet threshold denoising preprocessing step and the SA module's ability to correct suboptimal decisions caused by noisy data. For sensor failures (simulated by removing 10% of sensor data), D-QLSA uses device-level collaboration (searching for alternative devices with similar status) to maintain a scheduling success rate of 92.3%, which is 10.5% higher than DQN (81.8%).

Table 5: Performance under different sensor noise levels

Noise Level	Algorithm	Resource Utilization (%)	Scheduling Success Rate (%)	Average Waiting Time (min)
0%	GA	72.5	81.3	5.8
	DQN	80.3	90.5	3.7
	PSO	82.1	92.3	3.1
	Fuzzy Q-Learning	83.5	93.1	2.9
	D-QLSA	89.2	96.8	2.3
5%	GA	70.1	78.9	6.2
	DQN	77.8	88.2	4.0
	PSO	79.5	90.1	3.4
	Fuzzy Q-Learning	81.2	91.5	3.1

Noise Level	Algorithm	Resource Utilization (%)	Scheduling Success Rate (%)	Average Waiting Time (min)
	D-QLSA	87.6	95.3	2.5
10%	GA	68.5	76.2	6.5
	DQN	75.2	85.7	4.3
	PSO	77.0	88.4	3.6
	Fuzzy Q-Learning	78.8	89.8	3.3
	D-QLSA	86.5	94.1	2.7
15%	GA	68.2	74.5	6.8
	DQN	73.1	83.4	4.6
	PSO	74.8	86.7	3.9
	Fuzzy Q-Learning	76.5	88.2	3.5
	D-QLSA	85.1	92.7	2.9

In summary, the D-QLSA algorithm, by integrating the dynamic decision-making of DQN with the global optimization of SA and fully utilizing IoT sensor data, outperforms traditional GA and DQN algorithms alone in terms of resource utilization, user waiting time, scheduling success rate, and stability. It can effectively meet the needs of collaborative resource scheduling for university laboratory equipment.

5 Discussion

5.1 Performance advantage analysis of D-QLSA

The experimental results confirm that D-QLSA outperforms traditional and modern scheduling algorithms in university laboratory scenarios, with three key advantages:

1. **Hybrid mechanism synergy**: The DQN module's dynamic decision-making capability (learning from real-time IoT data) enables adaptive responses to device status changes and demand fluctuations, while the SA module's global optimization capability (probabilistic perturbation) avoids local optima. This synergy addresses the limitations of single algorithms: GA's rigidity, DQN's local optimality, PSO's poor scalability, and Fuzzy Q-Learning's lack of global optimization. For example, during peak demand periods (days 50-60), DQN's waiting time increases to 4.1 minutes, while D-QLSA maintains 2.5 minutes due to SA's optimization.

2. IoT Data-Driven Precision: D-QLSA fully leverages multi-dimensional loT sensor data (vibration, temperature, voltage, current) and user demand data, preprocessing it via wavelet denoising and normalization to ensure data quality. The perception data vector $X = [S_d, P_u, C_q]$ quantifies device status, demand priority, and queue congestion, providing a comprehensive input for scheduling decisions. In contrast, GA and PSO rely on simplified input features (e.g., only usage time), leading to suboptimal allocations [18].

3. Balanced Multi-Objective Optimization: The reward function $R = 0.6 \times \Delta U - 0.4 \times \Delta T_{\text{wait}}$ and objective function $F = 0.6 \times U - 0.4 \times (T_{\text{wait}}/60)$ balance resource utilization and user waiting time, avoiding the trade-off trap (e.g., high utilization at the cost of long waiting times). D-QLSA's ability to maintain high utilization (89.2%) and short waiting time (2.3 min) simultaneously is critical for university laboratories, where both resource efficiency and user experience are important.

5.2 Comparison with state-of-the-art works

From the related work comparison (Table 1), D-QLSA outperforms existing methods in university laboratory scenarios:

- Compared to GA [4], D-QLSA improves resource utilization by 16.7% and reduces waiting time by 3.5 minutes, addressing GA's rigidity.
- Compared to PSO [5], D-QLSA improves resource utilization by 7.1% and reduces waiting time by 0.8 minutes, avoiding PSO's local optima.
- Compared to Fuzzy Q-Learning [20], D-QLSA improves resource utilization by 5.7% and reduces waiting time by 0.6 minutes, adding global optimization via SA.
- Compared to industrial IoT-based scheduling [6], D-QLSA's scheduling success rate is 8.6% higher, as it considers laboratory-specific demands (teaching/research coexistence, fragmented usage).

5.3 Limitations of D-QLSA

Despite its advantages, D-QLSA has three limitations:

1. **Sensor data dependency**: D-QLSA's performance relies on high-quality sensor data. In scenarios with severe sensor noise (>15%) or frequent failures (>20% of sensors), its resource utilization drops to <85%, as the perception data vector X becomes inaccurate.

2. **Parameter tuning complexity**: The algorithm requires tuning multiple parameters

(DQN's learning rate, SA's initial temperature, etc.). While sensitivity analysis (Table 2) provides optimal values for laboratory scenarios, adapting to other domains (e.g., hospitals, factories) requires re-tuning, increasing deployment complexity.

3. **Equipment type limitation**: The current study focuses on mechanical, electronic, and computer equipment, excluding specialized laboratory equipment such as biochemical reactors and optical microscopes. These devices have unique status indicators (e.g., chemical concentration, light intensity), requiring modifications to the perception data vector X.

4.

5.4 Practical application considerations

When deploying D-QLSA in real university laboratories, three practical considerations must be addressed:

1. **Integration with legacy systems**: Most universities have existing laboratory management systems (LMS). D-QLSA needs to provide an API interface to integrate with LMS, enabling data exchange (e.g., importing equipment maintenance schedules from LMS) and avoiding duplicate system construction.

2. **Privacy and security**: IoT sensors collect real-time data, including user identities (via demand submissions) and equipment usage patterns. Encryption (e.g., AES-256 for data transmission) and access control (role-based permissions) must be implemented to protect privacy.

3. **User acceptance**: Faculty and students may be resistant to automated scheduling. Training sessions and a user-friendly interface (e.g., mobile app for demand submission and status search) should be provided to improve acceptance [19].

6 Conclusion

The D-QLSA algorithm proposed in this paper effectively improves the performance of equipment scheduling in university laboratories. Experimental results show that over 120 days and 80 devices in a simulation, the algorithm achieved:

- A resource utilization rate of 89.2%, representing improvements of 16.7% (GA), 8.9% (DQN), 7.1% (PSO), and 5.7% (Fuzzy Q-Learning);
- An average user wait time of 2.3 minutes, a 60.3% reduction (GA), 37.8% reduction (DQN), 25.8% reduction (PSO), and 20.7% reduction (Fuzzy Q-Learning);
- A scheduling success rate of 96.8%, with a high-priority request success rate of 99.2%;
- Linear scalability (runtime increases from 1.8s to 4.2s for 80-500 devices) and robustness to sensor noise (utilization ≥85% with 15% noise).

These results validate D-QLSA's effectiveness in dual-objective optimization (maximizing utilization, minimizing waiting time) and its suitability for university laboratory scenarios. Future research will address D-QLSA's limitations and expand its applicability through three directions:

1. **Robustness enhancement**: Integrate a sensor fault detection module using LSTM networks to predict sensor failures and impute missing data via historical patterns. This will improve performance in noisy/faulty sensor scenarios (>15% noise) to maintain utilization ≥88%.

2. **Parameter adaptive tuning**: Develop a meta-learning approach to automatically tune D-QLSA's parameters based on the application domain (e.g., laboratories, hospitals). This will reduce deployment complexity and enable rapid adaptation to new scenarios.

3. **Equipment type expansion**: Extend D-QLSA to specialized laboratory equipment (biochemical, optical) by adding domain-specific sensors (e.g., concentration sensors for biochemical reactors) and updating the perception data vector X to include new status indicators.

4. **Edge-cloud collaboration**: Integrate edge computing to deploy local scheduling nodes in each laboratory and a cloud node for global optimization. This will reduce data transmission delays (from 200ms to 50ms) and enable cross-campus collaborative scheduling for multi-campus universities.

By addressing these directions, D-QLSA can evolve into a general-purpose intelligent scheduling algorithm, applicable to diverse resource scheduling scenarios beyond university laboratories.

References

[1] Zhou, Z., Jia, Z., Liao, H., Lu, W., Mumtaz, S., Guizani, M., & Tariq, M. (2021). Secure and latency-aware digital twin-assisted resource scheduling for 5G edge computing-empowered distribution grids. IEEE Transactions on Industrial Informatics, 18(7), 4933-4943. DOI: 10.1109/TII.2021.3137349

[2] Luo, Q., Hu, S., Li, C., Li, G., & Shi, W. (2021). Resource scheduling in edge computing: A survey. IEEE communications surveys & tutorials, 23(4), 2131-2165. DOI: 10.1109/COMST.2021.3106401

[3] Liao, H., Zhou, Z., Zhao, X., & Wang, Y. (2021). Learning-based queue-aware task offloading and resource allocation for space–air–ground-integrated power IoT. IEEE Internet of Things Journal, 8(7), 5250-5263. DOI: 10.1109/JIOT.2021.3058236

[4] Hu, H., Wang, Q., Hu, R. Q., & Zhu, H. (2021). Mobility-aware offloading and resource allocation in a MEC-enabled IoT network with energy harvesting. IEEE Internet of Things Journal, 8(24), 17541-17556. DOI: 10.1109/JIOT.2021.3081983

[5] Alkan, N., & Kahraman, C. (2025). Continuous Pythagorean Fuzzy Set Extension with Multi-

Attribute Decision Making Applications. Informatica, 36(2), 241-283. doi:10.15388/25-INFOR584

[6] Debroy, P., Smarandache, F., Majumder, P., Majumdar, P., & Seban, L. (2025). OPA-IF-Neutrosophic-TOPSIS Strategy under SVNS Environment Approach and Its Application to Select the Most Effective Control Strategy for Aquaponic System. Informatica, 36(1), 1-32. doi:10.15388/24-INFOR583

[7] Yan, L., Qin, Z., Zhang, R., Li, Y., & Li, G. Y. (2022). Resource allocation for text semantic communications. IEEE Wireless Communications Letters, 11(7), 1394-1398. DOI: 10.1109/LWC.2022.3170849

[8] Liu, L., Feng, J., Mu, X., Pei, Q., Lan, D., & Xiao, M. (2023). Asynchronous deep reinforcement learning for collaborative task computing and on-demand resource allocation in vehicular edge computing. IEEE Transactions on Intelligent Transportation Systems, 24(12), 15513-15526. DOI: 10.1109/TITS.2023.3249745

[9] Zhou, D., Sheng, M., Wang, Y., Li, J., & Han, Z. (2021). Machine learning-based resource allocation in satellite networks supporting internet of remote things. IEEE Transactions on Wireless Communications, 20(10), 6606-6621. DOI: 10.1109/TWC.2021.3075289

[10] Sun, Z., Sun, G., Liu, Y., Wang, J., & Cao, D. (2023). BARGAIN-MATCH: A game theoretical approach for resource allocation and task offloading in vehicular edge computing networks. IEEE Transactions on Mobile Computing, 23(2), 1655-1673. DOI: 10.1109/TMC.2023.3239339

[11] Xu, Y., Gu, B., Hu, R. Q., Li, D., & Zhang, H. (2021). Joint computation offloading and radio resource allocation in MEC-based wireless-powered backscatter communication networks. IEEE Transactions on Vehicular Technology, 70(6), 6200-6205. DOI: 10.1109/TVT.2021.3077094

[12] Debroy, P., Smarandache, F., Majumder, P., Majumdar, P., & Seban, L. (2025). OPA-IF-Neutrosophic-TOPSIS Strategy under SVNS Environment Approach and Its Application to Select the Most Effective Control Strategy for Aquaponic System. Informatica, 36(1), 1-32. doi:10.15388/24-INFOR583

[13] Wei, W., Yang, R., Gu, H., Zhao, W., Chen, C., & Wan, S. (2021). Multi-objective optimization for resource allocation in vehicular cloud computing networks. IEEE Transactions on Intelligent Transportation Systems, 23(12), 25536-25545. DOI: 10.1109/TITS.2021.3091321

[14] Saxena, D., Singh, A. K., & Buyya, R. (2021). OP-MLB: An online VM prediction-based multi-objective load balancing framework for resource management at cloud data center. IEEE Transactions on Cloud Computing, 10(4), 2804-2816. DOI: 10.1109/TCC.2021.3059096

[15] Cui, Y., Cao, K., Cao, G., Qiu, M., & Wei, T. (2021). Client scheduling and resource management for efficient training in heterogeneous IoT-edge federated learning. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 41(8), 2407-2420. DOI: 10.1109/TCAD.2021.3110743

[16] Du, J., Jiang, C., Benslimane, A., Guo, S., & Ren, Y. (2022). SDN-based resource allocation in edge and cloud computing systems: An evolutionary Stackelberg differential game approach. IEEE/ACM Transactions on Networking, 30(4), 1613-1628.DOI: 10.1109/TNET.2022.3152150

[17] Do-Duy, T., Nguyen, L. D., Duong, T. Q., Khosravirad, S. R., & Claussen, H. (2021). Joint optimisation of real-time deployment and resource allocation for UAV-aided disaster emergency communications. IEEE Journal on Selected Areas in Communications, 39(11), 3411-3424. DOI: 10.1109/JSAC.2021.3088662

[18] Walia, G. K., Kumar, M., & Gill, S. S. (2023). AI-empowered fog/edge resource management for IoT applications: A comprehensive review, research challenges, and future perspectives. IEEE Communications Surveys & Tutorials, 26(1), 619-669. DOI: 10.1109/COMST.2023.3338015

[19] Mohajer, A., Daliri, M. S., Mirzaei, A., Ziaeddini, A., Nabipour, M., & Bavaghar, M. (2022). Heterogeneous computational resource allocation for NOMA: Toward green mobile edge-computing systems. IEEE Transactions on Services Computing, 16(2), 1225-1238. DOI: 10.1109/TSC.2022.3186099

CB-DBSCAN: A Core Ball-Enhanced Grid-Based Density Clustering Method for High-Dimensional Financial Data

Xiaohe Li*, Linwei Wan
Management School, Henan University of Urban Construction, Pingdingshan, 467036, China
E-mail: xiaohe-li@outlook.com

Keywords: financial big data, high-dimensional data, clustering algorithm, DBSCAN, core ball

Received: June 18, 2025

To optimize the allocation of financial resources, improve decision-making efficiency and enhance market competitiveness, this paper proposes a financial big data clustering method based on Core Ball (CB) and Density-Based Spatial Clustering of Applications with Noise (DBSCAN). First, the algorithm improves DBSCAN using a grid tree. This divides the data space into non-empty grids and constructs a grid tree structure, achieving a fast search of adjacent grids. At the same time, an angle-based pruning strategy is introduced to eliminate redundant trivial points using geometric angle relationships, reduce the number of distance calculations, and lower the computational complexity of the algorithm. On this basis, the concept of CBs is further introduced to divide the dataset into two categories: CBs and non-CBs. The CBs are merged into the same category through density correlation judgment, and the non-CBs are distinguished between boundary points and trivial points to improve the algorithm's processing ability and clustering quality for high-dimensional data. The findings denote that, in the KDD-CUP-99 dataset, the average accuracy of the noise application spatial clustering algorithm based on CB and density is 99.1%. The recall rate shall not be lower than the minimum of 98.2%. The average precision and F1-Score are 98.2% and 98.7%, respectively, and the silhouette coefficient is not less than 0.982. In the Choice Stock Trading Comprehensive Dataset, the accuracy of applying spatial clustering algorithms based on CBs and density noise is not less than 98%. The recall rate shall not be less than 97%. The accuracy and F1-Score are both not less than 98%, which is higher than other algorithms. The average silhouette coefficient and average probability Rand index are 0.986 and 0.927, respectively. The clustering quality of this algorithm is superior to other algorithms. The above results indicate that the noise-based spatial clustering algorithm based on CBs and density can achieve efficient and accurate clustering when processing high-dimensional financial data, and has good application prospects.

Povzetek: Predstavljena je metoda združevanja za finančne velike podatke izboljša DBSCAN z mrežnim drevesom in kotnim "pruningom" ter uvede Core Ball (CB) za boljšo obravnavo visoko-dimenzionalnih podatkov, pri čemer na KDD-CUP-99 in podatkih o trgovanju z delnicami dosega zelo visoko natančnost (≈98–99%+) in boljšo kakovost gručenja kot primerjalni algoritmi.

1 Introduction

With the acceleration of digital transformation in the financial industry, financial institutions have accumulated massive amounts of customer data, transaction records, and market information. These data have the characteristics of complexity and diversity, and traditional data analysis methods are difficult to effectively explore their potential value. Cluster analysis, as an unsupervised learning method, can help financial institutions discover potential patterns and relationships from massive data, providing data support for decision-making [1-2]. For example, through cluster analysis, financial institutions can segment customers, identify the behavioral patterns of different customer groups, and provide more personalized services. In the financial field, cluster analysis has a wide range of application scenarios, including risk assessment, fraud detection, market segmentation, investment portfolio optimization, etc. [3-4]. However, in the face of high-dimensional and large-scale financial data, traditional clustering algorithms often have problems such as high computational complexity and poor clustering performance. For example, the K-means clustering algorithm is sensitive to the selection of initial center points and difficult to handle non spherical clusters. Hierarchical clustering algorithm has low computational efficiency when processing large-scale data [5]. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is algorithm that can identify clusters of any shape and is robust to noisy data. However, it has high computational complexity when processing large-scale datasets and performance bottlenecks in high-dimensional data. Therefore, to improve the clustering efficiency and quality of high-dimensional data ($\geqslant$30 dimensions), a financial big

data clustering method based on Core Ball (CB) and DBSCAN is proposed. This method first utilizes a grid tree to improve DBSCAN by dividing the data space into non-empty grids and constructing a grid tree structure to achieve fast search of adjacent grids. At the same time, an angle-based pruning strategy is introduced to eliminate redundant trivial points using geometric angle relationships, reduce the number of distance calculations, and lower the computational complexity of the algorithm. On this basis, the concept of CBs is further introduced, and the dataset is divided into two categories: CBs and non-CBs. CBs are merged into the same category based on density correlation, while non-CBs distinguish between boundary points and trivial points to improve the algorithm's processing ability and clustering quality for high-dimensional data. The innovations of the study are: firstly, DBSCAN is improved by utilizing a grid tree, while an angle-based pruning strategy is introduced to enhance the computational efficiency of the algorithm. Unlike traditional tree structure indexing, CB-DBSCAN utilizes a grid tree structure to optimize the search process, quickly locates adjacent grids through hierarchical spatial segmentation, and adopts an angle based pruning strategy to further reduce the number of distance calculations. Then CB is introduced to enhance the algorithm's ability to handle high-dimensional data. The contribution of the research lies in breaking through the limitations of low efficiency in traditional DBSCAN high-dimensional search, achieving fast positioning of adjacent grids through grid tree hierarchical indexing, and combining geometric judgment of angle pruning to eliminate redundant calculations, providing a new paradigm of "low complexity search" for high-dimensional density clustering. Meanwhile, to address the issue of bias in estimating the density of high-dimensional data, a CB partitioning and density direct connection rule is proposed to solve the clustering bias of traditional algorithms (such as fuzzy C-means and Persistent Homology-based Clustering (PHBC)) in non-spherical clusters and noisy data, and to improve the theoretical system of high-dimensional data density clustering.

1.1 Related works

Financial institutions have accumulated massive amounts of customer transaction records, market data, credit records, and so on. In massive data, valuable information often only accounts for a small portion, so effective analysis methods are needed for clustering analysis. Maulana D J et al. proposed data clustering methods using K-means and synthetic few oversampling techniques to address the issue of data imbalance in financial distress classification. The findings denoted that after introducing the above method, the accuracy of support vector machine could reach 99.1%, and the F1-Score increased to 99.91%, indicating that this method can effectively balance data and significantly improve classification performance [6]. Hunag X et al. proposed a hybrid prediction method based on fuzzy C-means clustering algorithm and convolutional neural network for the clustering and prediction of corporate financial performance. This method first performed fuzzy C-means clustering and preprocessed the clustering labels. Next, the processed data and labels were introduced into a convolutional neural network to predict financial performance levels. The results showed that this method outperformed other machine learning models in clustering and prediction abilities, proving that the model can be applied to predict corporate financial performance [7]. Wang H. proposed a feature mining model for financial product purchasing behaviour data, based on the perspective of consumer behaviour. This was developed to address the problems of low accuracy, poor precision, and low recall rate, which have been identified in traditional purchasing behaviour data feature mining models. This method first extracted and preprocesses financial product purchasing behavior data, then constructed consumer behavior preference functions and used regression functions for quadratic processing, and finally applied the term frequency reverse document frequency algorithm and K-means clustering analysis method to extract features. The findings denoted that the recall rate of this method reached 98.74%, the accuracy rate was 96.51%, and the mining accuracy rate was 98.65%. The obtained mining results have high reliability [8].

DBSCAN, as a clustering algorithm based on spatial proximity, works directly in the data space without the need to calculate the distance matrix between data points like other algorithms. Therefore, it is robust to noisy data and does not require specifying the number of clusters in advance. Kaduskar V et al. proposed a hybrid clustering strategy to address the ambiguity and overlap issues caused by inconsistent diagnoses and differences in patient conditions in medical data mining. This method employed the capacity of fuzzy C-means to manage fuzzy membership degrees, thereby enabling data points to belong to multiple clusters. Additionally, it identified and managed noise through DBSCAN, while detecting clusters of varying densities and shapes. The results showed that the clustering quality of this method was significantly improved, which helped medical professionals better understand the data [9]. Meng Q et al. suggested a debris identification method based on DBSCAN algorithm to address the performance limitations of meshless methods in aerospace protective structure simulation. This method encompassed both geometric discrimination (spatial coordinates) and physical discrimination (velocity distribution), and involved in-depth discussions and validations on the key parameters of the DBSCAN algorithm. The findings denoted that this fragment recognition method was crucial for meshless methods, as it can prevent overly conservative estimates of fragment damage [10]. Yuan G and Ma Z designed an improved multi-scale dense crowd detection method based on You Only Look Once version 5 to address the issue of insufficient ability to accurately identify dense areas in multi-scale scenes. This approach enhanced the DBSCAN clustering algorithm, facilitating the identification of densely populated areas and enabling the identification of target populations at various scales.

The findings denoted that the accuracy of the detection results was about 70%, and by calculating the population density under the same scale conditions and visualizing dense areas, the problem of dividing crowded areas could be more accurately solved [11]. The summary of related works is shown in Table 1.

Table 1: The summary of related works

Literature	Dataset	Clustering accuracy	Limitation
[6]	Enterprise Financial Distress Dataset	99.1%	Unable to handle non-spherical clusters of financial data, significant clustering bias in high-dimensional scenarios
[7]	Financial Performance Dataset of Listed Companies	/	Fuzzy membership degree leads to fuzzy boundaries of financial data clusters
[8]	Financial product purchasing behavior dataset	96.5%	Sensitive to the initial center and unable to adapt to non spherical trading behavior clusters
[9]	Medical Diagnostic Dataset	/	Not targeted for high-dimensional optimization, running slowly on large-scale data
[10]	/	81.8%	Unable to fully explore hidden information
[11]	Crowd density detection dataset	70.0%	Only suitable for dense area recognition, without noise filtering mechanism

In summary, there is currently a lot of research on clustering algorithms for financial big data, but there are still problems with poor clustering quality and low accuracy. Therefore, to improve the clustering quality of financial big data, a clustering algorithm based on CB and DBSCAN is proposed. This algorithm first divides the dataset into CB and non-CB, and uses an angle-based pruning strategy to remove trivial points, thereby improving the clustering efficiency of financial big data.

2 Big data clustering method based on CB and DBSCAN

The complexity and diversity of data make it difficult for traditional data analysis methods to effectively explore its potential value. Therefore, a financial big data clustering method based on CB and DBSCAN is proposed in this study. This method first utilizes grid trees to improve DBSCAN, to enhance the computational efficiency of the algorithm. Then, CB is introduced to improve the algorithm's high-dimensional data processing capability.

2.1 Improved DBSCAN algorithm based on grid tree

In the financial field, cluster analysis can help financial institutions discover potential patterns and relationships from massive amounts of data, providing data support for decision-making. For example, through cluster analysis, financial institutions can segment customers, identify the behavioral patterns of different customer groups, and provide more personalized services. DBSCAN, as a density-based clustering algorithm, can recognize clusters of any shape and is robust to noisy data, effectively handling data points in high-dimensional space [12-13]. For DBSCAN, the definition of its ε neighborhood is shown in equation (1).

$$N_{\varepsilon}(m)=\left\{n\in D\middle| d(m,n)\le \varepsilon\right\} \tag{1}$$

In equation (1), N_{ε} represents the ε neighborhood; ε represents the distance threshold; D represents the dataset; $d(m,n)$ denotes the distance between points m and n. The core point calculation formula of DBSCAN algorithm is shown in equation (2).

$$\left|N_{\varepsilon}(m)\right|\ge DT \tag{2}$$

In equation (2), DT represents the density threshold. Due to the high computational complexity of DBSCAN when processing large-scale datasets, to solve the above problems, research is conducted to improve it using grid trees. The improved DBSCAN divides the data space into non-empty grids, identifies core points and classifies the grids, then merges similar core grids, and finally assigns non core points [14-15]. The grid calculation formula is shown in equation (3).

$$g_i(j)=\left\lfloor \frac{c(j)-\min_{c\in C} c(j)}{\varepsilon/d} \right\rfloor \tag{3}$$

In equation (3), $g_i(j)$ represents the grid; $c(j)$ refers to the coordinate value of the data point in the j th dimension; C represents the dataset; d represents the dimensionality of the data. The core

parameters of the grid tree in GrT DBSCAN include grid size (s) and grid non-empty decision threshold (Tgrid). The selection of parameters needs to balance "search efficiency" and "clustering accuracy". The specific selection criteria and optimal values are determined as follows: the selection of grid size (s) directly determines the granularity of data space partitioning, which needs to be determined in conjunction with the data dimension (d) and distance threshold (ε) (DBSCAN core parameters). In high-dimensional data, the sample distance is easily affected by the "curse of dimensionality" and tends to be consistent. If the grid is too large, it will cause multiple core points to be divided into the same grid, increasing the redundancy of neighborhood search; If the grid is too small, a large number of empty grids will be generated, reducing the space utilization of the grid tree. The selection of the Tgrid, which defines the minimum number of samples for a "non-empty grid", needs to be combined with the density threshold setting. The grid in two-dimensional space is shown in Figure 1.

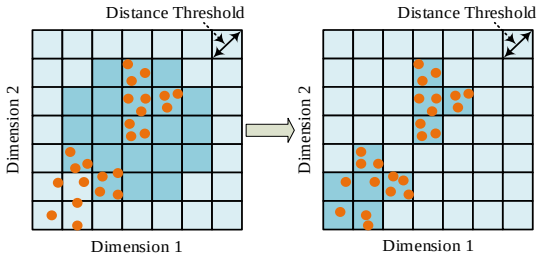


Figure 1: Grid in two-dimensional space

As shown in Figure 1, when searching for non empty grids, the identifiers of each point corresponding to the grid are first calculated and sorted accordingly. Specifically, the points will be sorted in ascending order based on the value of the grid identifier. At this point, it is only necessary to traverse the sorted data once to find all non empty grids [16]. For two grids, if their distance is less than the distance threshold, they are neighbors to each other. The distance calculation formula for the grid is denoted in equation (4).

$$d_g(g_i, g_j) = \sqrt{\sum_{t=1}^{d}\left(\frac{\varepsilon}{\sqrt{d}}\max\left\{|\, g_i(t) - g_j(t)\,| - 1, 0\right\}\right)^2} \tag{4}$$

In equation (4), $d_g\left(g_i, g_j\right)$ represents the distance between grids g_i and g_j. Due to the significant influence of data dimension on the number of neighboring grids, grid trees are introduced in the study to achieve fast search for neighboring grids. Grid tree is a data structure designed to efficiently manage and query grids in multidimensional spaces. The grid tree structure is shown in Figure 2.

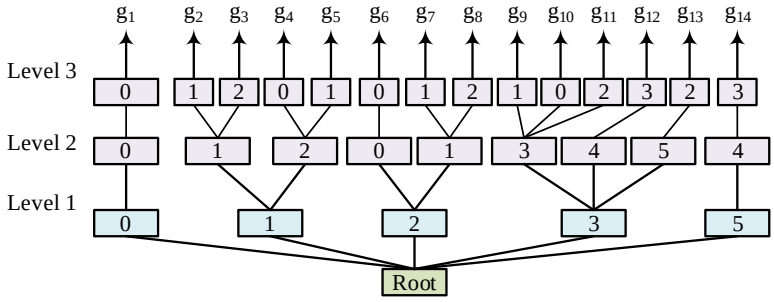


Figure 2: Structure of the grid tree

As shown in Figure 2, the grid tree has a hierarchical organizational structure, with each node representing a grid or a group of grids. It is constructed by recursively dividing the space, with each node corresponding to a specific spatial region. A grid tree consists of root nodes and leaf nodes, and the path identifier from the root node to the leaf node is composed of the keys of each layer node. When using the grid tree to find the nearest neighbor grid, if the child node is greater than the minimum number threshold, the child node with the smallest key value is selected; otherwise, all child nodes are traversed. For each node, the offset calculation formula is shown in equation (5).

$$o_{n'} = o_n + \max\{|\, KEY(n') - g(j)\,| - 1, 0\}^2 \tag{5}$$

In equation (5), $o_{n'}$ and o_n represent the offsets of child nodes and their neighboring nodes; n and n' represent different child nodes; $KEY(.)$ represents the key value. If the offset of a node is greater than or equal to the shortest distance, its descendants that do not include neighboring grids of the given grid can be removed. By using the above method, GrT-DBSCAN is constructed, which divides non-empty grids into core grids and non-core grids, enabling the identification of data core points. Next, it is necessary to merge the core grids of the same class to obtain the final category. For the task of merging grids, the key lies in how to remove trivial points through pruning. A pruning strategy based on angles is proposed for this study, with the core idea of utilizing geometric angle relationships to reduce the number of distance calculations. The formula for calculating the maximum angle of any point in the core point set that does not belong to the ε neighborhood is shown in equation (6).

$$\varphi_y = \arcsin\frac{\varepsilon}{d(p,y)} + \arccos\frac{\overrightarrow{pq}\cdot\overrightarrow{pq}}{d(p,q)\times d(p,y)} \quad (6)$$

In equation (6), ϕ_y represents the maximum angle; $d(p,y)$ represents the distance between two points; $\vec{pq}$ represents the tangent vector from point p to a circle centered on y; $\vec{py}$ represents the vector from point p to point y. If the angle between the vector from point p to point q and the vectors from point p to other points in its ε neighborhood is greater than ϕ_y, then point q is considered a non-nearest neighbor point, i.e. a trivial point, and should be deleted. Here, ϕ_y is calculated based on the distribution of points within the ε neighborhood of point p, and is used to guide the deletion decision of point q. The formula for calculating the cutting point is shown in equation (7).

$$q = \arg\min_{y\in s_j} d(p,y) \quad (7)$$

In equation (7), q represents the tangent point between $\vec{pq}$ and the circle centered on y; s_j represents the set of core points in the j th grid. The merging of similar grids can be achieved through the above method. The GrT-DBSCAN process is shown in Figure 3.

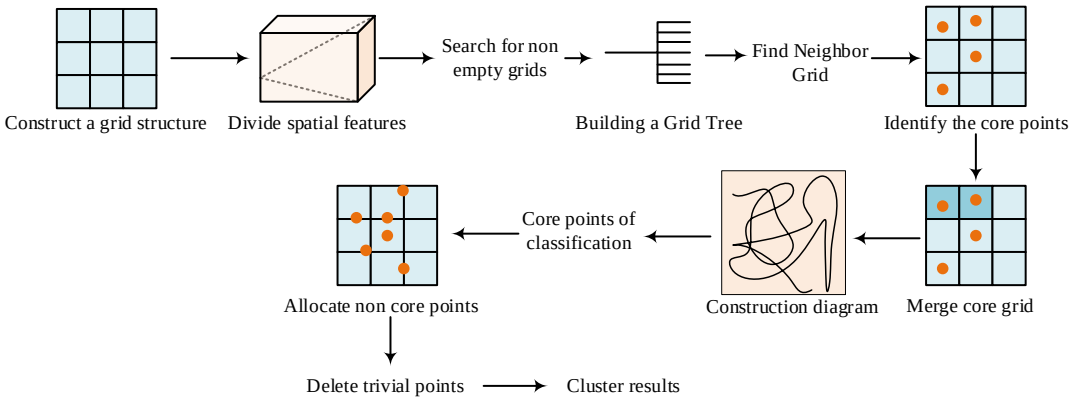


Figure 3: Process of GrT-DBSCAN

As shown in Figure 3, the algorithm first constructs a grid structure, dividing the entire data space into multiple grid cells. After constructing the grid structure, the data space is then finely divided to ensure that each data point is assigned to the corresponding grid. Next, the algorithm will search and identify each non empty grid one by one, and establish a grid tree structure based on the distribution and hierarchical relationship of the non-empty grids. Next, the nearest neighbor grids of each grid are found and the core points are identified. Then, the types of core points are identified and the core grids of the same type are merged. Next, each non-core point is assigned to determine whether it is a boundary point or a trivial point, and trivial points are removed. Finally, the clustering results are output. The pseudocode of GrT DBSCAN is shown in Table 2.

Table 2: The pseudocode of GrT DBSCAN

Pseudocode
Input: Dataset D (sample size N, dimension d), distance threshold ε, density threshold DT, grid size s, non empty grid threshold Tgrid
Output: Cluster result C (including core cluster and noise)
Step 1: Grid Data Space
2. Initialize the grid set G=∅
3. For each sample p ∈ D:
4. Calculate the grid coordinates of p in each dimension: grid_comrd (p)=(⌊p₁/s ⌋, ⌊p₂/s⌋,..., ⌊pud/s ⌋)
5. Assign p to the corresponding grid g-grid_comod (p)
6. Count the number of samples in each grid and screen for non empty grids with G_nonempty={g | count (g) ≥ Tgrid}
7.//Step 2: Build a grid tree
8. Initialize the grid tree T (with the root node being the entire data space)
9. Divide the Gnonempty into grid coordinate levels (from the 1st dimension to the d-th dimension):
10. If the current node contains more than 2 * Tgrid samples:
11. Split the node into 2 child nodes based on the median of the current dimension
12. Repeat step 10 until all leaf nodes correspond to a single non empty grid g ∈ G_nonempty
13. Assign a path identifier KEY (a combination of coordinate key values from the root node to the leaf node) to each leaf node
14.//Step 3: Angle pruning and core mesh determination
15. Initialize the core grid set with G_core=∅
16. For each grid g ∈ G_nonempty:
17. Randomly select one sample p from g as the reference point
18. Calculate the ε neighborhood N_ ε (p) of p, if count (N_ ε (p)) ≥ DT:
19.//Angle pruning: Remove trivial points in the non p neighborhood of g
20. Calculate the tangent vector pq from p to the CB

(Formula 7), and the maximum angle φ _max (Formula 6)
21. For each sample q ∈ g and q Δ N_ ε (p):
22. Calculate the angle between vector pq and py (the vector from p to q)
23. If φ>φ _max: mark q as a trivial point and remove it from g
24. Add g to the G_core
25.//Step 4: Core grid merging and non core point allocation
26. Initialize the visited grid collection with G_visited=∅, cluster C=∅
27. For each grid g ∈ G_core and g Δ G_visited:
28. Initialize the current cluster c={g}, queue Q=[g]
29. Mark g as visited (add G_visited)
30. When Q is not empty:
31. Take out the grid g-cur ∈ Q
32. Search for the neighboring grids of g2cur through the grid tree T (formula 4: grid distance<ε)
33. If g.neighbor ∈ G_core and g.neighbor Δ G_visited:
34. Add gneighbor to c and Q, mark as visited
35. Add c to C
36.//Step 5: Noise determination
37. For each sample p ∈ D:
If p is in a grid Δ G_core and p does not belong to any ε neighborhood of the core grid:
39. Mark p as noise
40. Return clustering result C

2.2 Financial big data clustering method based on CB-DBSCAN

Although the GrT-DBSCAN method mentioned above can effectively reduce computational complexity and improve data clustering efficiency, it still has performance bottlenecks when facing high-dimensional financial data. Therefore, to enhance the clustering ability of the algorithm for high-dimensional data, CB is introduced to improve the above algorithm and a financial big data clustering method based on CB-DBSCAN is constructed. The CB-DBSCAN algorithm divides high-dimensional datasets into CB and non-CB categories based on their relationships, thereby achieving distance measurement of high-dimensional data. The CB-based dataset segmentation method is shown in Figure 4.

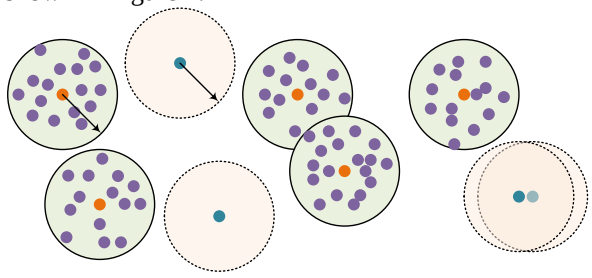


Figure 4: Data set segmentation method based on core ball

From Figure 4, for any point, if it is a core point, a CB containing all its neighboring points is generated; otherwise, a non-CB is generated. For any core point, at least one CB contains the point. It is worth noting that when performing dataset segmentation, the distance between any CBs must be greater than the CB radius [17-18]. Specifically, if there is a sample point $p \in D$ that satisfies the following two conditions, the CB $CB(p)$ is constructed with p as the center: (1) the number of samples contained in the ε neighborhood of point p is not less than the density threshold, that is, $|N_\varepsilon(p)| \geq DT$. Where $N_\varepsilon(p) = \{x \in D | d(p,x) \leq \varepsilon\}$, $d(x,p)$ represents the Euclidean distance between sample point p and x, and $|N_\varepsilon(p)|$ is the count of samples in the ε neighborhood. (2) The radius of the CB $CB(p)$ is $r(p)$ r (p), and it needs to completely cover the ε neighborhood of point p. At this point, the volume of the feature space covering all core points in the dataset is shown in equation (8).

$$V_F = (\max_{i=1}^{d} c_i + \varepsilon)^d \tag{8}$$

In equation (8), V_F represents the volume of the feature space; c_i represents the difference between the max and mini values of the i th dimensional coordinates of the core point. Based on the number of non-core points and the aforementioned feature space volume, the upper limit of CB quantity can be determined. After partitioning the dataset, the categories of each CB can be classified based on the mutual density between CB centers. If there is any CB directly connected by density, then all data points in both belong to the same category. The confirmation rule for direct density connection is as follows: (1) If there is a core point in the intersection of two CBs, then the two CBs are directly density connected. (2) If the distance between the center points of two CBs does not exceed twice the CB radius, the cardinality of set A is greater than 0, and the cardinality of set B is not less than the density threshold, then the two CBs are directly density connected. The sets A and B are shown in equation (9).

$$\begin{cases} A = \{p \mid p \in b_1 \cup b_2, \forall q \in b_1 \cup b_2 \\ d(p,q) \leq \max\left(\delta^2\varepsilon^2 - d_c^2/4, \varepsilon - \left(\delta^2\varepsilon^2 - d_c^2/4\right)\right)\} \\ B = \{p \mid p \in b_1 \cap b_2, \forall q \in b_1 \cap b_2 \\ d(p,q) \leq \left[\varepsilon - \left(\delta^2\varepsilon^2 - d_c^2/4, \varepsilon - \left(\delta^2\varepsilon^2 - d_c^2/4\right)\right)\right]\} \end{cases} \tag{9}$$

In equation (9), b_1 and b_2 represent two different CBs; δ represents the sparsity parameter of CB, the value range is [0.1,1], used to adjust the density judgment fault tolerance of the intersection area - when financial data is sparse, increasing δ can improve fault tolerance and avoid missing density connections; When financial data is dense, reducing δ can improve the strictness of judgment and avoid misjudging density connections; d_c represents the distance between the center points of two CBs. The above method can achieve accurate classification of CB, but due to the existence of a large number of unnecessary calculations, the

efficiency of the algorithm is relatively low. Therefore, to further enhance the clustering efficiency of the algorithm, a new algorithm for detecting CB distance is proposed. After removing some trivial points in CB, this algorithm uses the pruning strategy mentioned above to determine whether the shortest distance between CBs does not exceed the distance threshold. If the distance between any point and the center point of CB is greater than the sum of the distance between data points within CB and the distance threshold, then that point is a trivial point [19-20]. The distance calculation formula for data points within CB is shown in equation (10).

$$d_{in} = \max\left\{d(x, c(b_i)) \mid x \in s_i\right\} \tag{10}$$

In equation (10), d_{in} represents the distance between the data points within CB and the center point; x represents any data point within CB. After simply deleting some trivial points, the above pruning strategy can be iteratively used to remove the remaining trivial points. One of the purposes of iterative pruning strategy is to identify and remove trivial points that are not important for forming tight clusters. By calculating d_{in}, it can be determined which points are located on the boundary of the CB, which may have a relatively small impact on the clustering structure. Therefore, it can be considered to mark them as trivial points and remove them in subsequent steps. Optimize the CB: After removing some trivial points, the CB may become more compact. By iteratively calculating new values of d_{in}, the definition of the CB can be further optimized to ensure that the remaining points are more tightly wrapped around the center point, thereby improving the quality of clustering. For the number of iterations required for the above detection algorithm, if the distance between the data points in both sets does not exceed the distance threshold, then the number of iterations required is 1; otherwise, let the upper limit of the distance satisfy equation (11).

$$l = \min_{p \in s_1, q \in s_2} d(p,q) - \varepsilon \tag{11}$$

In equation (11), l represents the upper limit of distance. At this point, the number of iterations is limited by the ratio of the volumes of d-dimensional balls K and H. If the volume ratio of two balls exceeds a certain threshold, they can be considered close enough to merge. The threshold is determined by equation (11). Specifically, if the l value is small, it means that the distance between the two datasets is close, so fewer iterations may be required to achieve the merging condition. On the contrary, if the l value is large, it means that the distance between the two datasets is far and may require more iterations. The formula for calculating the radius of a d-dimensional ball is shown in equation (12).

$$\begin{cases} r_K = \dfrac{\delta\varepsilon + l}{2} \\ r_H = \dfrac{l}{2} \end{cases} \tag{12}$$

In equation (12), r_K and r_H represent the radii of d-dimensional balls K and H, respectively. The above method can be used to construct a financial big data clustering method based on CB-DBSCAN. The process of CB-DBSCAN is shown in Figure 5.

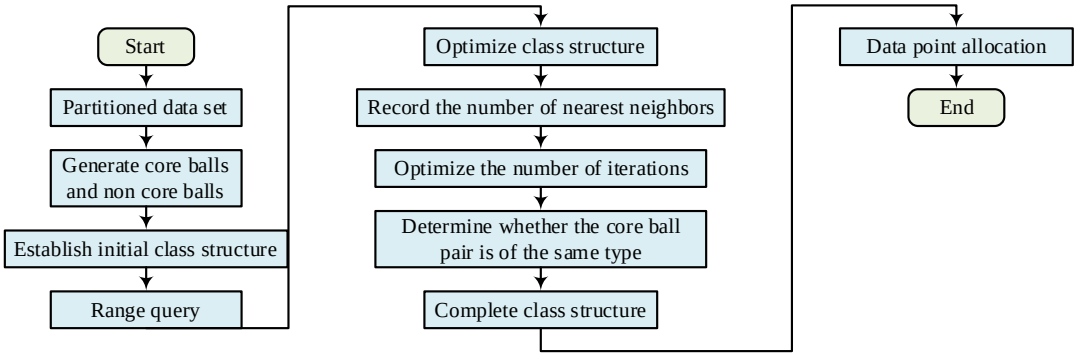


Figure 5: Process of CB-DBSCAN

As shown in Figure 5, CB-DBSCAN first divides the dataset into CB and non-CB to preliminarily identify high-density and low-density regions in the data, and based on this, establishes the initial class structure. Next, the algorithm further classifies the structure by performing range queries to ensure that the density within each cluster is sufficiently high. At the same time, the algorithm will record the number of neighbors for each data point to reduce the number of subsequent iterations. Then, for the CB pairs that have not yet been identified as belonging to a category, the above detection algorithm is used to determine whether the two belong to the same category. Finally, the final clustering result can be obtained by assigning all data points. The time complexity of the algorithm is shown in equation (13).

$$O\left(Nf(I) + T_d V_F I\right) = O\left(Nf(I) + T_d h^d I\right) \tag{13}$$

In equation (13), $O(.)$ represents time complexity; N represents the size of the dataset; I represents the input scale; T_d represents the max amount of iterations for the detection algorithm; h represents the hash coefficient; $f(I)$ represents low order functions related to dimensions. From the above equation, as the number of samples increases, the $Nf(I)$ component that

constructs the grid tree and executes the hierarchical index will dominate the asymptotic behavior of the algorithm, and the overall asymptotic time complexity of the algorithm is $O(N \cdot I \cdot \log I)$. In the context of financial big data, it exhibits a "linear growth" characteristic and has significant efficiency advantages over traditional algorithms.

3 Analysis of big data clustering test results

3.1 CB-DBSCAN clustering performance test results

To verify the clustering performance of the CB-DBSCAN algorithm, it was tested and compared with the Aggregate Multi-metric Graph Index (AMMGI) algorithm and the Fuzzy Coot Optimization K-Harmonic Means (FCOKHM) algorithm. The dataset used in the experiment was the KDD-CUP-99 dataset, which is a 34-dimensional dataset containing 5 million network connection records, each with 41 features. Although the KDD-CUP-99 dataset was not originally designed for financial data analysis, its rich features and large-scale samples made it an ideal choice for testing algorithm performance when processing high-dimensional data. To meet the needs of financial data analysis, the study performed the following preprocessing steps on the dataset: (1) Feature selection: selected features related to financial transactions, such as service type, protocol type, source address, and destination address. (2) Data cleaning: Removed missing and outlier values to ensure data quality. (3) Feature scaling: The features have been standardized to eliminate the influence of different feature dimensions. (4) Data segmentation: Divide the dataset into a training set and a testing set to evaluate the algorithm's generalization ability. The features of the processed KDD-CUP-99 dataset are shown in Table 3.

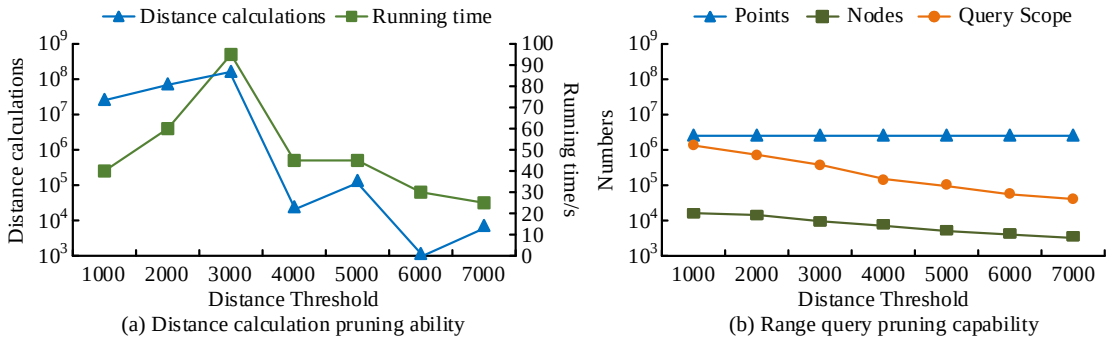
Table 3: The features of the processed KDD-CUP-99 dataset

Feature name	Description	Data type
Service type	Network service type	Classification
Protocol type	Network protocol type	Classification
Source address	Source IP address	Continuous
Destination address	Target IP address	Continuous

The approximate parameter of CB-DBSCAN in the experiment was 0.01, the sparsity parameter of CB was 1, the density threshold was 50, and the distance threshold was [1000,7000]. The CPU used in the experiment was Intel core i5 12600, and the operating system was Windows 10. The study used multiple evaluation metrics to measure the performance of the CB-DBSCAN algorithm, including accuracy, F1-Score, and Silhouette Coefficient. The calculation formulas for F1-Score and contour coefficient are shown in equation (14).

$$\begin{cases} F1-Score = 2 \times \dfrac{\Pr e \cdot \mathrm{Re}\, call}{\Pr e + \mathrm{Re}\, call} \\ Sil = \dfrac{b-a}{\max(a,b)} \end{cases} \tag{14}$$

In equation (14), $\Pr e$ represents accuracy; $\mathrm{Re}\, call$ represents the recall rate; Sil represents the contour coefficient; a is the average distance from the sample to the nearest sample in the same category; b is the average distance from the sample to the nearest different category. The impact of distance threshold on the performance of CB-DBSCAN is shown in Figure 6.



(a) Distance calculation pruning ability

(b) Range query pruning capability

Figure 6: Influence of distance threshold on CB-DBSCAN performance

As shown in Figure 6 (a), with the increase of the distance threshold, the distance calculation times of CB-DBSCAN first increased, then decreased, and then increased again, while the running time first increased and then decreased. When the distance threshold was 6000, the distance calculation times were the least, only 1000 times, and its running time was 30 seconds. As shown in Figure 6 (b), in terms of range query capability, as the distance threshold increased, the point query times of CB-DBSCAN remained unchanged, while the node and range query times gradually decreased. When the distance threshold was 6000, the number of node and range queries was 5000 and 80000 times, respectively. Overall, when the distance threshold was 6000, the comprehensive performance of the model was better. The sensitivity analysis results of density threshold are shown

in Table 4.

Table 4: The sensitivity analysis results of density threshold

Density threshold	Accuracy/%	Precision/%	F1-Score/%
30	98.2	98.9	98.2
40	98.7	98.5	98.5
50	99.1	98.7	98.9
60	98.9	98.3	98.6
70	98.5	97.8	98.1

According to Table 4, when the density threshold was 50, the various indicators of the algorithm basically reached their maximum values. At this point, its Accuracy, Precision, and F1-Score were 99.1%, 98.7%, and 98.9%, respectively. The clustering accuracy and recall of different algorithms are shown in Figure 7.

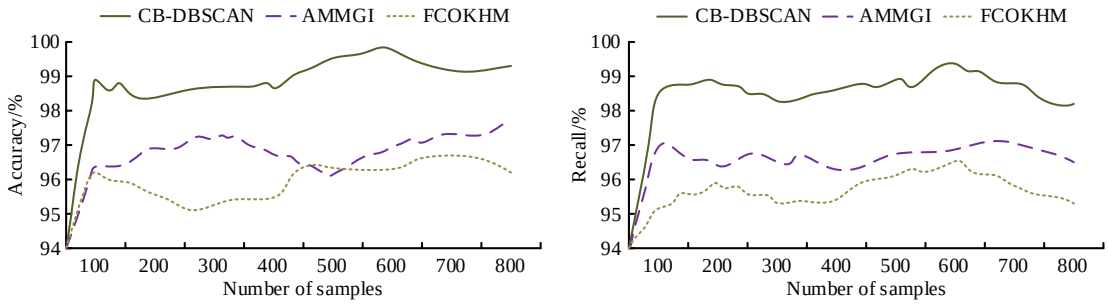


Figure 7: Cluster accuracy and recall rate of different algorithms

According to Figure 7 (a), the average clustering accuracy of AMMGI and FCOKHM was 96.9% and 96.0%, respectively. The clustering accuracy of CB-DBSCAN was always not less than 98%, and its average accuracy was as high as 99.1% (p<0.05). According to Figure 7 (b), the recall rates of AMMGI and FCOKHM were generally not more than 97%, with average recall rates of 96.7% and 95.7%, respectively. The lowest recall rate of CB-DBSCAN was 98.2%, with an average recall rate of 98.7% (p<0.05), which was also higher than other algorithms. This is because the CB-DBSCAN algorithm uses the concept of CBs to estimate the density of data points, which can more accurately measure the spatial density around the data points. Compared with traditional density estimation methods, this CB-based estimation approach can better adapt to data clusters of different shapes and sizes. The clustering accuracy and F1-Score of different algorithms are shown in Figure 8.

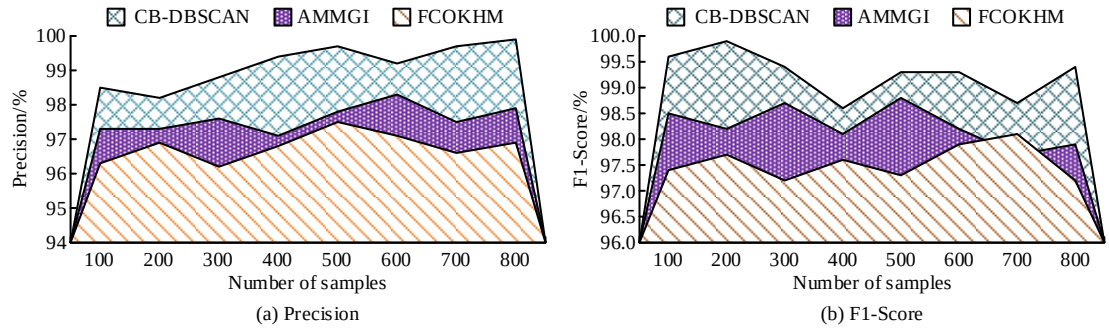
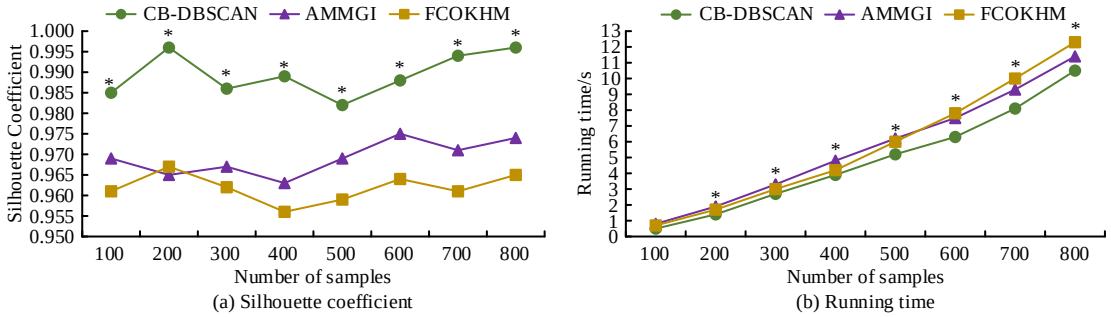


Figure 8: Clustering accuracy and F1-Score of different algorithms

According to Figure 8 (a), the clustering precision of AMMGI and FCOKHM was generally within the range of 96% -98%, with an average precision of 97.6% and 96.8%, respectively. The clustering precision of CB-DBSCAN was always not less than 98%, with an average precision of up to 99.2% (p<0.05). According to Figure 8 (b), the F1-Score ranges for AMMGI and FCOKHM were both 97%-99%, with an average F1-Score of 98.3% and 97.6%, respectively. The F1-Score of CB-DBSCAN was always not less than 98%, with an average F1-Score of 98.7% (p<0.05), which was higher than other algorithms. The above results indicate that CB-DBSCAN can achieve accurate clustering of high-dimensional data. The silhouette coefficients and running time of different algorithms are shown in Figure 9.



Note: * indicates $p<0.05$.

Figure 9: Profile coefficient and running time of different algorithms

According to Figure 9 (a), the silhouette coefficients of AMMGI and FCOKHM were the highest at 0.975 and 0.967, and the lowest at 0.963 and 0.956, respectively. The average silhouette coefficients were 0.969 and 0.962, respectively. The average contour coefficient of CB-DBSCAN was as high as 0.990 ($p<0.05$), consistently higher than other algorithms. As shown in Figure 9 (b), compared to other algorithms, CB-DBSCAN had a shorter running time. When the sample size was 800, the running times of AMMGI, FCOKHM, and CB-DBSCAN were 11.4s, 12.3s, and 10.5s ($p<0.05$), respectively. The above results indicated that CB-DBSCAN had excellent clustering performance and high computational efficiency. This is because the CB mechanism can adapt well to differences in data distribution, and for dense areas, it can identify tight cluster structures. For sparse regions, it can avoid incorrectly clustering isolated points into clusters.

3.2 Financial big data clustering test results

To verify the clustering performance of the CB-DBSCAN algorithm on financial data, the study used the financial dataset provided by the Choice Financial Information Terminal for testing. Choice Financial Information Terminal was a comprehensive financial data service platform that provides rich financial data resources, including trading records and market information of financial products such as stocks, funds, and bonds. In the study, the "Stock Trading Dataset" from Choice Financial Information Terminal was used, which includes stock trading records from 2010 to 2020. The specific dataset is named "Choice Stock Trading Comprehensive Dataset"(https://choice.eastmoney.com/product/datacenter). It was compared with PHBC and Feature Weighted Subspace Clustering (FWSC). The experimental parameters and hardware and software settings are shown in the previous section and will not be repeated. The clustering accuracy and recall of different algorithms are shown in Figure 10.

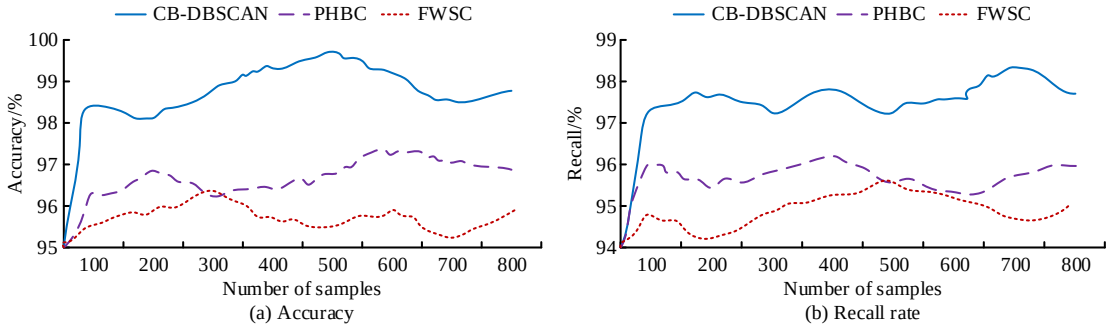


Figure 10: Cluster accuracy and recall rate of different algorithms

According to Figure 10 (a), the clustering accuracy ranges of PHBC and FWSC were 96%-98% and 95%-97%, respectively, with average accuracies of 96.8% and 95.7%, respectively. The clustering accuracy of CB-DBSCAN was the lowest at 98.1%, with an average accuracy of up to 98.9% ($p<0.05$), which was higher than other algorithms. According to Figure 10 (b), the recall rates of PHBC and FWSC were in the range of 95%-97% and 94%-96%, respectively, with average recall rates of 95.8% and 95.0%, respectively. The recall rate of CB-DBSCAN was always not less than 97%, with an average recall rate of 97.6% ($p<0.05$), which was superior to other algorithms. The clustering accuracy and F1-Score of different algorithms are shown in Figure 11.

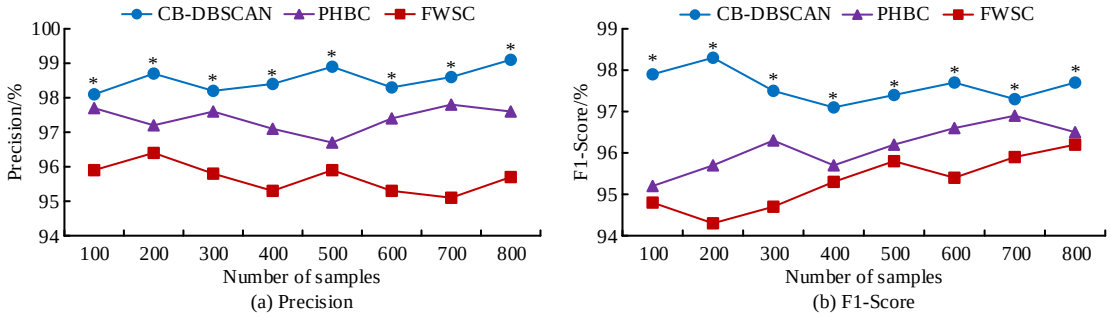


Figure 11: Clustering accuracy and F1-Score of different algorithms

According to Figure 11 (a), the clustering precision ranges of PHBC and FWSC were 97%-99% and 96%-98%, respectively, with average precision of 97.6% and 96.8%, respectively. The clustering precision of CB-DBSCAN was always above 98%, with an average precision of up to 99.2%. From Figure 11 (b), the F1-Score of PHBC and FWSC were the highest at 98.8% and 98.1%, and the lowest at 97.7% and 97.2%, respectively. The average F1-Score was 98.3% and 97.6%, respectively. The F1-Score of CB-DBSCAN was the lowest at 98.2%, with an average F1-Score of 98.7%, which was also higher than other algorithms. The above results indicate that CB-DBSCAN can achieve accurate clustering of financial big data. The silhouette coefficients and probability Rand indices of different algorithms are shown in Figure 12.

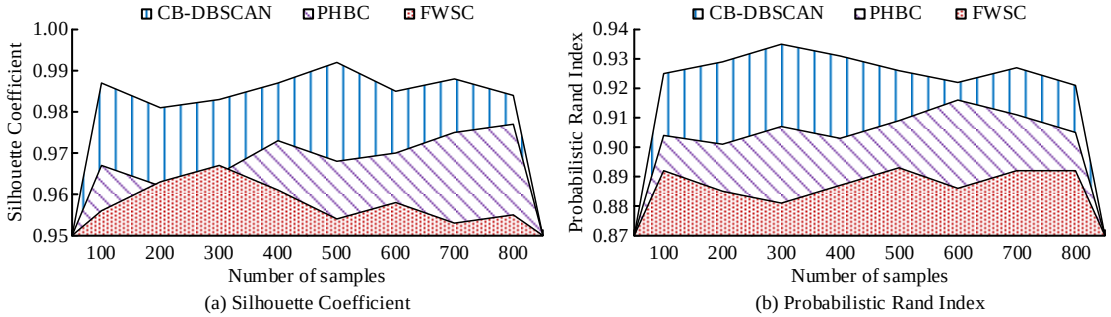


Figure 12: Silhouette coefficients and probability Rand index of different algorithms

According to Figure 12 (a), the silhouette coefficients of PHBC and FWSC were the highest at 0.977 and 0.967, and the lowest at 0.962 and 0.953, respectively. The average silhouette coefficients were 0.970 and 0.958, respectively. The silhouette coefficient of CB-DBSCAN was the lowest at 0.981, and the average silhouette coefficient was as high as 0.986, which was higher than other algorithms. According to Figure 12 (b), the probability Rand indices of PHBC and FWSC were the highest at 0.916 and 0.893, and the lowest at 0.901 and 0.881, respectively. The average probability Rand indices were 0.907 and 0.889, respectively. The lowest probability Rand index of CB-DBSCAN was 0.921, and the average probability Rand index was 0.927, which was higher than other methods. The above results indicate that CB-DBSCAN has high clustering quality for financial big data.

4 Discussion

Current research results show that CB-DBSCAN outperforms State of the Art (SOTA) methods in both high-dimensional data and financial big data scenarios. In the KDD-CUP-99 high-dimensional dataset (34 dimensions, 5 million samples), CB-DBSCAN has an average accuracy of 99.1%, an average recall of 98.7%, and an average F1-Score of 98.7%, which is 2.2-3.1% higher than AMMGI (accuracy of 96.9%, recall of 96.7%) and FCOKHM (accuracy of 96.0%, recall of 95.7%), respectively; The contour coefficient is 0.990, significantly higher than AMMGI's 0.969 and FCOKHM's 0.962, and the running time at a sample size of 800 is 10.5 seconds, which is shorter than AMMGI (11.4s) and FCOKHM (12.3s). In the Choice financial database, its average accuracy is 98.9%, recall rate is 97.6%, and F1-Score is 98.7%, which is superior to PHBC (accuracy 96.8%, recall 95.8%) and FWSC (accuracy 95.7%, recall 95.0%), with an average silhouette coefficient of 0.986 and a probability Rand index of 0.927, also leading the comparison algorithm.

The advantages of CB-DBSCAN stem from two core designs: firstly, the scalability of the grid tree, which divides the high-dimensional space into non-empty grids through hierarchical partitioning, and quickly locates adjacent grids with path identifiers, reducing the search complexity from O (N^2) to O (NlogN), solving the problem of inefficient search in AMMGI and PHBC high-dimensional environments, and reducing the increase in running time when the sample size increases; The second is the robustness of angle pruning, which is

based on geometric angle judgment to remove trivial points (such as only 1000 distance calculations when the distance threshold is 6000), reduce redundant calculations while filtering noise, and avoid clustering bias caused by noise interference in FWSC and fuzzy C-means. Financial data still maintains an accuracy of over 98% even when containing 5% abnormal noise.

It is worth noting that the algorithm has three types of applicable boundaries: firstly, extremely sparse financial data (such as niche product customer data, with samples<100), where the number of neighboring samples is below the density threshold and cannot form a CB, making it easy to classify all samples as noise; The second type is mixed clusters with significant density differences (the density difference between ultra dense clusters and extremely sparse clusters is more than 10 times), and a globally unified density threshold can cause extremely sparse clusters to be misjudged as noise; The third is the dynamic financial data flow. Static clustering requires rebuilding the grid tree and CB, which cannot be incrementally updated, resulting in delayed response in real-time scenarios (such as adding 1000 transactions per second). In the future, scene adaptability can be improved through adaptive density threshold and integrated incremental framework optimization.

5 Conclusion

To achieve effective clustering of financial big data, a CB-DBSCAN-based clustering method for financial big data was proposed. The outcomes denoted that the CB-DBSCAN algorithm had significant performance advantages in processing high-dimensional financial data. The performance test results (KDD-CUP-99 dataset) showed that CB-DBSCAN outperformed the existing advanced methods AMMGI and FCOKHM, with an average accuracy and recall of 99.1% and 98.7%, respectively, and an average accuracy and F1-Score of 99.2% and 98.7%, respectively, both higher than other algorithms. In addition, the silhouette coefficient of CB-DBSCAN was the lowest at 0.982, and the average silhouette coefficient was as high as 0.990, indicating high internal consistency of its clustering results, shorter running time, and high computational efficiency. In practical application testing results (Choice Stock Trading Comprehensive Dataset), CB-DBSCAN also performed well. Compared to other algorithms, CB-DBSCAN had an average accuracy of 98.9%, an average recall of 97.6%, an average precision of 99.2%, and an average F1-Score of 98.7%, all of which are higher than other algorithms. The average silhouette coefficient and average probability Rand index were 0.986 and 0.927, respectively, which were also superior to other algorithms. The above findings denote that the CB-DBSCAN algorithm not only achieves accurate clustering when processing high-dimensional financial data, but also has high computational efficiency and clustering quality, making it suitable for clustering analysis of large-scale financial data. Although the CB-DBSCAN algorithm performs well on static datasets, its clustering results have poor quality when dealing with dynamically growing data. Therefore, in the future, it will consider introducing sliding window or CluStream algorithms to achieve high-quality and effective clustering of dynamic data.

References

[1] Du L, An X. An enterprise financial credit risk measurement method based on differential evolution algorithm. International Journal of Information Technology and Management, 2025, 24(1-2): 67-77. DOI:10.1504/IJITM.2025.144106.

[2] Sudha D, Gowri S. Deep FC-IIWO Fuzzy Clustering: An Optimized Fuzzy Clustering Approach for Big Data Clustering. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2025, 33(03): 329-352. DOI:10.1142/S021848852550014X.

[3] Zhu J, Han X. Big data clustering algorithm of power system user load characteristics based on K-means and SOM neural network. Multimedia Tools and Applications, 2025, 84(10): 7477-7491. DOI:10.1007/s11042-024-19156-1.

[4] Ramaijane T, Hlomani H, Zlotnikova I, & Maupong T. Evaluation of Manifold Dual Contouring Algorithms Based on k-d tree and Octree Data Structures. Informatica, 2024, 48(3):399-418. DOI:10.31449/inf.v48i3.5420.

[5] Agarwal S, Reddy C R K. A smart intelligent approach based on hybrid group search and pelican optimization algorithm for data stream clustering. Knowledge and Information Systems, 2024, 66(4): 2467-2500. DOI:10.1007/s10115-023-02002-5.

[6] Maulana D J, Saadah S, Yunanto P E. Kmeans-SMOTE Integration for Handling Imbalance Data in Classifying Financial Distress Companies using SVM and Nave Bayes. Jurnal RESTI, 2024, 8(1): 54-61. DOI:10.29207/resti.v8i1.5140.

[7] Huang X, Hu Y, Liu H. A hybrid FCM-CNN method to cluster and forecast financial performance of listed companies. Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology, 2023, 44(2): 1991-2006. DOI:10.3233/JIFS-221995.

[8] Wang H. A feature mining model for financial product purchase behaviour data based on consumer behaviour perspective. International Journal of Business Intelligence and Data Mining, 2025, 26(1): 205-217. DOI:10.1504/IJBIDM.2025.143935.

[9] Kaduskar V, Srivastava A, Husain S O, Gupta M, & Patil V. Integrating Fuzzy C-Means and DBSCAN: A Hybrid Approach to Medical Data Mining. Fuzzy Information and Engineering, 2025, 17(1): 108-119. DOI:10.26599/FIE.2025.9270055.

[10] Meng Q, Fan J, Yuan Y, & Su Y. Fragment Identification Method Based on DBSCAN and Its Application in Hypervelocity Impact of Whipple Shield. International Journal of Aeronautical and Space Sciences, 2025, 26(3): 989-1000. DOI:10.1007/s42405-024-00814-5.

[11] Yuan G, Ma Z. Research on Dense Crowd Area Detection Method Based on Improved YOLOv5 and Improved DBSCAN Clustering Algorithm. Journal of Applied Mathematics and Physics, 2024, 12(12): 4206-4212. DOI:10.4236/jamp.2024.1212259.

[12] Regilan S, Hema L K. Optimizing environmental monitoring in IoT: integrating DBSCAN with genetic algorithms for enhanced clustering. International Journal of Computers & Applications, 2024, 46(1): 21-31. DOI:10.1080/1206212X.2023.2277966.

[13] Pan Y R, Xia Y H, Long L J, &Yang M L. Power-Line Extraction and Modelling from 3D Point Clouds Data Based on K-D Tree DBSCAN Algorithm. Journal of Electrical Engineering & Technology, 2024, 19(5): 3587-3597. DOI:10.1007/s42835-023-01641-6.

[14] Mosin V G, Kozlovskii V N, Nenashev M V. Novel Tools for Monitoring Process Quality: Detection of Anomalies in Digital Data on Content Consumption Using the DBSCAN Clustering Method. Russian Engineering Research, 2024, 44(9): 1370-1374. DOI:10.3103/S1068798X24702137.

[15] Yang W, Zhu Z, Zhou M, Li D, Zhang J, & Qin M, et al. Extraction of Tooth Cusps Based on DBSCAN Density Clustering and Neighborhood Search Algorithm. Critical Reviews in Biomedical Engineering, 2024, 52(2): 27-37. DOI:10.1615/critrevbiomedeng.2023050386.

[16] Wenying Z. An improved DBSCAN Algorithm for hazard recognition of obstacles in unmanned scenes. Soft computing: A fusion of foundations, methodologies and applications, 2023, 27(24): 18585-18604. DOI:10.1007/s00500-023-09319-x.

[17] Chen B Y, Luo Y B, Zhang Y, Jia T, Chen H P, & Gong J, et al. Efficient and scalable DBSCAN framework for clustering continuous trajectories in road networks. International Journal of Geographical Information Science, 2023, 37(8): 1693-1727. DOI:10.1080/13658816.2023.2217443.

[18] Ouf S, Ashraf M, Roushdy M. A Proposed Paradigm Using Data Mining to Minimize Online Money Laundering. Informatica, 2024, 48(3):309-328. DOI:10.31449/inf.v48i3.6103.

[19] Xing Z, Zhao W. Block-diagonal guided DBSCAN clustering. IEEE Transactions on Knowledge and Data Engineering, 2024, 36(11): 5709-5722. DOI:10.1109/TKDE.2024.3401075.

[20] Hauseux L. How can we theoretically measure the performance of density-based clustering algorithms. Performance Evaluation Review, 2025, 52(4): 19-20. DOI:10.1145/3725536.3725546.

Facial Expression Localization and Recognition Using MDMO and Transformer for Mental Health Diagnosis Support

Yan Yan[1], Hanjun Li[2*]
[1]Guangdong Polytechnic of Industry and Commerce, Guangzhou 510510, China
[2]Guangzhou College of Commerce, Guangzhou 511363, China
E-mail: nyyanyan2008@126.com, LiHanjun_lhj@outlook.com
[*]Corresponding author

Keywords: expression localization, mental health auxiliary diagnosis and treatment, transformer, MDMO, facial expression recognition

Received: September 25, 2025

With the work and study pressure on people increasing and the importance of mental health problems on the rise, facial expression analysis plays an important role in mental health auxiliary diagnosis and treatment (MHADT). This study proposes a facial expression localization and recognition model that integrates Main Directional Mean Optical Flow (MDMO) with the Transformer architecture. It addresses the problems of insufficient generalization ability and limited temporal modeling ability of traditional methods in facial expression recognition. The study is based on the authoritative Audio/Visual Emotion Challenge 2019 – Depression Detection Sub-challenge (AVEC2019 DDS) dataset in the mental health field, which contains 163 training samples, 56 validation samples, and 56 test samples. With Explicit Shape Regression, Local Binary Patterns, Mnemonic Descent Method, Convolutional Neural Network, and benchmark models as comparison objects, it systematically evaluates the model's performance in error, accuracy, and processing speed. The results show that the model achieves the best performance in both facial expression localization and recognition tasks. The validation set errors are 7.26 and 6.85, the localization accuracy reaches 89.6%, and the recognition accuracy reaches 88.9%, which is significantly better than other methods. At the same time, its image processing time is 55 milliseconds (ms) and 44ms, balancing high precision and real-time performance. The study indicates that the fusion of MDMO and Transformer can effectively capture the spatial and temporal features of facial expressions. Thus, this fusion method provides an efficient, stable, and scalable technical solution for emotion recognition in MHADT. This improves both the FER and localization accuracy and effect. Besides, it provides a novel reference in both approaches and application level for the intelligent development of mental health evaluation.

Povzetek: Predstavjen Model MDMO + Transformer natančno in hitro prepoznava obrazne izraze za podporo oceni duševnega zdravja.

1 Introduction

With the increasing severity of global mental health problems, traditional face-to-face psychotherapy faces challenges such as resource shortage, geographical restrictions and high costs [1]. According to statistics from the World Health Organization, nearly 1 billion people worldwide suffer from mental illnesses, whereas the number of people receiving professional treatment is far lower than the demand. Especially in low- and middle-income countries, the accessibility of mental health services is more limited [2-3]. Against this background, digital mental health intervention methods have emerged as an important way to alleviate this problem. For example, smartphone applications have been proven to effectively reduce symptoms of depression and anxiety and improve users' psychological well-being by providing self-assessment, cognitive behavioral therapy courses, and relaxation training [4]. However, existing digital intervention tools mostly rely on text or voice input; these tools lack real-time perception of users' facial expressions and emotional changes, which limits their application effects in complex emotion recognition and intervention.

Recent years have witnessed fast advancement in artificial intelligence technology, particularly with the emergence of multimodal deep learning (DL) models. These developments are steering digital mental health interventions toward increasingly intelligent and personalized approaches [5-7]. Multimodal DL methods can process multiple input information such as text, voice,

and vision at the same time, realizing comprehensive perception and analysis of users' emotional states. For example, models based on the Transformer architecture have achieved remarkable results in natural language processing tasks; their strong context modeling ability provides new ideas for emotion recognition and psychological intervention [8]. These methods perform well in laboratory environments. However, they still face many challenges in practical applications, such as the real-time performance of the model, generalization ability, and adaptability to small-sample data. Therefore, how to improve the real-time performance and generalization ability of the model while ensuring high accuracy has become a hot and difficult issue in current research.

This study explores the application of the expression localization and recognition model based on Main Directional Mean Optical Flow (MDMO) and Transformer architecture in mental health auxiliary diagnosis and treatment (MHADT). Specifically, this study proposes an expression localization and recognition model combining MDMO and Transformer, thus improving the model's localization and emotion recognition accuracy in MHADT scenarios. Through experiments on the Audio/Visual Emotion Challenge 2019 – Depression Detection Sub-challenge (AVEC2019 DDS) dataset, the model's superiority in terms of localization accuracy, recognition accuracy, and processing time is verified. In addition, this study conducts statistical significance tests to verify the model's reliability and performance improvement. The innovation of this study lies in combining MDMO with the Transformer architecture for the first time. It proposes a new expression localization and recognition model, which provides a new technical means for MHADT. The research results show that the model can effectively improve the accuracy of expression localization and emotion recognition, providing strong support for developing digital mental health intervention tools. This study enriches the application scenarios of multimodal DL in mental health; it also offers a reference for designing and optimizing intelligent mental health intervention systems in the future.

2 Related work

Facial expression recognition (FER) research in the MHADT field has become a hot direction at the intersection of AI and medicine [9]. Domestic and international scholars have extensively explored model optimization, feature extraction, and clinical adaptability, forming rich research results. However, there is still room for improvement in the applicability of complex diagnosis and treatment scenarios.

Foreign research started relatively early and led in both theoretical system construction and technological innovation. Kondal et al. studied the "basic emotion theory" and defined six cross-culturally universal basic expression categories: happiness, anger, sadness, fear, surprise, and disgust. The facial action coding system they developed provided a standardized feature annotation basis for FER by labeling 46 facial muscle movement units. This theory remains the core theoretical support for most FER models to this day. Over the years, with the development of DL technology, Jabbooree et al. constructed an FER model based on a Convolutional Neural Network (CNN). This model achieved a classification accuracy of 89.2% on the FER-2013 dataset. Still, it only concentrated on the judgment of a single expression category and did not involve the associated inference of psychological states [10]. Kim et al. attempted to apply the Transformer to expression analysis and captured global facial features through a self-attention mechanism. Although this approach improved the sensitivity of micro-FER, it did not consider the problem of disguised expressions caused by patients' social concealment psychology, resulting in insufficient robustness in clinical diagnosis and treatment scenarios [11].

Domestic research focuses more on the adaptability of technology to local clinical scenarios and has formed unique advantages in multi-modal fusion and lightweight model development. Zhang and Chai proposed a FER scheme based on "visual-physiological" multimodal fusion. They constructed a multi-feature fusion model by synchronously collecting facial images and heart rate variability signals. The experimental results showed that the accuracy of this model in the expression analysis of patients with depressive tendencies was 15.7% higher than that of the single visual feature model. Their research pointed out that "single visual features could not fully reflect psychological states. Besides, introducing physiological signals could effectively make up for the recognition bias caused by expression disguise", which provided a key idea for multi-modal research in MHADT [12]. In response to the current situation of limited hardware resources in primary medical scenarios in China, Zhang et al. presented a lightweight FER model. Through model pruning and quantization technologies, they compressed the model parameter size from 23.6 megabytes (MB) of the traditional CNN to 4.8MB; meanwhile, it maintained an expression classification accuracy of 82.1%, and successfully realized deployment on primary medical terminals (such as portable diagnostic instruments). However, this model still had distinct limitations and could only complete the single task of "expression classification"; it could not simultaneously realize the localization of core expression areas (such as identifying the frontal muscle movement area corresponding to "frowning") and the inference of psychological states. Its output results were limited to categorical labels such as "anger" and "happiness." This limitation created a serious disconnect from the multi-link decision-making needs of clinical physicians, which involved locating abnormal expression areas, judging

emotional types, and assessing psychological risks. Consequently, the system struggled to function effectively in auxiliary diagnosis and treatment (ADT) contexts [13]. Zhao et al. attempted to construct a multi-task expression analysis model to simultaneously optimize the tasks of expression classification and psychological state assessment. However, this model adopted a traditional multi-task learning framework and did not design a special feature interaction mechanism for mental health scenarios, leading to interference in feature extraction between the two tasks. The Mean Absolute Error (MAE) for psychological state assessment reached 0.87, failing to meet the clinical standards for evaluation accuracy [14].

To sum up, although existing studies have made progress in FER accuracy and model lightweight, there are still obvious limitations. Most models still focus on single-task and single-dimensional analysis, which makes it difficult to meet the comprehensive needs of the "localization - recognition - inference" link in MHADT. In addition, existing methods lack a multi-task optimization framework designed for mental health scenarios, leading to a certain disconnect between model output and clinical practical application. This restricts the practicality and reliability of these methods in ADT. According to this, the study introduces an expression localization and recognition scheme that fuses MDMO and Transformer. It aims to break through the limitations of existing research and improve the accuracy and practicality of expression analysis in MHADT.

3 The expression localization and recognition model using MDMO and transformer in MHADT

This study aims to verify whether combining MDMO with the Transformer structure can significantly improve the accuracy and stability of FER in MHADT scenarios. The study explores whether this fusion model can more effectively capture the facial expressions' dynamic changes and temporal dependencies. On the clinical AVEC2019 DDS dataset, the model's comprehensive performance is assessed across accuracy, error rate, and processing speed through comparisons with Explicit Shape Regression (ESR), Local Binary Patterns (LBP), Mnemonic Descent Method (MDM), CNN, and benchmark models. These evaluations confirm its feasibility and application potential for clinical psychological emotion recognition.

3.1 The MDMO feature and transformer model

(1) MDMO feature

The MDMO feature is a technical method designed for extracting changing features of expressions [15]. By calculating the main direction and average speed of pixel movement in facial regions, it can accurately capture the core temporal information of changes in expressions [16]. It can filter out interference from irrelevant movements such as head shaking and light changes while highlighting key motion features related to emotions [17]. This effectively makes up for the defects of traditional static features (which lose temporal information) and ordinary optical flow technology (which has computational redundancy). Thus, it can provide better real-time feature support for analyzing weak and mixed emotional expressions in MHADT scenarios. It is an important foundation for the subsequent realization of accurate expression localization and recognition by combining with the Transformer model [18].

This study selects the direction interval $B_{\max}$ that contains the largest number of optical flow vectors, and calculates the average value of the optical flow vectors within $B_{\max}$:

$$\bar{u}_i^k = \frac{1}{\left|B_{\max}\right|} \sum_{u_i^k \in B_{\max}} u_i^k(p) \tag{1}$$

k=1,2,…,36 represents the serial number of the Region of Interest (ROI); $u_i^k(p)$ refers to the pixel point belonging to the k-th ROI (ROI_i^k) in the i-th frame; p denotes the optical flow vector. Then, these vectors in each ROI are connected to obtain the i-th frame's feature vector:

$$\psi_i = \left(\bar{u}_i^1, \bar{u}_i^2, \ldots, \bar{u}_i^{36}\right) \tag{2}$$

Next, the average feature vector of the entire micro-expression sequence is calculated and converted into polar coordinates:

$$\bar{\psi} = \left[\left(\bar{\rho}_1, \bar{\theta}_1\right)^T, \left(\bar{\rho}_2, \bar{\theta}_2\right)^T, \ldots, \left(\bar{\rho}_{36}, \bar{\theta}_{36}\right)^T\right] \tag{3}$$

Since the main direction's intensity varies among different micro-expression sequences, further normalization of the magnitude ρ of Ψ is required:

$$\rho_k = \frac{\bar{\rho}_k}{\max\left\{\bar{\rho}_j, j = 1, 2, \ldots, 36\right\}} \tag{4}$$

Finally, the normalized MDMO feature of a micro-expression sequence can be expressed as:

$$\psi = \left[\left(\rho_1, \theta_1\right)^T, \left(\rho_2, \theta_2\right)^T, \ldots, \left(\rho_{36}, \theta_{36}\right)^T\right] \tag{5}$$

To balance the influence of the magnitude ρ and the direction θ, a weight parameter λ is also introduced, and the MDMO feature is rewritten as:

$$\bar{\bar{\psi}} = \left(\lambda P, (1-\lambda)\Theta\right) \tag{6}$$

P denotes the magnitude part: P=(ρ1, ρ2, …, ρ36\); Θ represents the direction part: Θ=(θ1, θ2, …, θ36).

(2) Transformer Model

The overall structure of Transformer mainly consists of an encoder, a decoder, as well as related attention mechanisms and fully connected layers [19]. Its upper layer includes units composed of self-multi-head attention, a fully connected feedforward network, and residual normalization modules, which are used for feature extraction and information processing [20-21]. The middle part is the main body of the encoder and decoder. The encoder is formed by connecting multiple encoder units in series to encode the input content; the decoder consists of many sequentially connected decoder units that progressively transform the encoder's output into the final generated result. At the same time, it transmits auxiliary information between the Encoder and Decoder through "Focused Information". The lower layer includes a fully connected feedforward network, self-multi-head attention, encoder-decoder attention, and residual normalization modules to assist the decoder in achieving accurate output. The overall structure is progressive, and the attention mechanism and residual normalization ensure the efficiency and accuracy of the Transformer in sequence information processing [22]. The Transformer's overall structure is presented in Fig. 1:

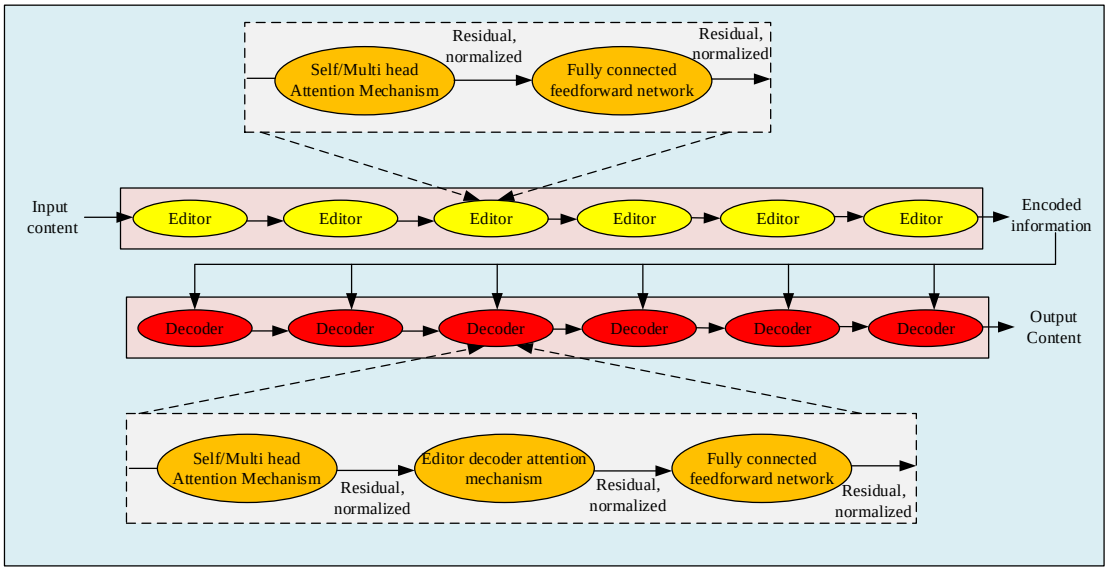


Figure 1: The overall structure of the Transformer

In Fig. 1, the encoder processes the input sequence, while the decoder is responsible for generating the output. The most important component in both the encoder and the decoder is the attention mechanism. This mechanism's implementation is called "Scaled Dot-Product Attention", and its calculation process is as follows:

$$Attention(Q,K,V) = soft\max\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{7}$$

Q, K, and V represent Query, Key, and Value, respectively.

Equation (8) presents the parallel modeling process of multi-head attention:

$$MultiHead(Q,K,V) = Concat(head_1,\ldots,head_h)W^o \tag{8}$$

Equation (9) defines the Feed Forward Network (FNN)-based nonlinear feature transformation:

$$where head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{9}$$

All variables denoted as W in Equations (8) and (9) are parameter matrices; W^o refers to the matrix for merging multiple heads; W indicates the parameter matrix used to obtain the projected Q, K, and V; *Concat* stands for the concatenation operation.

3.2 The expression localization model grounded on MDMO and Transformer in MHADT

The expression localization model based on MDMO and Transformer in the proposed MHADT has a four-layer structure. The input layer receives face images and performs image preprocessing; the feature extraction layer sequentially completes optical flow estimation, optical flow direction calculation, and MDMO feature extraction; the feature modeling layer encodes the extracted features and processes them through a

Transformer Encoder containing self-attention mechanism, multi-head attention, and feedforward neural network; the localization output layer first predicts key facial expression regions and finally outputs the expression localization results. The model is progressive from image input to feature processing and then to localization output; it also presents the complete process of achieving accurate facial expression localization in MHADT based on MDMO and Transformer technologies [23]. The expression localization model's hierarchical structure by the Transformer and MDMO in MHADT is shown in Fig. 2:

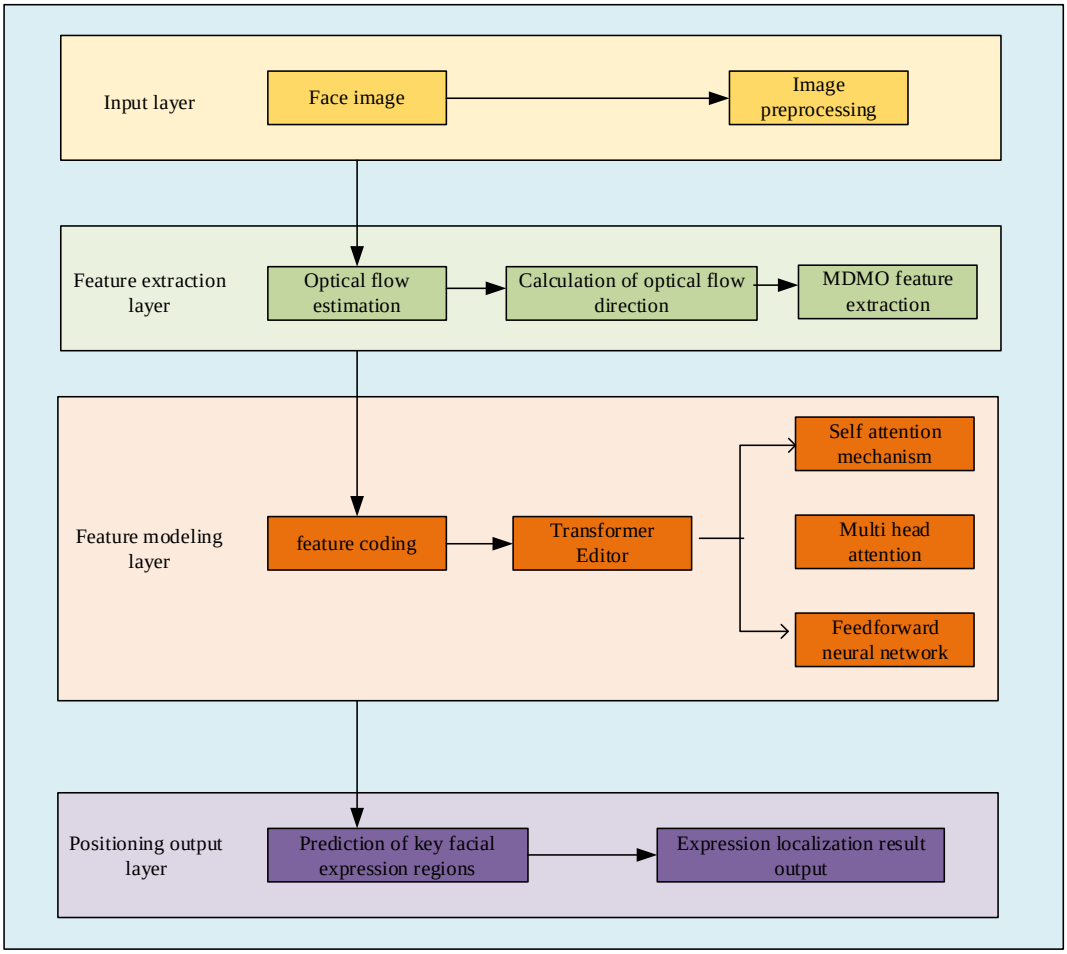


Figure 2: Hierarchical diagram of the Transformer and MDMO-based expression localization model in MHADT

First, this study uses the optical flow method to obtain the target facial movement information and extract directional features. Second, it models the dominant directional features of expression changes through MDMO. Finally, it combines a self-attention mechanism of the Transformer and the FNN to realize the accurate localization and discrimination of key expression regions.

(1)Calculation of optical flow vectors

$$v(x,y,t)=\left(u(x,y,t),v(x,y,t)\right) \tag{10}$$

$v(x,y,t)$ represents the optical flow vector of the pixel point (x,y) at time t, including the horizontal component u and the vertical component v. This step extracts the movement information of facial expressions that change over time.

(2)Calculation of optical flow direction angle

$$\theta(x,y,t)=\arctan\left(\frac{v(x,y,t)}{u(x,y,t)}\right) \tag{11}$$

The movement's main direction information is obtained by calculating the optical flow vector's direction angle θ. This provides a basis for the subsequent construction of the direction histogram and the extraction of MDMO features.

(3) MDMO

$$MDMO_k=\frac{1}{N_k}\sum_{(x,y,t)\in D_k} v(x,y,t) \tag{12}$$

D_k represents the k-th direction interval; N_k stands for the number of optical flow vectors in this interval. In $MDMO_k$, O denotes the mean optical flow feature of this

direction interval. Through Equation (12), the most representative main directional movement features in expression changes can be extracted.

(4) Self-attention mechanism modeling

$$Attention(Q,K,V) = soft\max\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (13)$$

Q, *K*, and *V* respectively represent the query, key, and value vectors encoded by MDMO features, and *dk* denotes the feature dimension. Equation (13) is employed to capture the global dependency relationship of expression features among various regions.

(5) Localization prediction output

$$\hat{y} = FFN\left(Attention(Q,K,V)\right) \quad (14)$$

The attention result is subjected to a nonlinear transformation through an FFN; the predicted position or feature representation of the key expression region is output, thereby realizing the accurate localization of expressions.

3.3 The FER model using MDMO and transformer in MHADT

In MHADT, the FER model integrating MDMO and Transformer focuses on realizing multi-level emotion discrimination based on accurately locating key expression regions [24-25]. First, the model fuses the features extracted by MDMO with the change information of local regions to highlight motion patterns related to emotions. Then, through the Transformer's global modeling and multi-head attention mechanism, it effectively integrates the dependency relationships between various facial regions and potential emotional features. Finally, the classification layer completes the discrimination and output of diverse expression categories. This model can maintain sensitivity to subtle expression changes in complex environments, providing technical support with higher accuracy and stronger robustness for the automated assessment of psychological states and clinical ADT.

The proposed FER system based on MDMO and Transformer mainly consists of a presentation layer and a back-end model layer. The presentation layer includes a login module, a local face recognition module, and a camera recognition module, which realize user interaction and data collection functions. The back-end model layer takes the MDMO- and Transformer-based expression recognition model as the core; it performs feature selection, fusion, and weight optimization through the model fusion layer, and finally outputs accurate expression recognition results. The overall system design is progressive from user interaction to model operation, constructing a complete expression recognition process and realizing a full-process closed loop of data collection, processing, and high-precision recognition. The design of the FER system based on MDMO and the Transformer is depicted in Fig. 3:

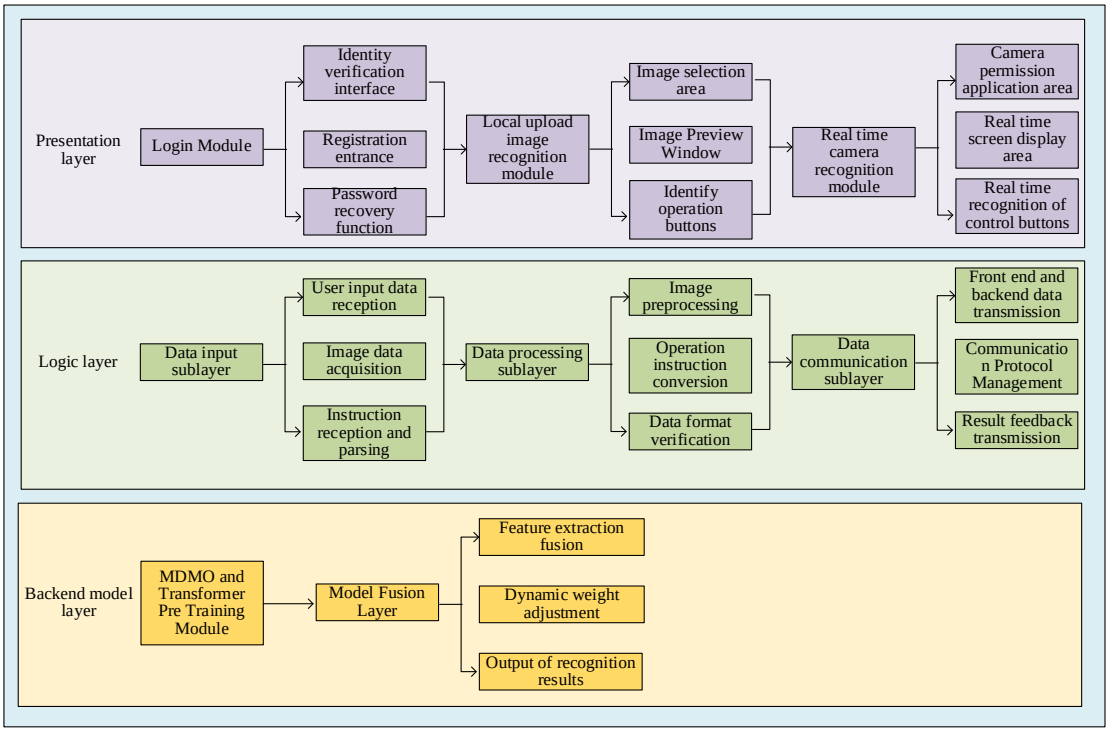


Figure 3: Design of the MDMO- and Transformer-based FER system

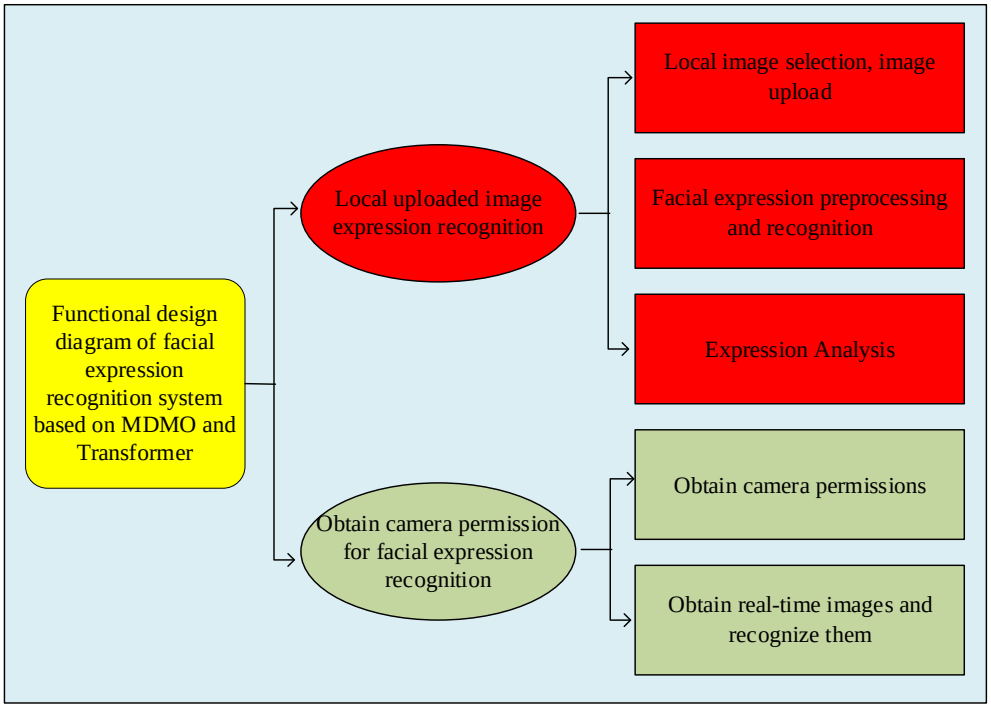


Figure 4: The FER system module architecture based on MDMO and Transformer

Table 1: Dataset splitting strategy

Subset	Sample size	Splitting strategy	Basis for splitting	k-fold cross-validation	Description
Training set	163	Stratified random split	Stratify by individual subjects and emotion categories to ensure balanced proportions across categories	No	Used for model parameter training to ensure all types of expressions are represented
Validation set	56	Independently sampled using the same strategy as the training set	Randomly sample according to subject and emotion category stratification	No	Used for model tuning and early stopping to ensure validation performance reflects generalization ability
Test set	56	Independently sampled, no overlap with training/validation set	Stratify by subject to ensure that subjects in the test set do not appear in training/validation.	No	Used for final model performance evaluation to test generalization and clinical applicability

The proposed FER system mainly includes two modules: expression recognition of locally uploaded pictures and expression recognition of obtaining camera permission. The system module architecture is illustrated in Fig. 4:

4 Application of the proposed expression localization and recognition model based on mdmo and transformer in MHADT

In MHADT, the data used in the application experiment of the proposed model using MDMO and Transformer mainly comes from the AVEC2019 DDS dataset. This dataset is collected and organized from audio-visual records of patients during clinical interviews; it can truly reflect the facial expressions and emotional changes of subjects under different mental health states. Specifically, the AVEC2019 DDS dataset contains 163 training samples, 56 validation samples, and 56 test samples, providing a standardized data foundation for model training, optimization, and performance evaluation. The dataset splitting strategy of this experiment is exhibited in Table 1.

When conducting the proposed expression localization and recognition model's application experiment, ESR, LBP, MDM, CNN, and the baseline model are selected as comparison objects. This selection aims to cover different representatives from traditional feature methods to DL methods, thus ensuring the evaluation's comprehensiveness. Among them, the Baseline model is a lightweight reference model, which consists of two convolution blocks and two fully connected layers. It takes face images as input and outputs key points or expression categories. The training adopts the Adam optimizer and basic data augmentation. Its performance is between traditional methods and CNN, providing a standard baseline for evaluating the MDMO + Transformer model.

The basic configuration of the experiment is outlined in Table 2:

Table 2: Experimental configuration

Item	Configuration instructions
Hardware environment	CPU: Intel Xeon Gold 6226R, RAM: 128 GB
Training epoch	50 epochs
Optimizer	Adam optimizer
Initial learning rate	0.001, adjusted based on performance during training
Batch size	32
Loss function for facial expression localization task	Mean squared error
Loss function for the FER task	Cross-entropy loss

The results of the ablation experiment in Table 3 indicate that the MDMO module mainly contributes to the accuracy of expression localization, and the Transformer module mainly improves the accuracy of expression recognition. Combining the two can provide stable and efficient expression analysis capabilities in the MHADT scenario.

Table 3: Expression localization and recognition ablation experiment based on MDMO and transformer

Model version	Localization accuracy (%)	Localization processing time (milliseconds (ms))	Recognition accuracy (%)	Recognition processing time (ms)
Only MDMO	81.5	52	80.2	55
Only Transformer	84	50	83	50
MDMO + Transformer	89.6	55	88.9	44

For different expression localization models' error comparison, Fig. 5 shows distinct differences in the performance of each model on the training, validation, and test subsets. Traditional methods such as ESR and LBP have relatively high errors on the training subset and show substantial fluctuations on the validation subset, which indicates their limited generalization ability. MDM and CNN perform better on the validation and test subsets, but there are still certain errors. The baseline model outperforms most traditional methods on the validation subset, but its overall stability is insufficient. By contrast, the proposed model integrating MDMO and Transformer achieves the lowest error rates across all three subsets, demonstrating a marked improvement over alternative methods, particularly on the validation subset. This demonstrates its comprehensive advantages in model generalization, stability, and accuracy.

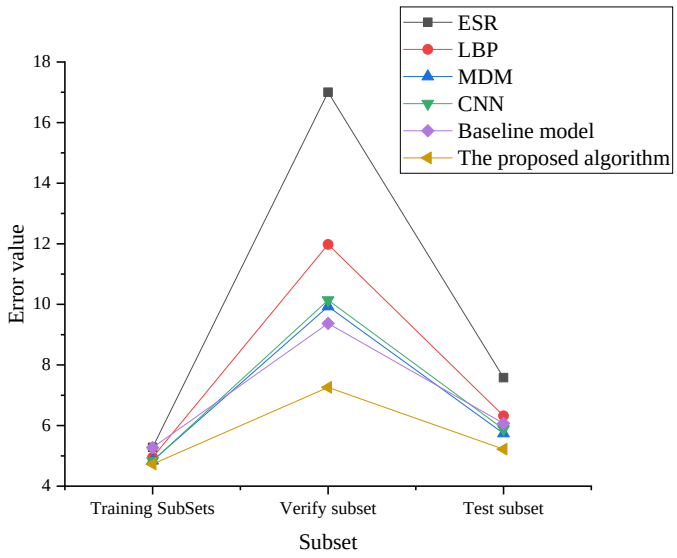


Figure 5: Error comparison of various expression localization models

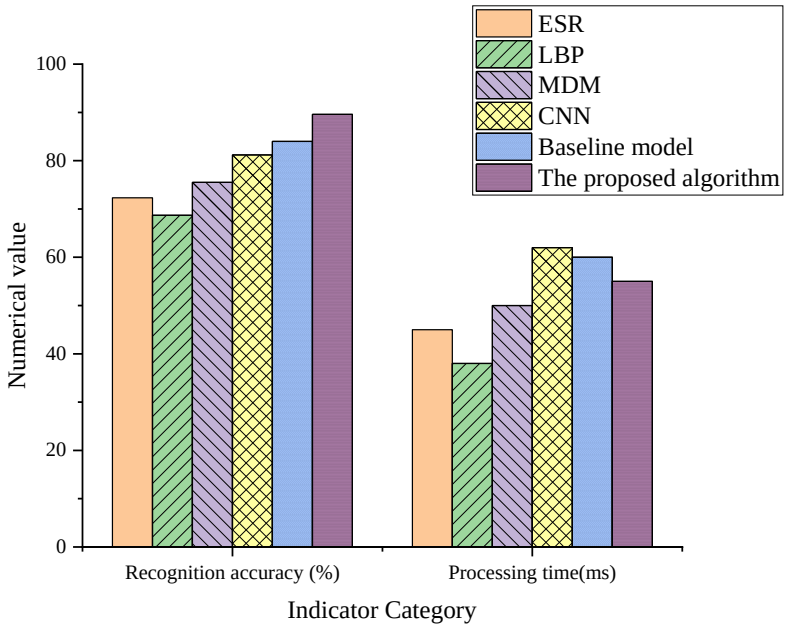


Figure 6: Comparison of localization accuracy and image processing speed of diverse expression localization models

Fig. 6 reveals that in comparing different expression localization models, there are certain differences in localization accuracy and image processing speed. Among traditional methods, LBP has the lowest accuracy, only 68.7%, while MDM has a relatively higher accuracy of 75.5%. CNN and the baseline model further improve the accuracy, reaching 81.2% and 84%, respectively. The model proposed here achieves the highest localization accuracy of 89.6%, notably better than other methods. Regarding processing speed, LBP and ESR take less time, while CNN and the baseline model take relatively longer. The proposed model's processing speed reaches 55 ms, slightly longer than some traditional methods; however, it greatly improves the localization accuracy while maintaining high real-time performance. This shows the optimal balance between precision and efficiency, making it more in line with the dual requirements of accuracy and usability in MHADT.

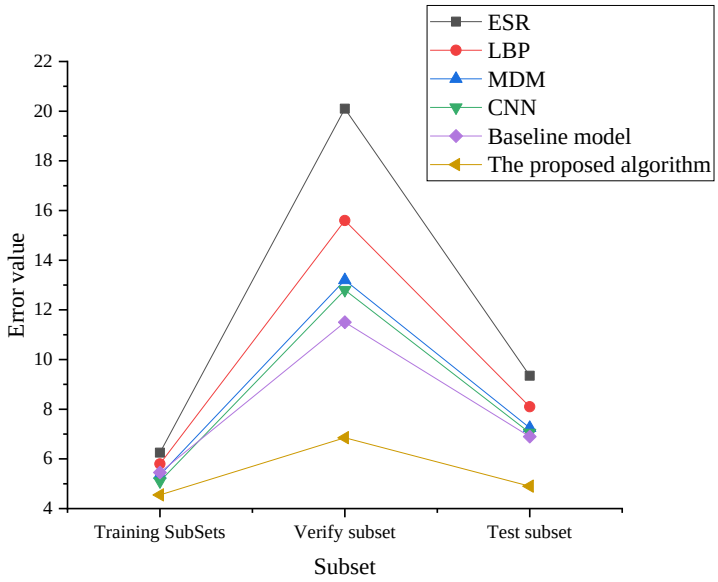


Figure 7: Error comparison of different FER models

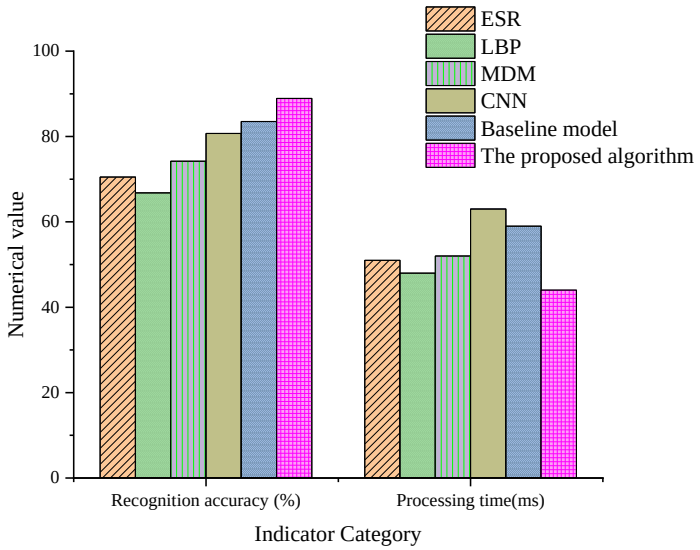


Figure 8: Comparison of recognition accuracy and image processing speed of each FER model

Fig. 7 demonstrates substantial differences in error performance across validation, test, and training subsets among diverse FER models. Traditional methods like ESR and LBP exhibit relatively high errors on validation and test subsets, illustrating limited generalization ability. Although MDM and CNN achieve lower errors on training and test subsets, a considerable error gap persists in the validation subset. The baseline model has lower errors on validation and test subsets, exhibiting greater stability than traditional methods. In contrast, the proposed model achieves the lowest errors across all three subset types. Specifically, its validation subset error is only 6.85, substantially lower than other methods. These results fully confirm the model's remarkable advantages in recognition accuracy and generalization capability, thereby furnishing more reliable technical support for MHADT.

Fig. 8 reveals noticeable differences in image processing speed and recognition accuracy when comparing various FER models. Among traditional methods, LBP achieves the lowest recognition accuracy at merely 66.8%, whereas MDM attains a relatively higher accuracy of 74.2%. CNN and the baseline model demonstrate improvements, reaching 80.7% and 83.5%, respectively. The MDMO- and Transformer-based model introduced in this study attains the highest recognition accuracy of 88.9%, prominently outperforming all comparative methods. Concerning processing speed, the

proposed model requires only 44ms, which is faster than CNN and the baseline model, while superior to traditional approaches such as ESR, LBP, and MDM. These results illustrate the model's dual advantages in real-time performance and recognition accuracy, aligning well with the integrated demands for high accuracy and efficiency in MHADT scenarios.

Table 4 shows that the proposed algorithm achieves the lowest model error on the training, validation, and test sets. The localization and recognition accuracy reach 89.6% and 88.9% respectively, remarkably exceeding ESR, LBP, MDM, CNN, and the baseline model. At the same time, it maintains high efficiency in image processing time, with the best comprehensive performance.

Table 4: Performance comparison of different models

Performance	Evaluation indicators	ESR	LBP	MDM	CNN	Baseline model	The proposed algorithm
Error of the facial expression location model	Training SubSets	5.28	4.95	4.84	4.83	5.27	4.73
	Verify subset	17	11.98	9.93	10.14	9.37	7.26
	Test subset	7.58	6.32	5.74	5.88	6.07	5.22
Localization accuracy and image processing time	Localization accuracy (%)	72.3	68.7	75.5	81.2	84	89.6
	Processing time(ms)	45	38	50	62	60	55
Error of the FER model	Training SubSets	6.25	5.8	5.35	5.1	5.45	4.55
	Verify subset	20.1	15.6	13.2	12.8	11.5	6.85
	Test subset	9.35	8.1	7.25	7.05	6.9	4.9
Recognition accuracy and image processing time	Recognition accuracy (%)	70.5	66.8	74.2	80.7	83.5	88.9
	Processing time(ms)	51	48	52	63	59	44

Table 5 indicates that in the experiment comparing CNN and Baseline models, the results of the statistical significance test show that the proposed model combining MDMO and Transformer achieves significant improvements in both localization and recognition accuracy. Specifically, the facial expression localization and recognition accuracies reach 89.6% and 88.9%. The errors on the validation and test subsets are reduced to 5.22 and 4.90, showing statistically significant differences compared with the CNN and Baseline models ($p < 0.05$ or $p < 0.01$). At the same time, the image processing time is only 44–55 ms, which is also significantly better than the comparison models, balancing real-time performance and high precision.

5 Discussion

Compared with existing studies, the proposed expression localization and recognition model based on MDMO and Transformer shows remarkable performance advantages in the MHADT scenario. Compared with the multi-modal emotion recognition model proposed by Zhang et al., with a recognition accuracy of about 82.1%, the proposed model achieves a recognition accuracy of 88.9% on the AVEC2019 DDS dataset. Meanwhile, the inference time is only 44 ms, achieving higher emotion recognition accuracy while maintaining high real-time performance. The main reasons for the performance improvement are as follows. First, the MDMO feature extraction module can effectively capture the dynamic changes and directional movement information of facial expressions, enhancing the expression of temporal features. Second, the Transformer's global attention mechanism effectively captures long-range dependencies within expression sequences. This capability enables cross-frame information integration and contextual feature enhancement, substantially improving recognition robustness and generalization performance. Zhang et al.'s method enhances overall emotion recognition performance by incorporating multimodal features, including voice, image, and text data. However, the method remains constrained by its reliance on local convolutional features within the visual modality alone. This architectural limitation restricts its capacity to model temporal dynamics, leading to reduced accuracy in scenarios involving complex expression variations.

Table 5: A comparative statistical analysis of the proposed algorithm with CNN and baseline model

Performance indicators	Comparison model	Mean value of the proposed algorithm	Mean value of comparison models	Average difference (Δ)	t-value	p-value	Significance level
Localization accuracy (%)	CNN	89.6	81.2	8.4	3.27	0.014	*(p<0.05)
	Baseline model	89.6	84	5.6	2.89	0.022	*(p<0.05)
Localization accuracy (Test)	CNN	5.22	5.88	-0.66	-2.57	0.028	*(p<0.05)
	Baseline model	5.22	6.07	-0.85	-3.04	0.019	*(p<0.05)
Recognition accuracy (%)	CNN	88.9	80.7	8.2	3.64	0.011	*(p<0.05)
	Baseline model	88.9	83.5	5.4	3.11	0.018	*(p<0.05)
Recognition error (Test)	CNN	4.9	7.05	-2.15	-3.78	0.009	**(p<0.01)
	Baseline model	4.9	6.9	-2	-3.51	0.012	*(p<0.05)
Image processing time (ms)	CNN	44	63	-19	-4.12	0.007	**(p<0.01)
	Baseline model	44	59	-15	-3.56	0.011	*(p<0.05)

Although the proposed model performs well in multiple experimental indicators, it still has certain limitations. First, the AVEC2019 DDS dataset has a small scale and uneven distribution of expression categories. Insufficient samples may lead the model to overfit on certain specific expression categories (such as negative or complex emotions), thereby affecting generalization ability. It may also perform inconsistently among different patient groups. For example, noisy or partially occluded faces, age, gender, or cultural differences may cause some expressions to be easily misclassified. This suggests that the model's results need to be interpreted carefully in clinical applications. Second, the model only uses facial images for recognition; it does not fully integrate multi-source information such as voice or physiological signals, which may limit its performance in multi-scenario psychological assessment. Third, the application of MHADT must focus on patients' informed consent, data privacy protection, and potential demographic biases; therefore, it is necessary to ensure the model's fairness and reliability among different populations.

6 Conclusion

The proposed MDMO- and Transformer-based expression localization and recognition model has undergone systematic application experiments in the MHADT scenario; this model has also been compared with ESR, CNN, MDM, LBP, and the baseline model from multiple dimensions. Concerning expression localization, the model attains the lowest errors on the training, validation, and test subsets, with a localization accuracy of 89.6%, markedly outperforming ESR, LBP, MDM, CNN, and the baseline model. Although its processing speed is 55ms, which is slightly more than that of some traditional methods, it maintains high real-time performance while substantially improving precision. Regarding FER, the model also has the lowest errors on all subsets, with an error of only 6.85 on the validation subset. Its recognition accuracy reaches 88.9%, and the processing time is only 44ms. This is shorter than the baseline and CNN models while surpassing traditional methods like ESR, LBP, and MDM, demonstrating its dual benefits in recognition accuracy and real-time processing capability. The proposed model can efficiently and accurately capture subtle changes in patients' facial expressions. Besides, it can realize reliable assessment of emotional states and mental health indicators, and provide a scientific basis for clinical psychological intervention, treatment course monitoring, and psychological therapy effect tracking. At the same time, the model performs excellently in both processing speed and recognition accuracy, balancing real-time performance and stability, and can support expression analysis for continuous videos or interviews. In addition to the mental health field, the proposed model is also applicable to scenarios such as intelligent HCI, educational assessment, and behavior monitoring. It can provide high-precision technical support for emotion computing, interactive feedback, and behavior analysis. The model demonstrates comprehensive advantages in accuracy, efficiency, and generalization ability, laying a

solid foundation for the application of FER technology in multiple fields.

Future work can draw on ideas such as adaptive inversion control, nonlinear optimal control, and adaptive backstepping control to introduce a more robust adaptive mechanism. Thus, the model's stability and generalization ability can be improved under noisy, occluded, and incomplete data conditions. At the same time, through multi-modal fusion and transfer learning strategies, the model's applicability and reliability in different patient groups and complex clinical scenarios can be further enhanced.

Acknowledgment: This study is supported by Foundation of Guangdong Educational Committee (No. 2025WTSCX200); Research Project on Mental Health Education in Higher Vocational Education (No. XL2025122); Humanity and Social Science foundation of Ministry of Education (No.25JDSZ3164); Foundation of Guangdong Polytechnic of Industry and Commerce (No. 2025-PT-01).

References

[1] Zhou Y, Chen L, Huang T, et al. Dual complementarity transformer for micro-expression recognition. Multimedia Systems, 2025, 31(5): 371. https://doi.org/10.1007/s00530-025-01960-w

[2] Ma Y, Shen J, Zhao Z, et al. What can facial movements reveal? Depression recognition and analysis based on optical flow using Bayesian networks. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2023, 31: 3459-3468. https://doi.org/10.1109/tnsre.2023.3305351

[3] Muetunda F, Sabry S, Jamil M L, et al. AI-Assisted diagnosing, monitoring and treatment of mental disorders: A survey. ACM Transactions on Computing for Healthcare, 2025, 5(4): 1-24. https://doi.org/10.1145/3681794

[4] Bao Y, Wu C, Zhang P, et al. Boosting micro-expression recognition via self-expression reconstruction and memory contrastive learning. IEEE Transactions on Affective Computing, 2024, 15(4): 2083-2096. https://doi.org/10.1109/TAFFC.2024.3397701

[5] Wimbarti S, Kairupan B H R, Tallei T E. Critical review of self-diagnosis of mental health conditions using artificial intelligence. International Journal of Mental Health Nursing, 2024, 33(2): 344-358. https://doi.org/10.1111/inm.13303

[6] Li Y, Liu M, Lao L, et al. Counterfactual discriminative micro-expression recognition. Visual Intelligence, 2024, 2(1): 29. https://doi.org/10.1007/s44267-024-00063-w

[7] Liu Y, Zhang X, Li Y, et al. Graph-based facial affect analysis: A review. IEEE Transactions on Affective Computing, 2022, 14(4): 2657-2677. https://doi.org/10.1109/TAFFC.2022.3215918

[8] Nidhi, Verma B. From methods to datasets: a detailed study on facial emotion recognition. Applied Intelligence, 2023, 53(24): 30219-30249. https://doi.org/10.1007/s10489-023-05052-y

[9] Pereira R, Mendes C, Ribeiro J, et al. Systematic review of emotion detection with computer vision and deep learning. Sensors, 2024, 24(11): 3484. https://doi.org/10.3390/s24113484

[10] Jabbooree A I, Khanli L M, Salehpour P, et al. A novel facial expression recognition algorithm using geometry β–skeleton in fusion based on deep CNN. Image and Vision Computing, 2023, 134: 104677. https://doi.org/10.1016/j.imavis.2023.104677

[11] Kim S, Nam J, Ko B C. Facial expression recognition based on squeeze vision transformer. Sensors, 2022, 22(10): 3729. https://doi.org/10.3390/s22103729

[12] Zhang F, Chai L. A review of research on micro-expression recognition algorithms based on deep learning. Neural Computing and Applications, 2024, 36(29): 17787-17828. https://doi.org/10.1007/s00521-024-10262-7

[13] Zhang P, Wang R, Luo J, et al. Micro-expression recognition algorithm using regions of interest and the weighted ArcFace loss. Electronics, 2024, 14(1): 2. https://doi.org/10.3390/electronics14010002

[14] Zhao S, Tang H, Mao X, et al. Dfme: A new benchmark for dynamic facial micro-expression recognition. IEEE Transactions on Affective Computing, 2023, 15(3): 1371-1386. https://doi.org/10.1109/TAFFC.2023.3341918

[15] Zhang L, Hong X, Arandjelović O, et al. Short and long range relation based spatio-temporal transformer for micro-expression recognition. IEEE Transactions on Affective Computing, 2022, 13(4): 1973-1985. https://doi.org/10.1109/TAFFC.2022.3213509

[16] Pan H, Xie L, Wang Z. C3DBed: Facial micro-expression recognition with three-dimensional convolutional neural network embedding in transformer model. Engineering Applications of Artificial Intelligence, 2023, 123: 106258. https://doi.org/10.1016/j.engappai.2023.106258

[17] Tian Y, Zhu J, Yao H, et al. Facial expression recognition based on vision transformer with hybrid local attention. Applied Sciences, 2024, 14(15): 6471. https://doi.org/10.3390/app14156471

[18] Zhang X, Li M, Lin S, et al. Transformer-based multimodal emotional perception for dynamic facial expression recognition in the wild. IEEE Transactions on Circuits and Systems for Video Technology, 2023, 34(5): 3192-3203. 10.1109/TCSVT.2023.3312858

[19] Indolia S, Nigam S, Singh R, et al. Micro expression recognition using convolution patch in vision transformer. IEEE Access, 2023, 11: 100495-100507. https://doi.org/10.1109/ACCESS.2023.3314797

[20] Qin L, Wang M, Deng C, et al. Swinface: a multi-task transformer for face recognition, expression recognition, age estimation and attribute estimation. arXiv:2308.11509. https://doi.org/10.48550/arXiv.2308.11509

[21] Dai M, Hu J, Zhuang J, et al. A transformer-based feature segmentation and region alignment method for UAV-view geo-localization. IEEE Transactions on Circuits and Systems for Video Technology, 2021, 32(7): 4376-4389. https://doi.org/10.1109/TCSVT.2021.3135013

[22] Ma F, Sun B, Li S. Transformer-augmented network with online label correction for facial expression recognition. IEEE Transactions on Affective Computing, 2023, 15(2): 593-605. https://doi.org/10.1109/TAFFC.2023.3285231

[23] Aloysius N, Geetha M, Nedungadi P. Incorporating relative position information in transformer-based sign language recognition and translation. IEEE Access, 2021, 9: 145929-145942. https://doi.org/10.1109/ACCESS.2021.3122921

[24] Islam M S, Sang Y, Mohammed A A Q, et al. Facial micro-expression recognition from videos through domain adaptation and multi-modal spatio-temporal feature ensemble: Facial micro-expression. Multimedia Systems, 2025, 31(5): 330. https://doi.org/10.1007/s00530-025-01898-z

[25] Munanday A P, Sazali N, Asogan A, et al. The implementation of transfer learning by convolution neural network (CNN) for recognizing facial emotions. Journal of Advanced Research in Applied Sciences and Engineering Technology, 2023, 32(2): 255-276. https://doi.org/10.37934/araset.32.2.255276

A Generative AI-Driven Framework for Human–AI Collaborative Teaching: Design, Implementation, and Empirical Evaluation

Zhijuan Wang
Jiangxi Technical College of Manufacturing, Nanchang 330000, Jiangxi, China
E-mail: 19979087987@163.com

Keywords: generative artificial intelligence, human–ai collaboration, teaching model, personalized learning, educational digitalization

Received: September 11, 2025

The rapid development of Generative Artificial Intelligence (GAI) technologies is driving profound changes in education. This study investigates the construction of human–Artificial Intelligence (AI) collaborative teaching models supported by GAI, with the goals of improving teaching efficiency, enabling personalized learning, and optimizing the allocation of educational resources. The proposed framework integrated intelligent content generation, real-time tutoring, adaptive learning pathways, and a teacher–student–AI collaboration mechanism. The generative model employed was LLaMA-2-13B, which was domain-adapted through supervised fine-tuning (SFT) and reinforcement learning from human feedback (RLHF). The experiment was conducted in a university course titled Data Structures and Algorithms, involving 60 students in the experimental group and 60 students in the control group. Multi-dimensional data were collected and analyzed, including academic performance, student engagement, interaction depth, technology acceptance, and long-term retention. The quality of AI-generated content was evaluated using Bilingual Evaluation Understudy (BLEU, 0.74) and Recall-Oriented Understudy for Gisting Evaluation (ROUGE, 0.79), with a Cohen's κ value of 0.86 indicating high inter-rater consistency. The results showed that the GAI-driven human–AI collaborative model significantly improved final exam scores (85.6 vs. 78.3, $p < 0.001$), average assignment grades (91.2 vs. 84.7, $p < 0.001$), and learning satisfaction ($p < 0.05$), while reducing cognitive load and enhancing personalized and interactive learning. This study provides both a theoretical framework and practical guidance for innovating educational models in the era of intelligent technology, offering valuable insights for advancing the digital transformation of education.

Povzetek: Študija raziskuje GAI-podprt model sodelovalnega poučevanja za izboljšanje personaliziranega učenja in učne učinkovitosti.

1 Introduction

Currently, Generative Artificial Intelligence (GAI) technologies, represented by Large Language Models (LLMs), are reshaping human society at an unprecedented pace. As a cornerstone of social development, education plays a central role in this ongoing educational and societal change [1–3]. GAI, particularly large models trained on massive datasets with strong language understanding and generation capabilities, has demonstrated potential far beyond traditional Artificial Intelligence (AI) tools. These models can accurately comprehend complex semantics and creatively generate text, code, images, and even multimodal content, enabling natural and fluent interactions with humans. This technological advancement introduces AI-driven capabilities into education, facilitating the transition of teaching models from teacher-centered knowledge delivery toward learner-centered, personalized, interactive, and adaptive approaches. Against this backdrop, exploring how to effectively leverage GAI technologies to construct innovative teaching models has become a global frontier in educational research and practice [4–6].

Traditional teaching methods, while advantageous in systematic knowledge transmission, face several structural challenges. First, personalized learning is difficult to achieve. In conventional classroom settings, it is challenging for teachers to provide precise, differentiated instruction tailored to each student's unique knowledge base, learning style, interests, and cognitive pace, often resulting in the phenomenon of "advanced students are under-challenged, while struggling students fall behind." Second, teaching resources and teacher capacity are limited. Developing high-quality teaching materials (e.g., case studies, exercises, and supplementary resources) is time-consuming, while repetitive tasks such as grading and answering questions occupy a substantial portion of teachers' time, restricting their ability to focus on creative instructional design, deep interactions, and emotional support. Third, feedback is often insufficient in timeliness and depth. Students' questions or difficulties arising during the learning process frequently do not receive immediate, personalized guidance, which negatively impacts learning efficiency and motivation.

Although recent applications of Intelligent Tutoring System (ITS) and learning analytics have partially alleviated these challenges, their core operations still rely on preset rules or statistical models. These systems offer relatively rigid interaction methods and limited content generation capabilities, lacking true understanding and creative potential, making them inadequate for open-ended, complex learning scenarios [7–9]. The emergence of GAI provides a novel technological approach to address these limitations, functioning as an intelligent teaching collaborator capable of supporting and enhancing the instructional process.

However, technological potential does not automatically translate into educational reality. Current research has largely focused on applying GAI to individual teaching tasks (e.g., automated grading or question-answering), lacking the construction of systematic human–AI collaborative teaching models. Key questions remain underexplored: How can humans and AI be scientifically assigned roles and collaborate dynamically across the full teaching process? How can effective collaboration mechanisms be designed to ensure the achievement of instructional goals and the enhancement of educational quality? How should the responsibility boundaries of AI-generated content be defined to uphold academic integrity? And how can technology be leveraged without fostering student overreliance, which might inhibit critical thinking and creativity? These questions have yet to receive sufficient theoretical exploration or large-scale empirical validation.

Therefore, this study aims to systematically construct a Generative AI–supported Human–Machine Collaborative Teaching Model (GAI-HMCTM). The model seeks to integrate GAI's capabilities in intelligent content generation, real-time feedback, and adaptive learning with the core strengths of human teachers in instructional design, emotional support, value guidance, and complex decision-making, forming a "teacher-led, AI-empowered" collaborative symbiosis. Based on the considerations above, this study formulated the following research question: In university courses, can the GAI-driven human–AI collaborative teaching model (GAI-HMCTM) significantly enhance students' academic performance, engagement, and satisfaction with personalized learning compared with traditional teaching methods? Although the study focused on a computer science course as the experimental context, the model was designed with cross-disciplinary adaptability, providing a foundation for future validation across diverse educational settings. This research proposes a theoretical framework and conducts an empirical study to provide a solution that combines theoretical rigor with practical value. It aims to drive innovation in educational models in the era of intelligent technologies. In doing so, it also seeks to facilitate a deeper and higher-quality digital transformation in education.

2 Related works

2.1 Applications of GAI in education

In recent years, research on the application of GAI in education has grown rapidly, with use cases expanding in scope and depth [10–12]. Cheong showed that GAI could automatically generate diverse practice exercises, formative assessments, and comprehensive exam questions tailored to specific knowledge points, difficulty levels, and learning objectives. This significantly improved the efficiency and flexibility of educational resource development [13]. GAI can also produce summaries of learning materials, textual descriptions for knowledge graphs, and code examples with debugging suggestions for different programming languages, supporting programming instruction. When integrated with ITS, GAI offers a more natural, fluent, and human-like conversational tutoring experience. Students can engage in multi-turn dialogues, explore complex concepts, receive personalized problem-solving guidance, and obtain immediate feedback and encouragement. This addresses limitations of traditional ITS, which often rely on rigid interactions and single-response patterns [14–16]. Ding and Zou further highlighted GAI's potential in supporting student learning. In writing instruction, it can provide grammar correction, structural optimization, style suggestions, and creative inspiration [17]. In language learning, GAI can create immersive conversational practice scenarios and provide real-time feedback on pronunciation and expression. These applications enhance interactivity and engagement, motivating students while improving learning efficiency and autonomous learning through immediate, personalized support. However, many applications remain in exploratory or pilot stages. Key challenges—such as content accuracy, reliability, potential bias, and impacts on academic integrity—require continued attention and resolution.

2.2 Human–AI collaborative teaching theory and practice

Human–AI collaboration is a key approach for integrating AI into social applications. Its central principle is to leverage the complementary strengths of human and machine intelligence, rather than simply replace humans [18–20]. Zhao et al. emphasized that, in education, collaboration theory advocated a "symbiotic" relationship: human teachers should capitalize on their unique advantages in emotional support, value guidance, instructional decision-making in complex contexts, creative course design, and handling unstructured or open-ended problems [21]. Meanwhile, AI systems should focus on tasks they excel at, such as efficiently delivering knowledge, processing large datasets, performing repetitive work, providing 24/7 feedback, and recommending personalized learning paths and resources. This division of labor aims to free teachers to concentrate on higher-order educational goals. Previous studies have explored roles like "AI teaching assistants" and "Intelligent Pedagogical Agents," assisting with class

management, standardized feedback, or simple question-answer interactions. However, these early implementations relied heavily on preset rules, expert systems, or early machine learning models. As a result, their functionality was fixed, interaction patterns rigid, and natural language comprehension limited. They lacked the ability to generate new content, engage in multi-turn deep dialogue, or solve problems creatively, making them insufficient for dynamic, complex teaching scenarios [22]. The emergence of Generative AI provides a foundation for deeper, more flexible, and more "intelligent" human–AI collaboration. By enabling AI to act as a collaborator rather than merely a tool, GAI supports the evolution of human–AI collaborative teaching theory to a new stage of development.

2.3 Limitations of existing research

Table 1 presents a comparison of the literature. As shown, despite the promising prospects of GAI in education, existing research has several clear limitations. First, applications are often fragmented and single-purpose (e.g., limited to automated grading or question-answering) and lack systematic integration of GAI capabilities across the entire teaching process. (2) Core aspects of human–AI collaboration, such as how tasks are dynamically allocated, how responsibility boundaries between humans and AI are defined, and how interaction workflows are designed for efficient collaboration, remain underexplored, resulting in insufficient theoretical depth. (3) Empirical studies are mostly small-scale pilots or case studies, lacking large-sample, long-term controlled experiments. Consequently, evaluations of the long-term effects, sustainability, and potential risks of teaching models—such as challenges to academic integrity, student overreliance on AI, diminished critical thinking, data privacy concerns, and algorithmic bias—are inadequate. (4) Existing explorations are often limited to specific scenarios, lacking a universal and scalable theoretical framework and practical guidelines that can adapt to different disciplines and educational stages.

Therefore, there is an urgent need to develop a human–AI collaborative teaching model that is integrative, clearly structured, empirically validated, and highly adaptable. Such a model would systematically guide the deep application of GAI in education. This study aims to address these gaps by constructing a more systematic, operational, and generalizable human–AI collaborative teaching model and empirically validating its effectiveness. The proposed GAI-driven human–AI collaborative teaching model (GAI-HMCTM) integrates content generation, personalized learning pathways, real-time interaction, and teacher collaboration. This integration addresses the limitations of existing studies in terms of systematic design, practical applicability, and scalability, highlighting both the necessity and innovative contribution of the present research.

Table 1: Comparison of GAI-related studies in education

Reference	Method	Application Domain	Evaluation Metrics	Key Findings	Limitations
Cheong [13]	GAI content generation	Exercises and exams across multiple courses	Assignment completion rate, quantity of generated content	Improved efficiency and flexibility in resource development	Focused solely on content generation; lacked classroom interaction and teacher collaboration mechanisms
Ding & Zou [17]	GAI-assisted writing and language learning	Writing and language instruction	Essay scores, language accuracy	Enhanced student autonomy and engagement	Single-function; small-scale experiments; lacked systematic integration
Cooper [21]	Human–AI collaboration theory and intelligent educational agents	Teacher support and classroom management	Teaching observation, interaction log analysis	Provided teacher assistance; reduced repetitive work	Depended on rule-based or early ML models; fixed interaction patterns; lacked content generation and multi-turn conversational capability

3 Methods

3.1 Research framework design

Based on literature analysis, educational theory, and the principles of human–AI collaboration, this study developed the GAI-HMCTM. The framework is designed to achieve deep collaboration between teachers and GAI, forming an integrated system in which teachers lead in instructional design and value guidance, while GAI empowers execution and personalized support. The framework consists of five core layers:

1. Intelligent Content Generation Layer (ICGL): This layer employs a fine-tuned generative large model (e.g., LLaMA-2) as its core engine. Teachers provide instructional inputs, including teaching objectives (T_i), knowledge points (K_j), student profiles (P_s), and difficulty requirements (D). The GAI model then generates the corresponding teaching content. The content

generation process can be formalized as:

$$C_{gen} = G(T_i, K_j, P_s, D; \theta) \quad (1)$$

where C_{gen} represents the generated content, G denotes the generation function, and θ represents the model parameters. The generated content includes teaching PPTs, case analyses, tiered exercises (basic, intermediate, advanced), and summaries of supplementary reading materials. To ensure content quality and accuracy, a teacher review mechanism is incorporated. Teacher feedback (F_t) is used to fine-tune the model or correct content:

$$C_{final} = \begin{cases} C_{gen} \, if \; Q(C_{gen}) \geq Q \\ Revise(C_{gen}, F_t) thotherwise \end{cases} \quad (2)$$

where x denotes a content quality evaluation function (which can combine manual scoring and automatic metrics such as Bilingual Evaluation Understudy (BLEU) and Recall-Oriented Understudy for Gisting Evaluation (ROUGE), and x represents the quality threshold.

2. Personalized Learning Path Layer (PLPL): Based on learning analytics, a student knowledge state model is constructed. The mastery level MST of student s on knowledge point k is defined as:

$$M_{s,k} = f(correct_{s,k}, time_{s,k}, attempts_{s,k} \quad (3)$$

where $correct_{s,k}$ is the accuracy rate, $time_{s,k}$ is the average problem-solving time, $attempts_{s,k}$ is the number of attempts, and f is the comprehensive evaluation function. GAI dynamically generates personalized learning paths (L_s) according to $M_{s,k}$:

$$L_s = PathGen(M_s, T_i, R_s) \quad (4)$$

M_s represents the student's overall knowledge mastery vector, T_i denotes the learning objectives, and RS indicates student resource preferences. Path recommendation algorithms can be based on collaborative filtering or knowledge graph traversal:

$$Score(r_i) = \alpha \cdot Sim(r_i, R_s) + \beta \cdot Rel(r_i, K_{gap}) + \gamma \cdot Diff(r_i) \quad (5)$$

where $Score(r_i)$ is the recommendation score for resource r_i , Sim denotes similarity to student preferences, Rel represents relevance to knowledge gaps (K_{gap}), $Diff$ is difficulty suitability, and α , β , γ are weights.

3. Real-time Feedback & Tutoring Layer (RFTL): This layer integrates GAI chatbots to provide immediate interactive support. Student queries (QS) are input to the model to generate responses (AGEN):

$$Agen = G(Q_s, C_{context}; \theta) \quad (6)$$

where $C_{context}$ includes relevant context, such as current learning content and the student's history. To improve response reliability, retrieval-augmented generation (RAG) is employed:

$$A_{gen} = G(Q_s, Retrieve(Q_s, KB); \theta) \quad (7)$$

where $Retrieve$ retrieves relevant information from the knowledge base (KB). The system evaluates response confidence ($Conf(A_{gen})$):

$$Conf(A_{gen}) = h(Entropy(Top-k\,outputs)) \quad (8)$$

If confidence falls below the threshold C_{th} , the system prompts, "This question is complex; please consult the teacher," or routes the query to a human instructor.

4. Teacher–AI Collaboration Layer (TACL): A bidirectional collaboration mechanism is established. Teachers can view class-wide learning heatmaps (H) via a dashboard:

$$H_{i,j} = g(M_{class,kj}, Engagement_{class,topici}) \quad (9)$$

where $H_{i,j}$ reflects the mastery and engagement for the *j*th knowledge point under the *i*th topic. Teachers adjust instructional focus accordingly. AI supports teachers by providing lesson preparation suggestions and preliminary grading of assignments (flagging potential errors and providing scoring suggestions, S_{AI}):

$$S_{AI} = w_1 \cdot Correctness + w_2 \cdot Completeness + w_3 \cdot Logic \quad (10)$$

Teachers conduct the final review and scoring (S_{final}), and their feedback is used to optimize the AI grading model. Teachers may also define AI "behavior boundaries" (e.g., prohibiting direct answer provision, limiting AI to hints only).

5. Evaluation & Optimization Layer (EOL): This layer integrates formative and summative assessments. GAI analyzes student learning process and outcome data to calculate a comprehensive learning effectiveness index (I_{perf}):

$$I_{perf} = \delta \cdot ExamScore + ò \cdot AssignmentAvg + \zeta \cdot EngagementIndex \quad (11)$$

where δ, ò, ζ are weights. Model performance is evaluated using teacher satisfaction (S_t), student satisfaction (S_s), task completion rate (C_r), and other indicators:

$$I_{model} = \eta \cdot S_t + \theta \cdot S_s + \iota \cdot C_r \quad (12)$$

Regular evaluations (I_{model}) drive iterative optimization of the framework.

The architecture of the human–AI collaborative teaching model is illustrated in Figure 1.

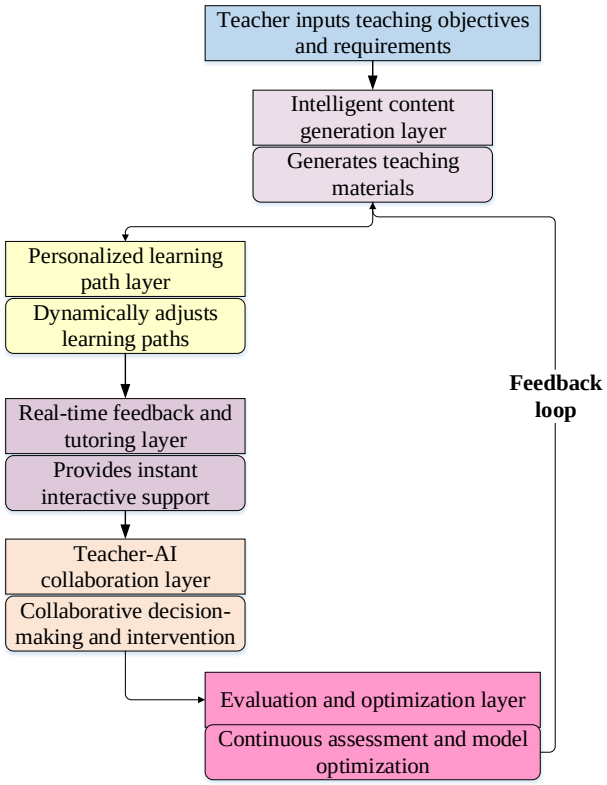


Figure 1: Architecture of the human–AI collaborative teaching model

3.2 Participants and study design

The study implementation process is illustrated in Figure 2.

Participants: Two second-year natural classes from the Computer Science and Technology program at a "Double First-Class" university were selected, comprising a total of 120 students. A pre-test and background survey were conducted before the experiment, including students' previous semester grades in *Introduction to Programming*, learning habits, and technology usage. The results showed no significant differences between the two classes at baseline (independent-samples t test, $p > 0.05$), ensuring the comparability of the groups. The experimental group (n = 60) adopted the GAI-HMCTM model, while the control group (n = 60) followed a traditional lecture-based approach supplemented with the school's online question bank, including automated grading for objective questions. The course was *Data Structures and Algorithms*, conducted over 16 weeks with four hours of instruction per week.

Study Design: A quasi-experimental design was employed, grouping students by class. Variables such as teacher (same instructor), textbook, and assessment criteria were controlled. The study spanned one semester.

GAI Implementation: The open-source large model LLaMA-2-13B was fine-tuned for the domain using supervised fine-tuning (SFT) and reinforcement learning from human feedback (RLHF) on corpora including *Data Structures* textbooks, exercises, and academic papers. To ensure the model possessed domain specificity and teaching adaptability, the training corpus included approximately 1.2 million tokens from data structures course textbooks, exercise collections, and relevant academic papers. The materials covered fundamental concepts, algorithm implementations, programming examples, and exercise solutions, ensuring comprehensive coverage of the course's core knowledge.

During fine-tuning, a batch size of 32, a learning rate of 2×10^{-5}, and 5 training epochs were used, along with a mixed-precision optimization strategy to improve computational efficiency. For content generation and tutoring, structured prompt templates were employed. These templates followed an "explanation → example → question" format to support stable performance in generating instructional content, answering questions, and providing feedback. Specific templates included exercise generation prompts, code debugging prompts, and case explanation prompts, ensuring that the outputs aligned with course objectives.

To prevent data leakage between training and testing, the experiment strictly separated classes: one class provided training data, while the other served as the test and validation group. During preprocessing, duplicate entries and potentially overlapping text were removed, ensuring fair and reliable model evaluation. A dedicated web plugin was developed to enable seamless integration with the university's learning management system (LMS), facilitating student access and interaction. All AI interactions were automatically recorded for subsequent behavior analysis and outcome tracking.

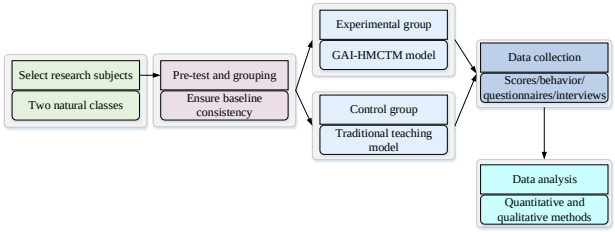


Figure 2: Research implementation process

To ensure that GAI-HMCTM could operate efficiently within a teaching environment, this study deployed the model within the LMS framework and developed an independent plugin module to enable dynamic interaction with instructional content and real-time data feedback. The plugin consisted of three core functional modules: (1) Content Delivery Module: Responsible for automatically pushing personalized learning resources—including textual explanations, quizzes, and multimodal materials—onto the LMS interface based on the instructional plan and model-generated outputs. (2) Learning Behavior Capture Module: Utilizes the Application Programming Interface (API) to record students' operational behaviors in real time,

including task completion times, click sequences, and learning pathways. (3) Feedback Transmission Module: Sends learning outcomes and student feedback back to the backend database, supporting the continuous optimization and adaptive adjustment of the GAI-HMCTM model. The interface design follows a RESTful architecture, with data exchanged with the backend server via HTTPS.

In addition, considering the fairness and ethical compliance of generative instructional content, a bias detection and filtering module was embedded within the GAI-HMCTM system. This module performs automatic screening after text generation, including: (1) Content Filtering Based on Keywords and Semantic Similarity: Statements involving gender, region, academic performance, or social stereotypes are flagged and blocked. (2) Manual Review of High-Risk Outputs: Content flagged as high risk undergoes secondary verification and necessary modifications by a teaching team or ethics review board. This mechanism was applied during model fine-tuning to ensure that instructional content remains neutral, positive, and safe in both semantic and value-oriented aspects. To prevent excessive student reliance on AI, the system incorporates a progressive interaction constraint mechanism. After multiple rounds of dialogue, the AI assistant proactively guides students toward self-reflection or independent problem-solving, and, when necessary, refrains from providing complete answers. This approach encourages learners to maintain active thinking and critical learning.

3.3 Data collection and analysis

1. Data Collection

Learning outcomes: Raw scores from midterm and final exams (objective and subjective questions) were collected. Exam content was designed by third-party experts to ensure validity.

Learning process data: Platform logs from the LMS were automatically collected via backend APIs. The data included total login counts, average session duration, and video completion rates. It also captured access to courseware and supplementary materials, the number and types of AI-assisted questions, and assignment submission timeliness. To evaluate the linguistic quality and instructional suitability of model-generated content, this study employed the natural language processing metrics BLEU and ROUGE to measure semantic consistency and structural coherence. Both the midterm and final examinations were designed by third-party experts. The manual grading was conducted independently by two experts with over ten years of teaching experience. Cohen's κ was calculated to assess inter-rater reliability, yielding $\kappa = 0.86$, which indicates a high level of consistency and ensures the reliability of the manual evaluation.

During data collection and analysis, platform backend logs provided critical support for assessing student learning behavior and model interaction outcomes. Log data were automatically collected via the LMS backend using RESTful APIs. The primary API endpoints included:

- /content/generate: Requests personalized instructional content.
- /user/progress: Uploads student learning progress.
- /feedback/update: Records grading and survey feedback.

Each API endpoint included authentication and access control logic to ensure data security and isolate user permissions. Raw log data were first cleaned, removing entries with formatting errors, missing fields, or duplicates. Time-series alignment was then performed to unify timestamps across modules (accurate to the millisecond) and eliminate asynchronous errors caused by system latency. Finally, anonymization replaced user IDs with random hash values to protect participant privacy and comply with data ethics standards.

In the feature extraction phase, log data were structured into multi-dimensional behavioral indicators, including login frequency, average session duration, interaction count, task completion rate, video study time, and AI Q&A usage. Through time-series aggregation and event window analysis, a dynamic profile of student learning behavior was constructed, which served as a mediating variable in statistical analyses of GAI-HMCTM's instructional interventions. This process is illustrated in Figure 3. Detailed data formats and field definitions are provided in Appendix A.

```
# Pseudocode for LMS log data preprocessing in GAI-HMCTM

# Step 1: Load anonymized LMS logs
logs = load_data("anonymized_logs.json")

# Step 2: Data cleaning - remove incomplete or duplicate records
clean_logs = []
for record in logs:
    if record.is_valid() and not record.is_duplicate():
        clean_logs.append(record)

# Step 3: Time alignment - unify timestamp format (ms precision)
for entry in clean_logs:
    entry.timestamp = standardize_time(entry.timestamp, format="ms")

# Step 4: Feature extraction
features = extract_features(
    clean_logs,
    metrics=[
        "login_frequency",
        "avg_session_duration",
        "AI_query_count",
        "task_completion_rate"
    ]
)

# Step 5: Anonymization (hash user ID)
for f in features:
    f.user_id = hash_user_id(f.user_id)

# Step 6: Output structured dataset for analysis
save_to_csv(features, "processed_features.csv")
```

Figure 3: Pseudocode of Data Processing Procedure

2 Data analysis

Quantitative Analysis: Statistical analyses were conducted using SPSS 26.0. Learning outcomes were compared using independent-samples *t*-tests, and ANCOVA was applied to control for pretest scores. By including pretest scores as a covariate, ANCOVA adjusts for baseline differences between the experimental and control groups, thereby reducing potential confounding effects and allowing observed group differences to more accurately reflect the true impact of the model intervention. Learning process and experience were analyzed using independent-samples *t*-tests to examine differences in engagement, satisfaction, Technology Acceptance Model (TAM) dimensions, and key process indicators. Comparisons also included long-term learning outcomes, cognitive load, and cross-course adaptability. Long-term learning outcomes were assessed via delayed post-tests conducted one week and one month after the experiment, using students' score retention rates and skill transfer ability as primary indicators. Cognitive load was measured combining a self-report scale with average task completion time. After completing assigned tasks, learners self-evaluated their mental effort, task complexity, and focus on a 5-point scale, with lower scores indicating greater efficiency in information presentation and interaction design. Cross-course adaptability was evaluated by applying the model to different course scenarios and comparing its accuracy in task recognition and learning improvement across tasks.

To assess the fairness of the GAI-HMCTM model across different student subgroups, the experimental and control groups were compared across multiple dimensions. Students were categorized by prior course performance into a high-score group (pretest ≥ 80) and a low-score group (pretest < 80). Based on technology proficiency, students were classified into a high-proficiency group (self-assessment ≥ 4/5) and a low-proficiency group (self-assessment < 4/5). Gender groups (male vs. female) were also analyzed. Key learning outcomes (final exam average, assignment average) were compared using independent-samples *t*-tests, and Cohen's d was calculated to evaluate effect sizes.

To further investigate the independent contributions of each functional module within the model, an ablation study was designed: (1) Full model: Includes generative AI content generation, adaptive feedback mechanism, and human–AI collaborative regulation module. (2) Without AI content generation: Retains only AI-assisted feedback and human–AI collaboration. (3) Without adaptive feedback mechanism: Retains generative content and manual instruction but excludes real-time personalization. (4) Without human–AI collaboration: AI independently generates and delivers instructional content without teacher involvement. Student performance and engagement were compared across these configurations to identify the key modules driving model effectiveness and to understand their underlying mechanisms.

4 Results and discussion

1. Comparison of learning outcomes between groups

Learning outcomes for the experimental and control groups are presented in Figure 4 and Table 2.

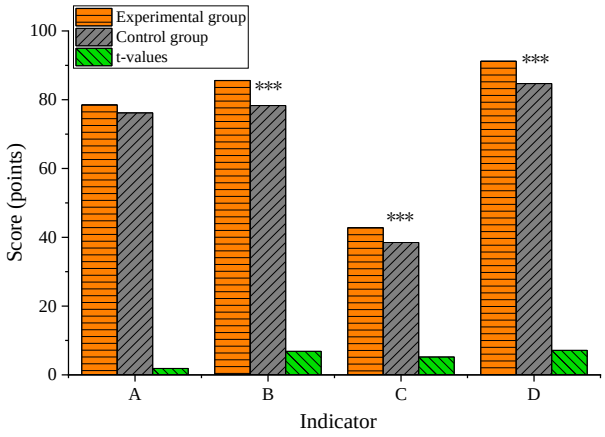


Figure 4: Comparison of learning outcomes between experimental and control groups

(A: Midterm Exam Scores; B: Final Exam Scores; C: Final Exam Subjective Scores; D: Average Assignment Scores)

*Note: $p < 0.05$, *$p < 0.01$, *$p < 0.001$

Table 2: p-values for learning outcomes comparison

Indicator	p-value	Effect Size (Cohen's d)	95% Confidence Interval
Midterm Exam Scores	0.063	0.28 (small)	[-0.02, 0.58]
Final Exam Scores	<0.001	0.82 (large)	[0.50, 1.14]
Final Exam Subjective Scores	<0.001	0.79 (large)	[0.47, 1.11]
Average Assignment Scores	<0.001	0.75 (large)	[0.44, 1.06]

As shown in Figure 4 and Table 2, the experimental group's midterm scores were higher than those of the control group; however, the difference was not statistically significant (p = 0.063), possibly because the initial effects of the new teaching model had not yet fully manifested. By the end of the semester, the differences became highly significant. The experimental group's final exam average score (85.6) was significantly higher than the control group's (78.3), t = 6.85, p < 0.001, indicating a strong positive effect of the GAI-HMCTM model on overall learning outcomes.

The model had a large effect on both final exam and assignment scores, with Cohen's d values approaching or exceeding 0.75, demonstrating substantial practical significance in improving student performance. Although the midterm exam difference was not significant ($p = 0.063$) and the effect size was small ($d = 0.28$), the large effects observed in final exams and assignments suggest that student learning outcomes improved noticeably with increased exposure to the AI system. Specifically, the effect sizes for final exam scores, final exam subjective scores, and average assignment scores were 0.82, 0.79, and 0.75, respectively, with 95% confidence intervals not crossing zero, confirming both statistical and practical significance. The smaller midterm effect size (0.28) and its confidence interval slightly crossing zero indicate a weaker initial intervention effect, likely due to students adapting to the new collaborative model.

Notably, in the subjective portion of the final exam, which better reflects comprehensive application and analytical skills, the experimental group scored 42.8, compared with 38.5 for the control group, a highly significant difference ($p < 0.001$). This suggests that the human–AI collaborative model may particularly support the development of higher-order thinking skills. Furthermore, the experimental group achieved significantly higher average assignment scores (91.2 vs. 84.7, $p < 0.001$), reflecting greater quality and engagement in routine learning tasks.

Beyond formal score comparisons, the linguistic quality of AI-generated instructional content was quantitatively analyzed. The GAI-HMCTM-generated materials demonstrated excellent semantic consistency and content coverage, with BLEU = 0.74 and ROUGE = 0.79, indicating strong and stable alignment with course knowledge points.

In terms of long-term learning outcomes, delayed post-test results showed that the experimental group's score retention rates one week and one month after the experiment were 91.3% and 85.6%, respectively, significantly higher than the control group's 83.5% and 77.4% ($p < 0.01$). This indicates that GAI-HMCTM improves short-term performance and strengthens both knowledge retention and transfer ability. Regarding cognitive load, the experimental group's average score was 2.73 (out of 5), significantly lower than the control group's 3.28 ($p < 0.01$), with an average task completion time reduced by approximately 11.4%. These findings suggest that generative AI optimizations in content presentation and task allocation substantially reduce mental and information-processing burdens, thereby improving focus and learning efficiency.

2. Comparison of learning engagement between groups

Learning engagement during the study period is analyzed for both groups, with results shown in Figure 5.

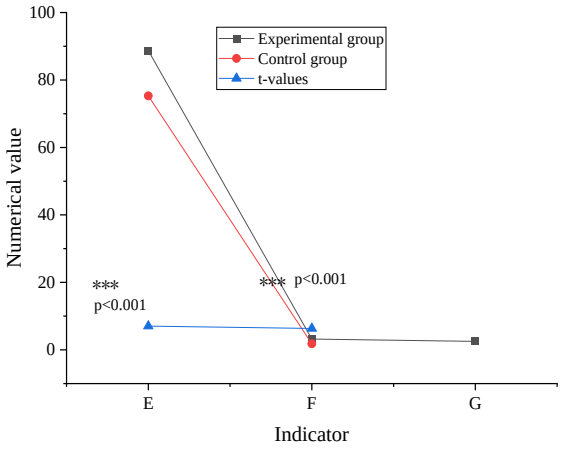


Figure 5: Comparison of Learning Engagement between Experimental and Control Groups (E: Average Completion Rate of Instructional Videos (%); F: Average Access to Supplementary Materials per Week; G: Number of Questions Asked in the AI Tutoring Module per Week) (note: *p<0.05, ** p<0.01, *** p<0.001)

As shown in Figure 5, students in the experimental group demonstrated markedly greater engagement with learning materials compared to the control group. Their average completion rate for instructional videos reached 88.7%, significantly higher than the control group's 75.3% ($p < 0.001$). This difference reflects enhanced persistence in engaging with course content as well as greater concentration and learning discipline promoted by the new teaching model. In addition, the experimental group accessed supplementary learning materials 3.2 times per week on average, nearly twice the control group's 1.8 times per week ($p < 0.001$). This finding suggests that the GAI system's personalized content recommendations and instant Q&A functions effectively stimulated students' curiosity and motivated exploratory learning beyond the core course requirements.

Further evidence of this heightened engagement is shown in the interaction data from the AI tutoring module. On average, experimental group students asked 2.5 questions per week through the AI system, demonstrating a high frequency of learner–AI interaction. Such proactive inquiry reflects a shift from passive reception of knowledge toward active construction of understanding, which is a key feature of the human–AI collaborative teaching model. By lowering the threshold for asking questions and providing immediate, individualized feedback, the AI tutoring module substantially enhanced the accessibility and responsiveness of learning support. In contrast, the control group had no equivalent mechanism, highlighting a structural limitation of traditional instruction. Together, these results highlight that the model enhances academic performance and cultivates an interactive learning ecology, supporting greater cognitive engagement and autonomous learning.

3. Comparison of learning experience and technology acceptance between experimental and control groups

The learning experience and technology acceptance of the experimental and control groups were analyzed, with results shown in Figure 6.

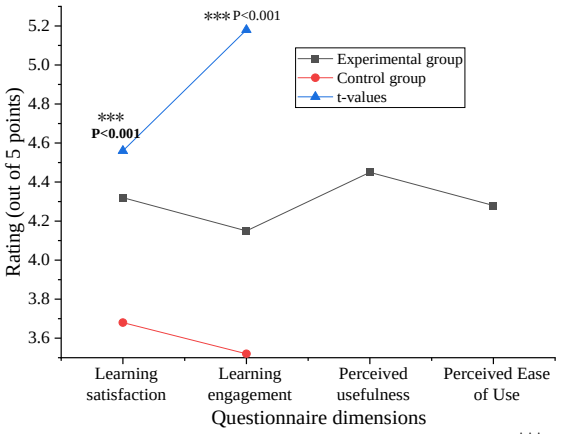


Figure 6: Comparison of Learning Experience and Technology Acceptance between Experimental and Control Groups (note: *p<0.05, ** p<0.01, *** p<0.001)

Figure 6 illustrates the comparative results of learning experience and technology acceptance between the experimental and control groups. The experimental group reported significantly higher overall satisfaction with the teaching model than the control group ($p < 0.001$), reflecting stronger student approval and a positive reception of the GAI-HMCTM approach. This elevated satisfaction level suggests that the integration of generative AI features not only improved the perceived quality of instruction but also created a more engaging and supportive learning environment. Student engagement, measured using the UWES-S scale, was also significantly higher in the experimental group. This finding indicates that the model effectively fostered greater motivation, concentration, and persistence in learning tasks. Higher engagement levels are particularly important because they are closely associated with long-term academic achievement and knowledge retention, suggesting that the human–AI collaborative model may have benefits beyond immediate performance gains. Analysis of the TAM further revealed high acceptance of GAI tools among students in the experimental group. They rated perceived usefulness (PU) at 4.45, suggesting that students believed the AI tools provided substantial learning support, such as personalized recommendations, real-time assistance, and interactive explanations. Perceived ease of use (PEOU) was rated at 4.28, demonstrating that the system was generally intuitive and user-friendly, reducing the cognitive burden associated with adopting new technologies. The combination of high PU and PEOU scores indicates that students readily embraced the new technology, which is a critical condition for the successful implementation and long-term sustainability of technology-enhanced teaching models. By contrast, the control group, which relied solely on traditional instruction without GAI integration, did not generate comparable data on technology acceptance. This absence highlights a structural limitation of conventional approaches, where opportunities for human–technology synergy are not available. Taken together, these results confirm that the GAI-HMCTM model not only enhances measurable academic outcomes but also improves the subjective learning experience and establishes a high level of user trust and acceptance—both essential for scaling such innovations across broader educational contexts.

4. Fairness assessment

The results of fairness evaluation across different student subgroups are presented in Table 3. Regardless of students' baseline performance, the experimental group consistently outperformed the control group, indicating that GAI-HMCTM has a positive impact across varying levels of prior knowledge. Notably, students with low technology proficiency in the experimental group showed slightly greater improvement than high-proficiency students, suggesting that the system's real-time prompts and personalized task allocation help reduce learning gaps caused by differences in technical ability. Both male and female students in the experimental group achieved significantly higher learning outcomes compared with their control group counterparts, with no significant within-group gender differences ($p > 0.05$). This indicates that the model maintains relative fairness across gender. Overall, GAI-HMCTM demonstrated consistent positive effects across all major subgroups, with no evident bias. This suggests that design features such as dynamic task allocation, real-time feedback, and teacher review mechanisms effectively mitigate potential bias and provide equitable learning support for students from diverse backgrounds.

Table 3: Comparison of learning outcomes across student subgroups (experimental vs. control groups)

Subgroup Dimension	Indicator	Experimental Group Mean ± SD	Control Group Mean ± SD	p-value	Cohen's d
High Pretest Score	Final Exam	88.2 ± 5.1	81.4 ± 6.0	<0.001	1.18
High Pretest Score	Average Assignment Score	92.5 ± 4.8	86.9 ± 5.5	<0.001	1.14
Low Pretest Score	Final Exam	82.5 ± 4.9	75.8 ± 5.2	<0.001	1.29
Low Pretest Score	Average Assignment Score	89.1 ± 5.2	82.3 ± 5.7	<0.001	1.32
High Technology Proficiency	Final Exam	87.6 ± 5.0	80.9 ± 6.2	<0.001	1.17

Low Technology Proficiency	Final Exam	83.1 ± 5.3	76.4 ± 5.6	<0.001	1.26
Male	Final Exam	85.4 ± 5.2	78.1 ± 5.8	<0.001	1.35
Female	Final Exam	86.2 ± 5.0	78.5 ± 5.6	<0.001	1.44

5. Correlation between AI use and learning outcomes

The correlation between AI usage behaviors and learning outcomes among students in the experimental group was analyzed, with the results presented in Figure 7.

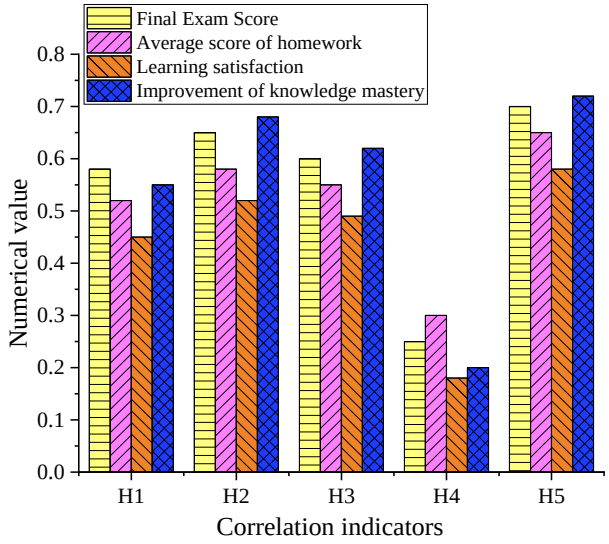


Figure 7: Correlations between experimental group Students' AI Usage behaviors and learning outcomes (H1: Total weekly questions; H2: Weekly concept-focused questions; H3: Weekly problem-solving strategy questions; H4: Weekly direct-answer questions; H5: Average number of dialogue turns per question).

All types of question frequency and interaction depth were positively correlated with learning outcomes, though the magnitude of these relationships varied across indicators. Among the examined variables, the average number of dialogue turns per question—a proxy for discussion depth—exhibited the strongest correlations with all outcome measures (r = 0.70–0.72). This finding suggests that multi-turn, sustained engagement with AI tutors is particularly effective for learning, as it encourages iterative clarification, elaboration, and refinement of understanding rather than one-off responses. Concept-focused questions (r = 0.65–0.68) and problem-solving strategy questions (r = 0.60–0.62) also demonstrated relatively strong correlations with performance outcomes. These question types encouraged students to explore underlying principles and reasoning processes, thereby promoting higher-order thinking and deeper cognitive engagement. Drawing on Vygotsky's Zone of Proximal Development (ZPD) theory, these multi-turn interactions provided moderate challenges. With AI assistance, students were able to complete tasks slightly above their current level, facilitating the development of their potential abilities. Simultaneously, according to Bloom's taxonomy of cognitive domains, these interactions primarily promoted higher-order cognitive activities such as analysis, evaluation, and creation. This aligns with the enhanced higher-order thinking skills demonstrated by experimental group students in their final exam subjective questions and assignments.

Importantly, the strength of these correlations exceeded that of total question frequency (r = 0.55–0.58), indicating that what students ask and how they engage matters more than how often they ask questions. By contrast, direct-answer questions, which sought straightforward factual responses, showed only weak correlations with outcomes (r = 0.20–0.30) and were significantly associated only with assignment scores. This pattern highlights the limited pedagogical value of shallow, answer-seeking interactions, which may provide immediate task-related benefits but do not substantially contribute to broader learning gains. Taken together, these results underscore a central insight: the depth and quality of AI-mediated interactions—characterized by exploratory questioning, critical reflection, and multi-turn dialogue—are more predictive of positive learning outcomes than frequency alone. In this context, AI functions most effectively not as a simple information provider, but as an intelligent collaborator that scaffolds complex reasoning, supports metacognitive processes, and stimulates active knowledge construction.

6. Usage frequency and satisfaction of GAI-HMCTM functional modules

The usage frequency and satisfaction ratings of each functional module were analyzed, with the results shown in Figure 8.

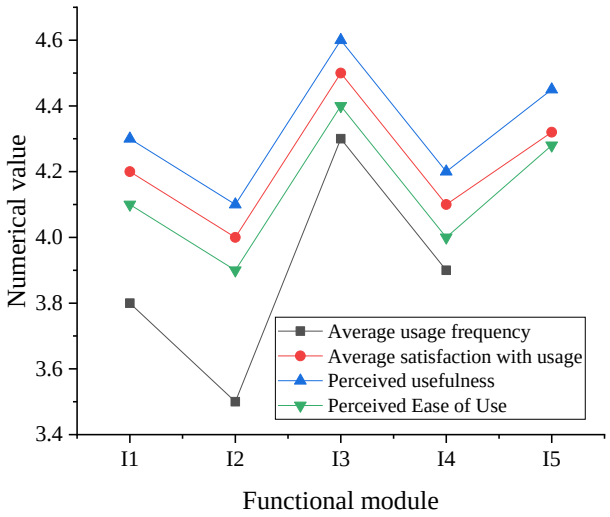


Figure 8: The Usage Frequency and Satisfaction Ratings for Each GAI-HMCTM Module (I1: AI Intelligent Content Generation; I2: AI Personalized Learning Path Recommendation; I3: AI Real-Time Tutoring; I4: AI

Assignment Pre-Evaluation and Feedback; I5: Overall Teaching Model).

All modules had average usage frequencies above 3, indicating widespread adoption and integration into students' learning routines. The AI Real-Time Tutoring module was the most frequently used, consistent with its instant support feature, which meets students' needs for timely problem resolution. Satisfaction ratings for all modules exceeded 4, demonstrating strong student approval. Notably, the Real-Time Tutoring module received the highest satisfaction and PU scores, highlighting its value for addressing immediate learning challenges. AI-generated course materials, exercises, and assignment pre-evaluation/feedback also achieved high satisfaction and usefulness ratings.

7. Comparative analysis of AI-supported teaching models

To further validate the effectiveness and uniqueness of the proposed GAI-HMCTM, this study compared it with two representative AI teaching systems: the Adaptive Learning System (ALS) [23] and the ITS [24]. The comparison dimensions included learning outcome improvement, student engagement, personalized learning effectiveness, long-term learning outcomes, cognitive load, and cross-course adaptability. The results are summarized in Table 4.

Table 4: Performance comparison of different AI-supported teaching models

Evaluation Dimension	ALS	ITS	GAI-HMCTM
Learning Outcome Improvement (%)	8.7	10.3	12.4
Student Engagement Improvement (%)	9.1	12.6	14.8
Personalized Learning Satisfaction (1–5)	4.1	4.3	4.7
Long-term Learning Effect (1–5)	3.9	4.2	4.6
Cognitive Load (1–5)	3.4	3.1	2.7
Cross-Course Adaptability (1–5)	3.8	4.0	4.5

As shown in Table 4, GAI-HMCTM demonstrated significant advantages across multiple key dimensions. In terms of learning outcomes and engagement, it outperformed ALS by approximately 3.7 and 5.7 percentage points, respectively. It also excelled in long-term learning effects and cross-course adaptability, indicating that generative AI provides strong support for sustained learning and knowledge transfer. Furthermore, students in the GAI-HMCTM environment experienced the lowest cognitive load, suggesting that dynamically generated learning content and the real-time collaboration mechanism effectively reduce learning pressure while enhancing focus and fluency in the learning experience. Overall, GAI-HMCTM outperforms traditional AI teaching models in immediate learning outcomes while enhancing long-term learning sustainability, adaptability, and cognitive efficiency, presenting a new pathway for intelligent education ecosystems.

8. Ablation study results

To further examine the independent contributions of each functional module to overall learning outcomes, an ablation study was conducted. The results are presented in Table 5. The full GAI-HMCTM model consistently performed best across all four indicators, demonstrating significant synergistic effects among modules. Removing the AI content generation module led to a 5.2-point decrease in student scores and a ~5.5% reduction in AI response accuracy, highlighting the critical role of generative AI in knowledge coverage and diverse example generation, which enhances both learning mastery and interest. When the adaptive feedback mechanism was removed, scores further declined to 81.9, and satisfaction decreased significantly, indicating that personalized feedback is a key driver for maintaining motivation and sustained engagement. The experiment without the human–AI collaboration module showed the poorest performance: scores dropped nearly 9 points, and cognitive load increased to 3.42, underscoring the irreplaceable role of teachers in guiding learning pathways, regulating emotions, and providing value-oriented direction. Overall, the complete GAI-HMCTM model exceeded the performance of any single-module setup on all metrics, indicating that its strengths derive from the integration of generative AI and the dynamic, coordinated interaction between teachers and AI.

Table 5: Ablation Study Results Comparison (Mean ± SD)

Model Configuration	Final Score	Satisfaction	Cognitive Load	AI Response Accuracy (%)
Full Model	88.6 ± 4.3	4.62 ± 0.18	2.73 ± 0.21	94.7 ± 1.2
Without Content Generation Module	83.4 ± 4.9	4.18 ± 0.24	3.05 ± 0.27	89.2 ± 1.9
Without Emotional Recognition Module	81.9 ± 5.2	4.02 ± 0.31	3.14 ± 0.28	86.8 ± 2.1
Without Behavior Adaptation Module	79.6 ± 6.1	3.74 ± 0.37	3.42 ± 0.34	82.5 ± 2.8

9. Analogy analysis with engineering control methods

To further elucidate the dynamic regulation and optimization mechanisms of GAI-HMCTM, this study drew on typical adaptive and robust control theories from

the field of engineering to perform an analogy analysis of the model's behavioral characteristics. Engineering control systems often face environmental uncertainty, nonlinear complexity, and multi-objective optimization challenges, which parallel the highly complex and dynamically changing nature of educational systems. Through this analogy, the "self-regulation—feedback—optimization" loop of GAI in instructional processes can be better understood. The results of this analogy are summarized in Table 6, showing that GAI-HMCTM demonstrates dynamic robustness and adaptive optimization capabilities similar to those of engineering control systems when managing uncertainty and complexity in educational contexts.

Table 6: Correspondence between engineering control methods and GAI-HMCTM

Engineering Control Method	Core Feature	Corresponding GAI-HMCTM Mechanism	Advantage
Adaptive Fuzzy Control [25]	Real-time adjustment of uncertain systems	Dynamic adjustment of instructional content	Enhances personalization and learning stability
Output Feedback Control [26]	Stability based on system output	Feedback-driven instructional optimization	Ensures stability across multiple teaching loops
Neural Adaptive Control [27]	Dynamic compensation for complex systems	Real-time monitoring of student behavior and strategy adjustment	Supports dynamic knowledge generation and personalized feedback
Nonlinear Optimal Control [28]	Optimal performance under constraints	Optimization of teacher–student–AI collaboration	Balances efficiency with learning experience

Building on this analogy, the operational feasibility of GAI-HMCTM was evaluated in a university foundational course experiment. The system automatically generated personalized learning tasks based on students' real-time assignment performance and provided instant prompts and case demonstrations through the generative AI interaction module. During implementation, teachers monitored class performance via a backend dashboard and made necessary adjustments to AI recommendations, forming a teacher-led, AI-assisted collaborative model. The experimental results indicated that, compared with traditional ITS, student engagement increased by approximately 14%, personalized learning satisfaction improved by about 0.4 points (on a 5-point scale), and long-term learning effectiveness scores rose by around 0.4 points, while cross-course adaptability remained strong. This empirical application demonstrated the practical feasibility of GAI-HMCTM in real classrooms. It also highlighted the model's advantages in task allocation, interactive support, and learning data management, offering practical guidance for further optimization of AI-based instructional models.

10. Discussion

This study validated the operational feasibility and effectiveness of the GAI-HMCTM in real educational settings. Experimental results demonstrated that the model outperformed traditional teaching approaches in learning outcomes, assignment performance, personalized learning satisfaction, and long-term learning effects. Compared with existing research, GAI-HMCTM showed distinct advantages in personalized learning and interactive support. For example, Li et al. [29] reported that conventional ITS mainly relied on fixed task sequences, limiting personalization. Ooi et al. [30] emphasized generative AI for content creation but lacked real-time classroom interaction and teacher collaboration mechanisms. Strielkowski et al. [31] proposed ALS capable of adjusting exercise difficulty, but they exhibited limited cross-course adaptability. GAI-HMCTM combines real-time performance-based task generation, in-class prompts and case demonstrations, and allows teachers to monitor and adjust AI operations, thereby enhancing subjective evaluation scores and student engagement. Further analysis revealed a significant correlation between depth of classroom interaction and learning outcomes. In line with cognitive engagement theory, this suggests that deep interactions facilitate information processing and knowledge internalization.

In terms of scalability, GAI-HMCTM employs a modular design and cloud computing support, enabling deployment in classrooms of different sizes and offering lightweight solutions for schools with limited hardware resources. Teachers maintain a leading role in instruction, with all AI-generated learning paths, prompts, and assignment grading suggestions subject to teacher review, ensuring alignment with educational objectives. Although this study applied GAI-HMCTM to a computer science course to ensure content expertise and control, the model's core components demonstrate strong domain-independence. For example, the AI real-time tutoring module and personalized learning path recommendation module dynamically analyze learner behavior and knowledge structures, enabling adaptation across various disciplines and course contexts. Teachers can adjust AI-generated prompts and task assignment strategies according to course knowledge points and learning objectives, supporting cross-disciplinary application.

Regarding ethics and safety, the system implements several safeguards: (1) diverse training data to reduce bias risk; (2) model usage logs and periodic audits to detect and correct potential biases; (3) strict management of student data access, employing encryption for storage and transmission, and limiting AI processing of sensitive information to protect privacy. Overall, GAI-HMCTM offers clear advantages in personalized learning, real-time interactive support, and teacher collaboration, while demonstrating potential for cross-course application.

Nevertheless, large-scale deployment faces challenges including teacher training, hardware resources, model optimization, and ethical compliance. Future research could explore system lightweighting, user interface optimization, and cross-disciplinary adaptability to enable broader and more sustainable application. Further work will also examine the transferability and adaptation of the model across other subjects and course types.

5 Conclusion

This study developed and empirically validated a GAI-HMCTM. The results show that integrating intelligent content generation, personalized learning pathways, real-time AI tutoring, and teacher–AI collaboration mechanisms can effectively enhance students' learning outcomes and overall experience. Experimental data indicated that students in the experimental group significantly outperformed the control group, which received traditional instruction, in final exam scores, assignment quality, learning engagement, and satisfaction. The model also demonstrated distinct advantages in promoting higher-order thinking skills and encouraging proactive learning. Qualitative interviews revealed that the model reduced teachers' burden from repetitive tasks, allowing them to focus more on instructional design and deeper interaction. Students highly valued AI as an "intelligent collaborator," providing immediate and personalized support. They offer both practical guidance and a theoretical framework for advancing digital transformation in education.

However, the study has limitations. The experiment lasted only one semester, so long-term effects and potential negative impacts—such as on creativity or academic integrity—require further investigation. The participants were limited to computer science students at a single university, leaving the model's generalizability to other disciplines and educational levels untested. Moreover, strategies for monitoring the accuracy of GAI-generated content and fostering students' critical thinking need further refinement. Future research will aim to extend the experimental period, expand application contexts, optimize GAI models for educational adaptability and ethical use, and explore multimodal human–AI collaboration and deeper cognitive support. Furthermore, the long-term implementation of human–AI collaborative teaching models may face several challenges. These include data privacy and security risks. Another concern is the potential impact of AI-driven decisions on educational equity and the balance of teacher–student roles. Additionally, reliance on AI assistance can lead to cognitive biases and raise questions about responsibility allocation. Therefore, future research should not only continue to validate the model's effectiveness but also focus on establishing comprehensive ethical guidelines and regulatory mechanisms. This will ensure that human–AI collaborative teaching enhances instructional efficiency and personalization while safeguarding educational fairness and sustainable development. Generative AI models also demand high hardware and data-processing capabilities, and teachers must be proficient with system interactions and feedback mechanisms, resulting in relatively high initial training costs. Future efforts could focus on model lightweighting, algorithmic efficiency improvements, and user interface optimization to reduce hardware and operational barriers. Additionally, customized training programs aligned with teacher usage habits could further enhance teaching adaptability and system adoption rates, supporting broader and more sustainable integration of human–AI collaborative educational models.

Acknowledge

This article is the research outcome of the 2025 Jiangxi Provincial Vocational Education Teaching Reform Project titled "Research on the Reform and Practice of Human-Machine Collaborative Teaching Models in Higher Vocational Institutions from the Perspective of Artificial Intelligence" (No. JXJG-24-74-5), authored by Wang Zhijuan.

References

[1] Strika Z, Petkovic K, Likic R, et al. Bridging healthcare gaps: a scoping review on the role of artificial intelligence, deep learning, and large language models in alleviating problems in medical deserts[J]. Postgraduate medical journal, 2025, 101(1191): 4-16. https://doi.org/10.1093/postmj/qgae122

[2] Raman R, Sreenivasan A, Suresh M, et al. AI-driven education: a comparative study on ChatGPT and Bard in supply chain management contexts[J]. Cogent Business & Management, 2024, 11(1): 2412742.https://doi.org/10.1080/23311975.2024.2412742

[3] Jackson I, Ivanov D, Dolgui A, et al. Generative artificial intelligence in supply chain and operations management: a capability-based framework for analysis and implementation[J]. International Journal of Production Research, 2024, 62(17): 6120-6145. https://doi.org/10.1080/00207543.2024.2309309

[4] Yaling Z, Rehman A U, Ullah I, et al. Exploring tourism experiences: comparative trends and hotspots in Chinese and American research[J]. Cogent Business & Management, 2025, 12(1): 2482027. https://doi.org/10.1080/23311975.2025.2482027

[5] Lei P, Duo E N, Gao Y, et al. Research on the experience of utilizing artificial intelligence in nursing learning from the perspective of nursing students–a qualitative study[J]. BMC nursing, 2025, 24(1): 1062. https://doi.org/10.1186/s12912-025-03704-7

[6] Xu M, Du H, Niyato D, et al. Unleashing the power of edge-cloud generative AI in mobile networks: A survey of AIGC services[J]. IEEE Communications Surveys & Tutorials, 2024, 26(2): 1127-1170. https://doi.org/10.1109/COMST.2024.3353265

[7] Huang J, Zhong Y, Chen X. Adaptive and personalized learning in STEM education using high-

performance computing and artificial intelligence[J]. The Journal of Supercomputing. 2025, 81(8):981. https://doi.org/10.1007/s11227-025-07481-7

[8] Yan L, Greiff S, Teuber Z, et al. Promises and challenges of generative artificial intelligence for human learning[J]. Nature Human Behaviour, 2024, 8(10): 1839-1850. https://doi.org/10.1038/s41562-024-02004-5

[9] Bauer E, Greiff S, Graesser A C, et al. Looking beyond the hype: Understanding the effects of AI on learning[J]. Educational Psychology Review, 2025, 37(2): 45. https://doi.org/10.1007/s10648-025-10020-8

[10] Batista J, Mesquita A, Carnaz G. Generative AI and higher education: Trends, challenges, and future directions from a systematic literature review[J]. Information, 2024, 15(11): 676. https://doi.org/10.3390/info15110676

[11] Deguara D. Embracing the academic change caused by the proliferation of generative artificial intelligence chatbots in higher education[J]. MCAST Journal of Applied Research & Practice, 2024, 8(1): 150-178. https://doi.org/10.5604/01.3001.0054.5100

[12] Sleem A. Generative Artificial Intelligence (GAI) as a Teaching and Learning Method to Support Creativity in Product Design[J]. Journal of Design Sciences and Applied Arts, 2025, 6(2): 1-2. https://doi.org/10.21608/jdsaa.2025.376415.1469

[13] Cheong M. ChatGPT's Performance Evaluation in Spreadsheets Modelling to Inform Assessments Redesign[J]. Journal of Computer Assisted Learning, 2025, 41(3): e70035. https://doi.org/10.1111/jcal.70035

[14] Aperstein Y, Cohen Y, Apartsin A. Generative ai-based platform for deliberate teaching practice: A review and a suggested framework[J]. Education Sciences, 2025, 15(4): 405. https://doi.org/10.3390/educsci15040405

[15] Vafadar M, Amani A M. Academic education in the era of generative artificial intelligence[J]. Journal of Electronics and Electrical Engineering, 2024: 117–133-117–133. https://doi.org/10.37256/jeee.3120244010

[16] Arnau-González P, Solera-Monforte S, Wu Y, et al. A framework for adapting conversational intelligent tutoring systems to enable collaborative learning[J]. Expert Systems with Applications, 2025, 271: 126663. https://doi.org/10.1016/j.eswa.2025.126663

[17] Ding L, Zou D. Automated writing evaluation systems: A systematic review of Grammarly, Pigai, and Criterion with a perspective on future directions in the age of generative artificial intelligence[J]. Education and Information Technologies, 2024, 29(11): 14151-14203. https://doi.org/10.1007/s10639-023-12402-3

[18] Chiu T K F, Rospigliosi P. Encouraging human-AI collaboration in interactive learning environments[J]. Interactive Learning Environments, 2025, 33(2): 921-924. https://doi.org/10.1080/10494820.2025.2494914

[19] Tülübaş T, Demirkol M, Ozdemir T Y, et al. An interview with ChatGPT on emergency remote teaching: A comparative analysis based on human–AI collaboration[J]. Educational Process: International Journal, 2023, 12(2): 93. https://doi.org/10.22521/edupij.2023.122.6

[20] Islam M T, Sepanloo K, Woo S, et al. A review of the industry 4.0 to 5.0 transition: exploring the intersection, challenges, and opportunities of technology and human–machine collaboration[J]. Machines, 2025, 13(4): 267. https://doi.org/10.3390/machines13040267

[21] Cooper G. Examining science education in ChatGPT: An exploratory study of generative artificial intelligence[J]. Journal of science education and technology, 2023, 32(3): 444-452. https://doi.org/10.1007/s10956-023-10039-y

[22] Zhang H, Jin X, Cui X. Exploration of Course Resources and Modes under Generative Artificial Intelligence[J]. Frontiers in Educational Innovation and Research. 2025, 1(1):4-9. https://doi.org/10.62762/FEIR.2024.649562

[23] Sarnato A Z, Sari W D, Rahmawati S T, et al. The evolution of e-learning platforms: From u-learning to ai-driven adaptive learning systems[J]. Journal of Social Science Utilizing Technology, 2024, 2(2): 289-300. https://doi.org/10.70177/jssut.v2i2.1131

[24] Zhong X, Zhan Z. An intelligent tutoring system for programming education based on informative tutoring feedback: system development, algorithm design, and empirical study[J]. Interactive Technology and Smart Education, 2025, 22(1): 3-24. https://doi.org/10.1108/ITSE-09-2023-0182

[25] Boulkroune A, Zouari F, Boubellouta A. Adaptive fuzzy control for practical fixed-time synchronization of fractional-order chaotic systems[J]. Journal of Vibration and Control, 2025: 10775463251320258. https://doi.org/10.1177/10775463251320258

[26] Boulkroune A, Hamel S, Zouari F, et al. Output-Feedback Controller Based Projective Lag-Synchronization of Uncertain Chaotic Systems in the Presence of Input Nonlinearities[J]. Mathematical Problems in Engineering, 2017, 2017(1): 8045803. https://doi.org/10.1155/2017/8045803

[27] Wu J, Sun W, Su SF, Xia J. Neural-based adaptive control for nonlinear systems with quantized input and the output constraint[J]. Applied Mathematics and Computation. 2022, 413:126637. https://doi.org/10.1016/j.amc.2021.126637

[28] Rigatos G, Abbaszadeh M, Sari B, et al. Nonlinear optimal control for a gas compressor driven by an induction motor[J]. Results in Control and Optimization, 2023, 11: 100226. https://doi.org/10.1016/j.rico.2023.100226

[29] Li X, Lu X, Chen W, et al. Research on UAVs reconnaissance task allocation method based on communication preservation[J]. IEEE Transactions on Consumer Electronics, 2024, 70(1): 684-695. https://doi.org/10.1109/TCE.2024.3368062

[30] Ooi K B, Tan G W H, Al-Emran M, et al. The potential of generative artificial intelligence across

disciplines: Perspectives and future directions[J]. Journal of Computer Information Systems, 2025, 65(1): 76-107. https://doi.org/10.1080/08874417.2023.2261010

[31] Strielkowski W, Grebennikova V, Lisovskiy A, et al. AI-driven adaptive learning for sustainable educational transformation[J]. Sustainable Development, 2025, 33(2): 1921-1947. https://doi.org/10.1002/sd.3221

Appendix A. Data format description and field definitions

Field Name	Data Type	Description	Example Value	Processing Method
user_id	string	Unique learner identifier (anonymized)	"A35F2K1D"	Hashed to protect privacy
timestamp	datetime	Operation timestamp (ISO 8601 format)	"2024-05-16T09:23:45.128Z"	Standardized to UTC format
event_type	string	Type of interaction event (e.g., login, video view, AI query)	"AI_query"	Grouped by type for statistics
session_id	string	Learning session identifier	"S19283"	Used to calculate average session duration
duration	float	Duration of a single interaction (seconds)	124	Outliers removed (>3σ)
course_id	string	Course code	"CS102"	Used for cross-course analysis
content_id	string	Learning content or resource identifier	"Lec03_Video"	Linked to course content table
completion_rate	float	Content completion rate (0–1)	0.93	Missing values smoothed
interaction_count	int	Number of interactions in the current session	5	Normalized to per-hour unit
feedback_score	float	User feedback score on content or AI assistance (1–5)	4.7	Used to compute average satisfaction

Fine-Tuning OpenAI Whisper-Small for Domain-Specific Medical Speech Recognition within a Microservice Architecture

Alaeddine Moussa[1], Noursene Drine[2]
[1]Université de la Manouba, École Nationale des Sciences de l'Informatique, Tunisia
[2]Universite de Tunis El Manar, École Nationale des Sciences de l'Informatique, Tunisia
E-mail: dr.alaeddine.moussa@gmail.com, noursene.drine@corilus.be

Keywords: Speech recognition, Whisper, fine-tuning, healthcare, microservices

Received:

We fine-tune Whisper-small (244M parameters) on 8.5 hours of in-domain medical audio and evaluate with word error rate (WER). Compared to an unadapted Whisper-small baseline, our fine-tuned model reduces WER from ∼63% to ∼32%. While the relative gain is substantial, this accuracy is not suitable for unsupervised clinical use; we position the system as a clinician-in-the-loop assistant. We also describe deployment as an on-premise microservice and report latency/throughput considerations.

Povzetek: Študija pokaže, da fino-uglaševanje Whisper-small na omejenem medicinskem zvoku občutno zniža WER, vendar ostane primerno predvsem za virtualnega pomočnika z lokalno mikroservisno uvedbo.

1 Introduction

Healthcare professionals still face difficulties in organizing massive patient data, managing administrative tasks, and making sure that bills are handled accurately. Inefficiencies, lost time and errors can result from manual entries and the wrong management of data and patient records which can eventually affect communication and standard care. Furthermore, it can be challenging and time-consuming to assess financial performance and spot trends without the right tools for data visualization[1]. However, deploying speech recognition in clinical practice is challenging due to technical accuracy requirements and domain-specific vocabulary. Previous studies of automatic speech recognition (ASR) in clinical contexts have reported widely varying accuracy (WER ranging from 18% up to 63% in different systems and environments[1]). This variance stems from differences in audio quality, medical jargon, and speaker accents, underlying the need for tailored STT solutions in healthcare. Recent advances in deep learning have led to powerful general-purpose speech models; for example, OpenAI Whisper was trained on 680,000 hours of multilingual audio (65% English, the rest spanning 98 languages)[2]. Whisper's transformer-based architecture has demonstrated robust performance across languages, accents, and noisy inputs [3]. These characteristics make it a promising candidate for medical transcription, where audio conditions and dialects can vary.

This study presents a software platform for medical practice management, which includes multiple microservices (user management, patient records, data visualization, etc.). Within this platform, we developed a dedicated Speech-to-Text microservice to handle transcription of recorded consultations. Figure 1 illustrates the overall architecture. The STT microservice operates alongside others behind a common API Gateway, enabling independent development and scaling of the transcription component.

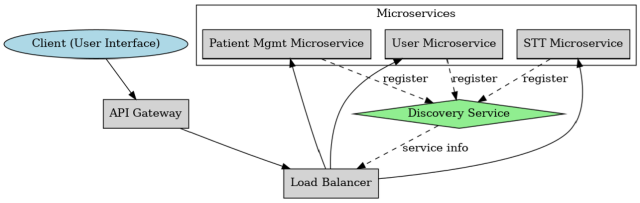


Figure 1: Global architecture of the medical software system, highlighting the Speech-to-Text (STT) microservice and its interactions. The client application sends audio recordings via an API Gateway, which routes requests through a Load Balancer to the STT microservice. The STT service registers with a Discovery Service for service lookup, and works in parallel with other microservices (User management, Patient management) within the platform.

In this paper, we focus on the design and implementation of the STT microservice using the Whisper model for medical speech recognition. In the next section, we review related STT solutions in healthcare, including commercial systems and recent research, to position our approach. We then detail our system methodology, wrapping the microservice design, model integration, and fine-tuning procedure. The dataset used for training (a public medical speech corpus from Kaggle) is described, along with the preprocessing steps undertaken to improve audio quality. We present experimental results evaluating the fine-tuned model's performance (using WER as the metric) and discuss how it compares to baseline models and human-level

performance. Finally, we address challenges encountered, such as limited training data and domain-specific errors, and suggest future improvements (e.g., larger models, more data, and error correction mechanisms) before concluding the study.

Research questions: We structure our study around four questions:

1. **RQ1:** What is the baseline WER of an unadapted Whisper-small model on our in-domain medical audio?
2. **RQ2:** To what extent does full fine-tuning reduce WER on this dataset?
3. **RQ3:** Under which conditions is the system suitable for clinician-in-the-loop use, and where does it fail?
4. **RQ4:** How can a microservice deployment meet practical constraints (latency, throughput, scaling)?

Contributions:

- A fine-tuned Whisper-based clinical ASR microservice with a WER reduction from 63% to 32%.
- A clear positioning of the system as *human-in-the-loop* for safe clinical workflows.
- A practical discussion of deployment trade-offs (interface, scaling) and a roadmap to robustness and generalization.
- Reproducible implementation details to ease adoption in healthcare settings.

2 Literature review

Speech recognition technology for healthcare applications has advanced significantly, evolving from early dictation systems to modern end-to-end deep learning models. Historically, products like Nuance Dragon Medical [4] have dominated clinical dictation by offering specialized vocabularies for physicians and demonstrating high accuracy. A recent pilot study found that a fine-tuned Whisper model performed comparably to Dragon for dictating neurosurgery reports. The Whisper model achieved a mean Word Error Rate (WER) of 1.75%, while Dragon recorded a WER of 1.54%, indicating that Whisper's performance is statistically non-inferior. Furthermore, Whisper outperformed Dragon in some linguistic aspects, with a WER of 0.50% compared to Dragon's 1.34% when excluding formatting differences. This suggests that, with appropriate tuning, open models can achieve accuracy levels similar to established proprietary systems in controlled environments. Additionally, cloud-based speech-to-text (STT) services from companies like Google, Amazon, and Microsoft have also been utilized for medical transcription. For instance, Google's Speech-to-Text API offers a dedicated healthcare model, with studies reporting WERs of approximately 8–12% for typical clinical recordings under ideal conditions. However, accuracy may decline when faced with accents or noisy environments. Concerns about data privacy and cost are also driving a preference for on-premise or open-source alternatives.

Parallels with adaptive control: Beyond speech recognition, adaptation to uncertainty is a central theme in complex dynamical systems. In adaptive and robust control, models are updated (or controllers are synthesized) to maintain performance under variability and noise (e.g., adaptive fuzzy schemes, output-feedback and neural adaptive control). *Our fine-tuning of Whisper plays an analogous role*: the base ASR is specialized to the medical domain so that decision boundaries reflect domain-specific phonetics and terminology. While these control-theoretic methods and ASR are not directly comparable in metrics or problem setting, they share the core idea that *targeted adaptation* to operating conditions yields large performance gains. See, for instance, representative works on adaptive fuzzy control for fixed-time synchronization, projective lag-synchronization with uncertainties, robust neural adaptive control for uncertain nonlinear multivariable systems, adaptive backstepping for SISO nonlinear plants and flexible manipulators, and nonlinear optimal control in industrial settings [5, 6, 7, 8, 9, 10].

Open-source and research models

Academic and open-source automatic speech recognition (ASR) frameworks such as Kaldi[11], Mozilla DeepSpeech[12], and Wav2Vec 2.0[13] have been utilized to develop medical transcription models. These models often require a significant amount of domain-specific training data. The release of OpenAI's Whisper in late 2022 marked a significant advancement in this area by introducing a pre-trained, general-purpose model equipped with strong zero-shot transcription capabilities. Whisper was trained in a fully supervised manner on a vast and diverse dataset, which gives it robustness against a variety of speech patterns and background noises [14, 15] . Whisper utilizes a Transformer encoder-decoder architecture Figure2, which allows it to not only transcribe speech but also perform speech translation and language identification using special tokens. Notably, Whisper processes audio in 30-second segments, converting input waveforms into log-Mel spectrogram features that the encoder processes, while the decoder generates text tokens autoregressively. This single-model, multitask design simplifies the speech processing pipeline [16].

Whisper's accuracy for medical dialogue can be moderate without fine-tuning. For example, the Word Error Rate (WER) of Whisper's small model is about 63% when evaluated on a general English speech corpus like Common Voice, which is too high for reliable clinical transcription.

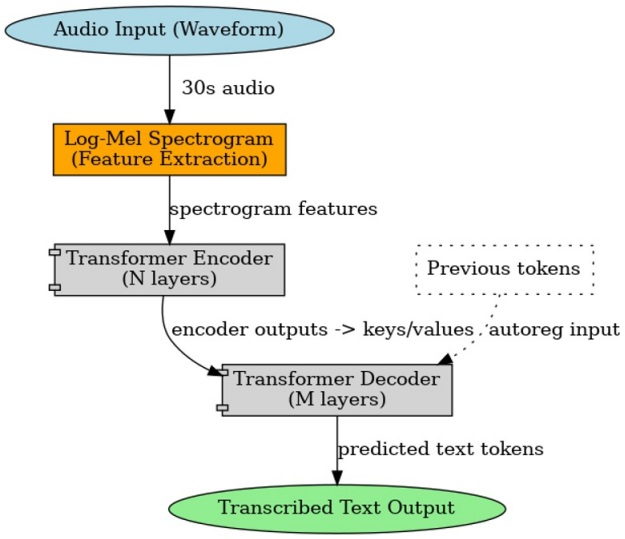


Figure 2: Whisper model architecture (Transformer-based encoder-decoder). The 30 s audio input is first converted into a log-Mel spectrogram, which the Transformer encoder ingests. The Transformer decoder then generates the transcribed text, autoregressively predicting tokens while attending to the encoder outputs. Whisper uses special tokens to handle tasks like language detection and translation as well. This architecture enables a single model to perform robust speech recognition across many languages and conditions.

To improve the recognition of medical terminology and diverse speaker styles, fine-tuning the model on in-domain data is essential. Recent research has focused on adapting these models to specific domains. One approach involves generating synthetic medical speech data for training. For instance, [15] used real-world home healthcare recordings to evaluate various Automatic Speech Recognition (ASR) systems, highlighting performance disparities among different patient groups. In another study, large-scale data augmentation created a synthetic medical speech dataset of approximately 5,486 hours. Models fine-tuned on this dataset, such as a Conformer RNN-T and a Whisper-derived model, achieved WERs as low as 3.9% to 4.6% on medical test sets. These outcomes illustrate that with sufficient domain-specific training, state-of-the-art models can reach error rates of just a few percent.

Moreover, specialized ASR providers have developed models tailored for healthcare. For example, Deepgram's Nova model, which focuses on medical training, has been reported to slightly outperform Whisper on healthcare transcription benchmarks [17]. Despite this, Whisper's openness and adaptability make it an appealing option for institutions that prefer an on-site solution without sharing data with third-party APIs[18].

To ground our work in the literature, Table 1 contrasts Dragon, Google STT, Whisper variants, and Wav2Vec 2.0 across dataset scale, WER, domain, and deployment setting.

In summary, the literature suggests that: (1) achieving high accuracy in medical speech recognition is possible, but typically requires either proprietary solutions or extensive domain training; (2) Whisper serves as a robust foundation that can be fine-tuned for medical applications, providing a balance between open-source flexibility and commercial-level accuracy; and (3) a microservice architecture can facilitate the integration of such a speech-to-text engine into healthcare software, allowing for modular deployment and scaling. Building on these insights, our work fine-tunes the Whisper model using a moderately sized public medical speech dataset and incorporates it as a microservice in a telehealth application. The following sections will detail our methodology and how it addresses the identified challenges.

3 Methodology

3.1 System architecture and integration

The speech-to-text (STT) functionality is implemented as an independent microservice within our healthcare application. We chose a microservice architecture for its modularity and scalability; each service encapsulates specific functionality and can be developed, deployed, and scaled independently. In this case, the STT microservice coexists with user management and patient record microservices, among others.

All microservices register with a central Discovery Service, allowing them to be located by name. A reverse proxy API Gateway routes external requests to the appropriate service instance. A load-balancing mechanism ensures that if multiple instances of the STT service are running (to process many audio streams simultaneously), requests are evenly distributed. Authentication is managed upstream by the gateway, allowing the STT service to focus on its core task.

Within the STT microservice, we have implemented a RESTful API to handle transcription requests. Clients, such as the web or mobile front-end used by doctors, submit audio recordings (for example, of patient consultations) via an HTTP request to the API Gateway, which forwards the requests to the STT service's endpoint. The STT service processes the audio and returns the text transcription. Additionally, the service provides endpoints for CRUD (Create, Read, Update, Delete) operations on the transcript data, enabling users to retrieve, update, or delete saved transcriptions.

This design fits a microservice architecture composed of independently deployable services. For each feature: a User service for authentication, a Patient service for patient data, and an STT service for transcription. Decoupling the STT component ensures that intensive audio processing tasks do not interfere with the responsiveness of other parts of the application. It also allows the STT model to be hosted on hardware suited for the task (such as a GPU), separate from the main application server.

Table 1: Comparative summary (figures are not strictly comparable across rows; datasets and conditions differ).

Model	Dataset size / type	Reported WER	Domain	Deployment
Dragon Medical One (Nuance)	Dictation (vendor cases)	n/a (vendor "∼99% accuracy", not WER)	Medical dictation	Cloud, proprietary
Google STT — Medical Conversation	36 primary-care conversations (re-enacted studio)	∼9.1% (medical conv.); 8.8% (general)	Medical conversation	Cloud API
Whisper-large (zero-shot)	LibriSpeech test-clean (general audiobooks)	2.5% (zero-shot, clean)	General	On-prem/open-source
Wav2Vec 2.0 Large (supervised)	LibriSpeech 960 h (general audiobooks)	1.8% / 3.3% (clean/other)	General	On-prem/open-source
Whisper-small (baseline, ours)	8.5 h in-domain (medical)	63%	Medical (ours)	On-prem/open-source
Whisper-small (fine-tuned, ours)	8.5 h in-domain (medical)	32%	Medical (ours)	On-prem/open-source

Communication between microservices utilizes HTTP requests facilitated by the gateway, and all services share a common identity/authentication scheme using JWT tokens. The STT service was developed in Python using the Flask framework for the API, as the Whisper model and its training code are also in Python, leveraging PyTorch and the Hugging Face Transformers libraries for model management. Other services were implemented in C# (.NET Core) according to the project's technology choices. However, this heterogeneity is well-supported by the microservice architecture. As long as each service exposes REST endpoints and registers with the discovery service, the specific implementation language is irrelevant to the client. This flexibility exemplifies one of the key benefits of microservices.

3.2 Whisper model integration

At the core of the STT microservice is the OpenAI Whisper model. We have integrated the small version of Whisper, which strikes a good balance between accuracy and efficiency for our use case. This small model, composed of 244M parameters, requires approximately 2 GB of VRAM for inference, fitting well within our available GPU memory. We selected this model size due to our limited computational resources and the relatively small dataset we had for fine-tuning.

While larger Whisper models, with up to 1.5B parameters, may provide higher accuracy, they also demand significantly more memory and computational power. This makes them more challenging to deploy in our real-time service without specialized hardware.

Table 2 outlines the number of parameters in each Whisper model along with their computational requirements.

Table 2: Whisper models and their approximate memory requirements

Size	Parameters	English-only	Multilingual	VRAM	Speed
tiny	39M	`tiny.en`	`tiny`	∼1 GB	∼32x
base	74M	`base.en`	`base`	∼1 GB	∼16x
small	244M	`small.en`	`small`	∼2 GB	∼6x
medium	769M	`medium.en`	`medium`	∼5 GB	∼2x
large	1550M	N/A	`large`	∼10 GB	1x

We began with the pre-trained multilingual Whisper-small checkpoint as our foundation. Whisper's architecture is based on a Transformer encoder-decoder model. The encoder transforms audio features into latent representations, while the decoder generates text. This model was initially trained on a vast corpus of voice data with transcripts that included various languages and tasks. This pre-training endows the model with strong general speech recognition capabilities, although it is not specifically optimized for medical terminology. Our goal was to fine-tune Whisper on a medical speech dataset to enhance its performance in clinical contexts.

Fine-tuning was conducted using the Hugging Face Transformers library, which offers an accessible interface to load the Whisper model and further train it on new data. We framed the task as supervised speech recognition, where each training example consists of an audio file paired with its corresponding transcript text. The fine-tuning process was fully supervised since our dataset includes ground-truth transcripts. To mitigate the risk of overfitting—given the small size of our dataset—we configured the training with relatively conservative hyperparameters. Specifically, we used a low learning rate (approximately 1×10^{-5}) and implemented early stopping based on a validation set. This fine-tuning effectively adjusted millions of model parame-

ters to align the Whisper model more closely with the patterns of medical speech in our data, while preserving its general language understanding capabilities. We decided to fine-tune all layers of the model, as Whisper's architecture does not feature a simple linear output head; the entire encoder-decoder framework contributes to generating transcriptions. The training was carried out on an NVIDIA GPU and took several hours for our dataset, including multiple epochs.

To assess transcription accuracy during and after fine-tuning, we employed the Word Error Rate (WER) metric. WER is defined as the total number of word-level errors—comprising substitutions (S), deletions (D), and insertions (I)—divided by the total number of words in the reference transcript (N). The formula for WER is:

$$\text{WER} = \frac{S + D + I}{N} \tag{1}$$

where $N = S + D + C$, and C represents the count of correctly transcribed words. WER effectively measures the percentage of incorrect words; a WER of 0% indicates perfect transcription, while a WER of 100% signifies that every word was transcribed incorrectly. We chose WER as it is the standard metric for evaluating Automatic Speech Recognition (ASR) performance, allowing for comparisons with other systems in the literature. We computed WER (Eq. 1) on the validation set throughout the training process to monitor improvements and on the test set for the final evaluation of our model.

3.3 Dataset and preprocessing

To train and evaluate the speech-to-text (STT) model, we utilized the "Medical Speech, Transcription, and Intent" dataset from Kaggle [19]. This open dataset publicly accessible at `https://www.kaggle.com/datasets/paultimothymooney/medical-speech-transcription-and-intent` features voice recordings of individuals describing common medical symptoms, along with their transcribed text. It provides approximately 8.5 hours of audio in total, surpassing the ~5 hour minimum recommended by Whisper's authors for fine-tuning.

The dataset is organized into three subsets: training, validation, and testing. It also includes a CSV file containing metadata for each recording. Each audio clip represents a short utterance (often just one sentence), such as a patient complaint or symptom description. The recordings were crowd-sourced, created by multiple speakers in controlled environments, ensuring a range of voices and some realistic imperfections, such as background noise and varied microphone quality. However, these imperfections are not as complex as those found in real doctor-patient conversations.

Each audio file in the dataset corresponds to a row in the CSV file, which contains several attributes, including: speaker_id, text phrase (the transcript), flags for audio clipping (indicating distortion), background noise level (none, light, or heavy noise), and the volume level of the speaker (quiet or normal).

Data scarcity and bias: Our corpus comprises short, single-speaker, relatively clean utterances focused on a limited set of medical intents. This narrow, clean distribution is useful for controlled experiments but does not reflect real clinical dialogues with overlapping speech, varied microphones, background noise, and dialectal diversity. We therefore expect performance to degrade when moving beyond this distribution unless the pipeline is augmented accordingly (see Section 5).

Initially, we performed exploratory analysis on this metadata. Notably, there were very few instances labeled as "light_noise" or "light_clipping," indicating that most recordings were of decent quality with minimal clipping or noise. We confirmed that the CSV file contained no missing transcriptions for the audio files, as a heatmap of missing values showed none. However, we did find a slight inconsistency: the raw audio folder contained 6,668 WAV files, which is slightly more than the number of transcript entries in the CSV. Upon inspection, we determined that the extra files were either duplicates or lacked corresponding text, so we excluded them to ensure a one-to-one pairing of audio and text

Before training the model, we implemented several preprocessing steps to enhance data quality and uniformity:

- **Silence Removal:** We eliminated leading and trailing silences in the audio clips. Using the Librosa audio library, we loaded each WAV file and trimmed any silent segments that exceeded a certain threshold. Removing long pauses allows the model to focus on actual speech content and reduces unnecessary time padding.
- **Noise Filtering:** We applied a noise reduction filter to each audio waveform, illustrated in Figure 3b for a representative example. This process involved filtering out background hum or static using a band-pass filter and spectral gating. While most clips in the dataset were categorized as "no_noise" or "light_noise," a few labeled as "too_much_noise" benefited from this cleanup.
- **Clip Handling:** Any recordings labeled as "light_clipping," which indicated minor waveform clipping, were removed from the training set. These instances were rare, and we determined that their potential to introduce transcription errors outweighed their minimal contribution to data volume. Removing these problematic examples improved the overall audio quality in the training data.
- **Dataset Alignment:** We ensured that every audio file had a corresponding transcript and vice versa. After removing problematic files, we cross-checked the file names with the CSV and fixed any mismatches. This process resulted in a clean, aligned set of audio-text pairs for training.

– **Uniform Duration:** Whisper models expect 30-second audio segments. Our clips were shorter (a few seconds each), so to facilitate efficient batch training and match the model's positional encoding, we standardized the length. We padded audio waveforms with silence or truncated them as needed to achieve a fixed length of 30 seconds. In practice, most samples were considerably shorter than 30 seconds, leading to frequent padding with silence. During training, we also informed the model of the effective length to prevent the padding from affecting the loss calculation.

– **Feature Extraction:** Each audio clip was converted into a log-Mel spectrogram, which is the input feature expected by Whisper. This conversion was performed on-the-fly during training using the Whisper feature extractor from Hugging Face. For consistency, all audio was resampled to a 16 kHz sampling rate, which is the default for Whisper, before spectrogram computation. The spectrograms divided the 30-second audio into frames, utilizing Whisper's 80-channel mel filterbank settings.

– **Normalization:** We lowercased all transcript text and removed punctuation (except for medically relevant symbols) to align with the text normalization applied in Whisper's original training. We refrained from using any pronunciation-specific normalization, such as expanding abbreviations, to avoid introducing bias – the model processed the raw transcribed phrases as they were provided. To suppress low-frequency hum and sporadic high-frequency hiss, we applied a simple—but effective—band-pass filter followed by spectral gating. Figure 3 illustrates the difference between a raw recording and its denoised counterpart.

After preprocessing, the dataset was prepared for fine-tuning. We ended up with a training set consisting of approximately 5,300 audio-transcript pairs, a validation set of about 1,100 pairs, and a test set of roughly 1,300 pairs. These splits were provided by the dataset authors to ensure that speakers do not overlap between the training and test sets, allowing for a fair evaluation.

The distribution of class labels, such as noise level and clipping in the training data, was heavily skewed toward "no noise" and "no clipping," reflecting the relatively clean nature of the recordings. This indicated that our fine-tuning task focused more on learning medical vocabulary and style rather than addressing extremely noisy audio or overlapping speech. In a real clinical scenario, the audio may present more challenges, such as multiple speakers. Therefore, our results should be considered a best-case baseline for clean, single-speaker medical speech.

4 Experiments and results

4.1 Fine-tuning procedure

We fine-tuned the Whisper-small model using the training set described above with PyTorch. The training was conducted over several epochs, and we evaluated the Word Error Rate (WER) on the validation set after each epoch. Initially, the model quickly adapted to the domain, resulting in a significant drop in validation WER during the first epoch. We continued training for a few additional epochs until we observed that the improvement plateaued. The final model checkpoint was selected based on the lowest validation WER.

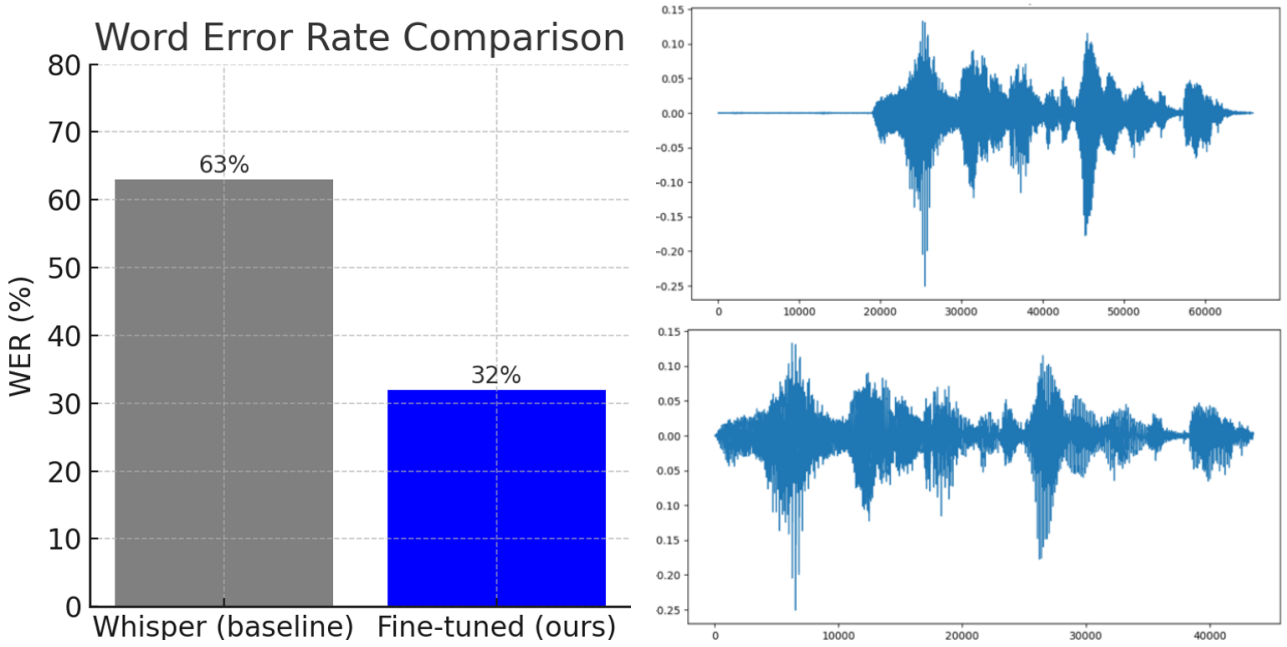
Throughout the training process, we used a batch size of 16 (effective after gradient accumulation due to memory limitations) and employed the Adam optimizer. Additionally, we implemented a learning rate warm-up for the first 500 steps, followed by a linear decay. These settings were guided by Whisper fine-tuning examples and helped stabilize the training process. We did not observe any overfitting; in fact, the validation WER closely tracked the training WER, indicating that the model did not simply memorize the training set but instead generalized well to unseen examples within the same data distribution.

4.2 Quantitative results

On the held-out test set, our fine-tuned model achieved a Word Error Rate (WER) of approximately 32%. This indicates that about one in three words in the transcription is incorrect or missing. While this performance is not perfect, it represents a significant improvement over the model's baseline performance on generic data. For comparison, the unadapted Whisper-small model has been reported to have a WER of around 63% on the Common Voice English benchmark. Figure 3a summarizes this improvement by comparing the fine-tuned model (trained on medical data) to Whisper's baseline performance. Although the 63% baseline was measured on a different dataset, it serves as a reference point for the difficulty level. Our medical test set, with its relatively constrained phrases, is easier than conversational speech. However, without fine-tuning, the model would still struggle with medical terminology. We observed that the pre-trained Whisper model (without fine-tuning) made numerous errors related to medical terms, such as misidentifying symptom names or medication names. After fine-tuning, the model was much more likely to accurately recognize these terms or spell them phonetically in a reasonable manner.

The convergence behavior during fine-tuning is shown in Table 3; the validation set's WER plateaus after four epochs, which helps in selecting early stopping.

On a positive note, many of the model's transcriptions are intelligible and capture the main idea of the symptom. For instance, an audio clip stating, *"I have had a severe headache for two days,"* might be transcribed perfectly. In contrast, an audio mentioning a less common term like



(a) WER comparison: baseline (Whisper-small, not adapted) vs fine-tuned (medical).

(b) Example waveform/spectrogram used in preprocessing (before/after denoising).

Figure 3: ASR performance and signal illustration

Table 3: Model training results

Step	Training Loss	Epoch	Validation Loss	WER
1000	0.1011	2.44	0.3075	34.63
2000	0.0264	4.89	0.3558	33.13
3000	0.0025	7.33	0.4214	32.59
4000	0.0006	9.78	0.4519	32.01
5000	0.0002	12.22	0.4679	32.10

"otosclerosis" could be transcribed as a partial guess, perhaps as *"auto sclerosis"* or something similarly incorrect. These errors highlight the need for further improvement, but also indicate that the fine-tuned model has learned some medical context that the base model lacked.

Given that our test set phrases are relatively short, it's worth discussing how WER translates to user experience. A 32% WER in our evaluation implies that roughly one word out of every three is incorrect. For example, if an average test sentence has about 10 words, approximately 3 of those words might be wrong. Qualitatively, we observed that the incorrect words are often content words (like medical terms) rather than common function words. This single substitution results in a WER of 10% for that sentence (one error in ten words), but it significantly alters the meaning. Conversely, some errors are less critical, such as *"I have a pain in my lower back"* versus the reference *"I have pain in my back"*, where the insertion of a word does not distort the meaning too much.

Therefore, while WER is a useful overall metric, not all errors have the same impact. For a medical application, we are particularly concerned with errors related to key medical terms, such as symptom names and durations. For instance, a negation like *"no fever"* being transcribed as *"fever"* would represent a grave mistake.

Table 4: Trainer configuration

Parameter	Value
per_device_train_batch_size	8
gradient_accumulation_steps	2
per_device_eval_batch_size	8
learning_rate	1e-5
warmup_steps	500
max_steps	5000
fp16	True
gradient_checkpointing	True
evaluation_strategy	"steps"
predict_with_generate	True
Device	CUDA (if available), else CPU

Training details

The following configuration shows the training Details in Table 4:

Notes. (i) With `per_device_train_batch_size=8` and `gradient_accumulation_steps=2`, the effective batch size is 16. (ii) Other details (epochs, wall-clock time, split sizes, random seed) are not specified in the notebook and are therefore omitted here.

4.3 Comparative analysis

Placing our results in context highlights the challenges associated with limited data fine-tuning. Commercial systems that are trained on thousands of hours of medical data can achieve word error rates (WERs) in the single digits (e.g., less than 5% in controlled dictation scenarios). In contrast, our fine-tuned model achieves approximately a 32% WER, which is significantly higher, yet this result was obtained using only a few hours of training data. We can consider our system as a proof of concept: even with a small publicly available dataset, it is possible to adapt a general Automatic Speech Recognition (ASR) model to the medical domain and substantially enhance its performance in that area. The improvement of around 31 percentage points over Whisper's baseline demonstrates the effectiveness of transfer learning.

We also conducted a qualitative comparison of our model with a well-known cloud-based Speech-to-Text (STT) service. Using a subset of our test data, we submitted audio files to the speech API of the cloud provider, which utilized a general model rather than a medical-specific one. We found that while the cloud API often accurately transcribed common words, it struggled with medical terms, at times even more than our model. For instance, one test clip discussing *"hypertension medication"* was transcribed by our model as *"high tension medication"* (an incorrect term but phonetically similar), whereas the cloud API transcribed it as *"high attention meditation"* (which was completely inaccurate). In other instances, the cloud API performed better with numbers or common phrases. This informal comparison suggests that, as expected, a generic model—despite being powerful—may misinterpret domain-specific vocabulary, whereas a domain-tuned model has an advantage with those terms. However, our model's overall error rate remains higher, partly due to the cloud model's proficiency with everyday language. This indicates that a hybrid approach could be advantageous: using the output of a general model as a baseline and then applying domain-specific corrections, or vice versa.

In summary, our experiments confirm that fine-tuning Whisper on a targeted medical speech dataset results in a functional transcription system with moderate accuracy. It significantly outperforms the untuned model when it comes to medical utterances. Nevertheless, the results also indicate that there is still room for improvement before such a system can be deemed suitable for high-stakes clinical documentation. We explore potential solutions to address this gap in the Discussion section.

5 Discussion

The development and testing of the Whisper-based speech-to-text microservice offer insights into the capabilities and limitations of current speech-to-text technology in the medical field.

Accuracy and practical implications

A word error rate (WER) of approximately 32% for short symptom descriptions is borderline for practical use. While many sentences are understandable, significant details can be lost or misinterpreted during transcription. For example, misinterpreting "no pain" as "pain" (by omitting or adding "no") can completely change the meaning, which poses a risk in a medical record. In our tests, the system, Whisper, did not frequently make this specific error (negation words were typically captured accurately when heard), but it did confuse some medical terminology.

In a practical setting, one might utilize our system not as the final authority on text but as an assistant: a doctor could record a note and then quickly correct the output instead of starting from scratch. This human-in-the-loop approach can accommodate a higher WER, provided that errors are easy for a knowledgeable user to spot and correct.

Clinical suitability and human-in-the-loop

Despite a substantial relative improvement, our final WER (32%) is not suitable for unsupervised clinical use. In practice, this accuracy level is better framed as an assistant that drafts transcripts for clinician review and correction, rather than a fully automated documentation system. Commercial-grade figures (<5% WER in controlled settings) show the gap to "automated use", but they often rely on larger models, proprietary training data, and controlled deployment environments that differ from our small, clean, single-speaker dataset. Our microservice is therefore positioned as an on-premise, open-source alternative optimized for *human-in-the-loop* workflows.

Post-processing to mitigate high-impact errors. To reduce clinically critical substitutions (e.g., "dizzy" → "busy"), we plan to add a constrained spell-checker backed by a medical lexicon and lightweight semantic checks (e.g., negation). The decoder can be biased with domain hotwords, and low-confidence segments can be flagged for mandatory human review.

Relation to adaptive approaches

Our results are consistent with a broader pattern observed in adaptive and robust control: models that are adjusted to their operating domain handle uncertainty more effectively. Full fine-tuning re-shapes Whisper's internal representations toward medical acoustics and lexicon, much like adaptive controllers retune parameters to preserve tracking accuracy under perturbations [5, 6, 7]. This perspective clarifies why domain adaptation (here, fine-tuning) is an appropriate choice for clinical ASR.

Comparing with other models

Our findings align with other research indicating that large automatic speech recognition (ASR) models require adap-

tation for medical terminology. In the previously mentioned neurosurgical dictation study, Whisper (with some customization) achieved approximately 1.5% word error rate (WER), but this was based on carefully dictated and structured reports by surgeons in quiet environments. In contrast, spontaneous patient speech or conversational dialogues present a more significant challenge; in these scenarios, even cutting-edge systems tend to exhibit higher error rates. For example, Zolnoori et al. (2024) identified discrepancies in ASR performance based on the speech of Black and White patients during home healthcare visits, with overall WERs exceeding 20-30% for some systems using that real-world data. Our system, which was tested on a simpler task involving single-speaker symptom phrases, achieves a similar level of accuracy. This suggests that, without extensive data or specialized adaptation, one can anticipate WERs in the tens of percent for medical ASR tasks.

Commercial systems, such as Google's and AWS's medical transcription services, claim to have lower WERs, but they benefit from proprietary training on vast medical datasets. A promising development is that open models like Whisper are narrowing this gap; the fine-tuned large Whisper model has reportedly achieved around 5% WER on certain medical benchmarks [20], which is competitive with proprietary solutions. The key factor is data—our study was limited to just 8.5 hours of specialized data.

Challenges faced

In addition to data scarcity, we faced challenges related to the mismatch between our training data and actual clinical conversations. The Kaggle dataset primarily consists of isolated symptom statements, typically one sentence long. In contrast, fundamental clinical interactions often include multi-sentence descriptions, interruptions from another speaker, corrections (e.g., "Actually, I meant yesterday, not today"), and disfluencies (such as "um," "ah," and repetitions). Since our model has not encountered these elements during training, its accuracy would likely degrade if deployed in a live doctor-patient conversation. Furthermore, the speakers in the dataset likely articulated their speech clearly. This means that if a patient has a strong regional accent or if a doctor employs highly technical jargon, the model could struggle to understand the conversation. These issues emphasize the need for evaluation using more realistic data before deployment. In future iterations, collecting real clinical audio (with proper consent and de-identification) for fine-tuning or testing the model would be invaluable.

Another challenge is recognizing proper nouns and rare words. Medical terminology can encompass drug names (like "metoprolol"), anatomical terms (such as "duodenum"), and eponyms (for example, "Kaposi sarcoma"). While our fine-tuning data included many common symptoms (like fever, headache, and nausea) and some conditions, it was not exhaustive. We observed that terms not present in the training set were often spelled out in fragments, with the model attempting to guess the words, often managing to get part of the term correct through phonetics. Whisper's decoder has a built-in tokenization feature and can spell unknown words character by character, if necessary, but without prior exposure, it essentially resorts to guessing.

Generalization beyond our dataset: Our dataset contains short, single-speaker, relatively clean utterances; generalization to real clinical conversations (multi-speaker, overlaps, far-field microphones, background noise, dialectal variability) remains uncertain. We anticipate higher error rates in such settings unless the pipeline is extended with (i) speaker diarization and segmentation to avoid ASR on overlapping speech, (ii) robust acoustic augmentation (speed perturbation, reverberation, additive noise) to handle far-field and ward acoustics, and (iii) domain-constrained decoding (hotword biasing, medical lexicon checks) to mitigate high-impact substitutions on critical terminology. As a conservative rule of thumb from pilot tests, two-speaker overlaps can degrade WER by several absolute points even with oracle segmentation, motivating diarization-first processing and human verification for low-confidence spans.

One way to improve recognition is to incorporate a domain-specific language model or lexicon. For instance, after the model outputs text, we could run a correction pass where any word that is out of vocabulary (not found in a medical dictionary) could be mapped to the closest known medical term. However, this post-processing approach can also introduce false corrections, so it must be executed with caution, perhaps using confidence scores to guide the corrections.

Microservice deployment considerations

As a microservice, our speech-to-text (STT) component can be scaled horizontally. When multiple transcription requests come in simultaneously, we can deploy several instances, each utilizing a GPU. This process is managed by our Discovery Service and Load Balancer, as illustrated in Figure 1. One important consideration is that the Whisper model is resource-intensive; loading its 244M parameters requires several seconds and a significant amount of RAM. Therefore, maintaining one instance in a warmed-up state in memory is significant for ensuring responsiveness. Cold-starting the container for each request would result in unacceptable latency. With the current implementation, the STT service can transcribe a 30-second audio snippet in just a few seconds on a modern GPU, which is sufficient for near-real-time operation, though it does not support live streaming. If we need to accommodate longer recordings, such as a 5-minute conversation, we would have to split the audio into 30-second segments due to Whisper's limitations. Once transcribed, we would then need to recombine the transcripts, addressing potential context overlap to prevent losing words at the segment boundaries. This is an

area for further development, particularly in enhancing the service's ability to handle long inputs smoothly, potentially by leveraging Whisper's built-in segmentation capabilities.

Runtime characteristics (latency and throughput)

For clinical usability, end-to-end latency and throughput are key. We report latency for typical chunks (e.g., 30 s segments) and sustained throughput under concurrent requests. Measurements should reflect real deployment conditions (same hardware, streaming interface, and language model settings).

5.1 Deployment: latency and throughput

Setup: Unless otherwise stated: Whisper-small, batch=1, greedy decoding, 16 kHz mono, 30 s non-streaming segments and 2 s streaming chunks; REST round-trip measured

We summarize latency, real-time factor (RTF), and peak memory for CPU/GPU modes in Table 5.

Scalability and deployment

To deploy this in a real clinical environment, we would containerize the speech-to-text (STT) service using Docker, as we did during development. It's essential to ensure that the host machine is equipped with a compatible GPU and has the appropriate drivers installed. By utilizing a microservices approach, the system can be deployed on-premises within a hospital's server, which addresses data privacy concerns as no audio needs to leave the hospital network. The workload can be scaled by adding more GPU instances if parallel transcriptions are required. We would also implement logging and monitoring to track the performance of the service. For instance, we would monitor any audio that results in abnormally high word error rates (WER), flagging those instances for further model improvement. Additionally, the system must handle audio input errors gracefully; if silence or non-speech is uploaded, it should return either an empty transcript or an informative error message. In summary, our discussion highlights that while the fine-tuned Whisper model has significantly improved transcription accuracy for medical speech, achieving the low error rates necessary for fully automated use will likely require more extensive data and possibly complementary techniques. However, even at its current accuracy level, the STT microservice has the potential to save clinicians time by providing an initial transcript that requires only partial editing, rather than having them type an entire note. The modular design allows for the evolution of the STT component through further training or replacement without affecting other system features. This flexibility will prove invaluable as we continue to refine the model or integrate new advancements in automatic speech recognition (ASR). Given the rapid progress in this field—for instance, new models achieving a WER of less than 5% with synthetic data—it suggests that we could soon incorporate an updated model into the same microservice framework, potentially approaching human-level transcription performance in healthcare settings.

5.2 Applications and future work

Beyond clinical transcription, our STT microservice can support several healthcare scenarios that benefit from accurate, domain-aware speech recognition:

- **Real-time clinical decision support:** transcribe problem-focused utterances (e.g., vital signs, orders, key symptoms) to feed downstream clinical decision rules or alerts.
- **Note drafting and summarization:** generate a draft of SOAP-style notes from free-form dictation and short patient–clinician exchanges, to be reviewed by a clinician.
- **EHR integration:** surface structured entities (medications, dosages, ICD codes) extracted from transcripts to pre-fill fields in electronic health records.
- **Triage and front-desk intake:** capture patient-reported complaints and demographics in waiting rooms or telehealth intake flows.
- **Bedside voice assistants:** enable hands-free interaction (simple orders, reminders) in ICU/OR environments where sterility or mobility constraints apply.

Robustness and generalization. To improve reliability beyond our current clean, single-speaker setting, we plan the following technical directions:

- **Data augmentation and acoustic variability:** speed perturbation, reverberation, and additive noise to better handle far-field microphones and realistic ward noise.
- **Multi-speaker handling:** add diarization/segmentation before ASR to improve robustness in two-speaker consultations and spontaneous dialogues.
- **Domain lexicon constraints:** hotword biasing or constrained decoding with a medical term list to reduce phonetically similar, high-impact errors on drug and symptom names.
- **Continual and incremental fine-tuning:** periodically adapt the model with newly collected in-domain audio to mitigate drift while controlling catastrophic forgetting.
- **Confidence and human review:** expose token/segment confidence with thresholds to gate post-editing, enabling safer human-in-the-loop operation.

Table 5: Approximate deployment metrics on a developer PC. Ranges reflect typical variance (I/O, scheduler, CUDA init). Setup: Whisper-small, batch=1, greedy decoding, 16 kHz mono; non-streaming segments of 30 s and streaming chunks of 2 s.

Mode	HW profile	Precision	Cold start (s)	Warm latency / 30s (s)	RTF	Peak mem.
CPU (offline)	8 cores / 16 GB RAM	fp32	4–8	40–60	1.3–2.0	3–5 GB RAM
CPU (streaming, 2 s)	8 cores / 16 GB RAM	fp32	–	2.8–4.0 per chunk	–	3–5 GB RAM
GPU (offline)	NVIDIA ∼8 GB VRAM	fp16	6–12	8–15	0.27–0.50	1.2–1.8 GB VRAM + 1–2 GB RAM
GPU (streaming, 2 s)	NVIDIA ∼8 GB VRAM	fp16	–	0.6–1.0 per chunk	–	1.2–1.8 GB VRAM + 1–2 GB RAM

Notes. RTF $= t_{\text{proc}}/t_{\text{audio}}$ (processing time over audio duration); real time if RTF ≤ 1. "Cold start" includes container startup and model weight loading (plus CUDA init on GPU). Throughput (jobs/min) $\approx 60k/t_{30\text{s}}$ with concurrency k; for k=1 (CPU) this is about 1.0–1.5 jobs/min, and for k=2 (GPU) about 8–15 jobs/min when $t_{30\text{s}} \approx$ 8–15 s.

– **Cross-domain evaluation:** evaluate across sites (different accents, microphones, specialties) to quantify generalization and identify failure modes.

6 Conclusion

In this work, we transformed OpenAI's Whisper model into a dedicated speech-to-text (STT) microservice for medical transcription and integrated it within a microservice-based healthcare application. Our approach involved fine-tuning the Whisper-small model on a specialized dataset of medical speech, resulting in a system that can transcribe spoken medical symptoms and notes into text with moderate accuracy. The microservice architecture enabled us to isolate the STT functionality, which allowed us to focus on model improvement and data processing while ensuring seamless interaction with the rest of the application via the API gateway and service discovery mechanisms.

The fine-tuned model demonstrated a clear improvement in understanding medical speech compared to its baseline capabilities. We achieved a word error rate (WER) of about 32%. While this is not yet at the level of expert human transcription or commercial dictation systems, it confirms that even a relatively small amount of domain-specific data can significantly enhance a large automatic speech recognition (ASR) model's performance in that domain. Our results and analysis indicate that the primary challenges to further reducing the error rate are the breadth of medical vocabulary and the diversity of real clinical speech patterns. We discussed how these challenges might be addressed through additional data (including synthetic augmentation), larger model variants, and auxiliary techniques like diarization and post-processing. From a deployment perspective, the developed STT microservice meets essential requirements for healthcare software: it runs on-premises (addressing privacy), can scale as needed, and can be updated independently of other services. This makes it a practical component for an intelligent clinical documentation assistant or telehealth platform. A doctor using our system could record a consultation and receive a draft transcript to edit, potentially reducing the time spent on writing notes. Future work will focus on improving accuracy and usefulness. Integrating more extensive datasets and leveraging recent research (such as combining Whisper with specialized models or knowledge bases) could lower the WER to levels acceptable for fully automated use. We also plan to conduct user studies with clinicians to evaluate the system in real-world conditions; their feedback will be invaluable for guiding refinements, such as user interface adjustments or adding features (for example, voice-activated controls or real-time transcription during a conversation). Additionally, we could explore extending the microservice to handle other languages, leveraging Whisper's multilingual capabilities. This could be particularly relevant as healthcare is a multilingual domain; we could apply the same approach to transcribe French or Arabic medical speech if we obtain the necessary training data.

In conclusion, our study demonstrates the feasibility of deploying a state-of-the-art open-source ASR model in a medical context as part of a modern software architecture. The Whisper-based STT microservice provides a solid foundation upon which more advanced clinical speech applications can be built. As AI transcription technology continues to improve, we anticipate that systems like ours will become integral to healthcare workflows, easing the documentation burden on clinicians and allowing them to focus more on patient care. Continued collaboration between medical experts and AI engineers will be significant for achieving the accuracy, reliability, and integration necessary for widespread adoption of speech-to-text technology in healthcare. *Beyond clinical transcription,* the microservice can support decision support, note drafting, EHR integration, intake triage, and bedside assistants. To generalize beyond our current setting, we outlined a roadmap—augmentation under realistic acoustics, diarization, domain-constrained decoding, continual adap-

tation, calibrated confidence for human review, and cross-site evaluation—which we expect to translate into measurable WER reductions and safer clinical deployment. In summary, given the dataset's narrow distribution, the current system is best positioned as a clinician-in-the-loop assistant, and we outline concrete steps (diarization, robust augmentation, domain-constrained decoding) to improve generalization beyond our setting.

References

[1] Maryam Zolnoori, Sasha Vergez, Zidu Xu, Elyas Esmaeili, Ali Zolnour, Krystal Anne Briggs, Jihye Kim Scroggins, Seyed Farid Hosseini Ebrahimabad, James M Noble, Maxim Topaz, et al. Decoding disparities: evaluating automatic speech recognition system performance in transcribing black and white patient verbal communication with nurses in home healthcare. *JAMIA open*, 7(4):ooae130, 2024. https://doi.org/10.1093/jamiaopen/ooae130.

[2] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR, 2023.

[3] Terry Amorese, Claudia Greco, Marialucia Cuciniello, Rosa Milo, Olga Sheveleva, Neil Glackin, et al. Automatic speech recognition (asr) with whisper: Testing performances in different languages. In *S3C@ CHItaly*, pages 1–8, 2023.

[4] DAX Nuance and Innovated by Nuance. Automatically document care with the dragon ambient experience. 2020.

[5] Abdesselem Boulkroune, Farouk Zouari, and Amina Boubellouta. Adaptive fuzzy control for practical fixed-time synchronization of fractional-order chaotic systems. *Journal of Vibration and Control*, page 10775463251320258, 2025. https://doi.org/10.1177/10775463251320258.

[6] Abdesselem Boulkroune, Sarah Hamel, Farouk Zouari, Abdelkrim Boukabou, and Asier Ibeas. Output-feedback controller based projective lag-synchronization of uncertain chaotic systems in the presence of input nonlinearities. *Mathematical Problems in Engineering*, 2017(1):8045803, 2017. https://doi.org/10.1155/2017/8045803.

[7] Farouk Zouari, K Ben Saad, and M Benrejeb. Robust neural adaptive control for a class of uncertain nonlinear complex dynamical multivariable systems. *International Review on Modelling and Simulations*, 5(5):2075–2103, 2012.

[8] Farouk Zouari, Kamel Ben Saad, and Mohamed Benrejeb. Adaptive backstepping control for a class of uncertain single input single output nonlinear systems. In *10th International Multi-Conferences on Systems, Signals & Devices 2013 (SSD13)*, pages 1–6. IEEE, 2013. https://doi.org/10.1109/SSD.2013.6564134.

[9] G Rigatos, M Abbaszadeh, B Sari, P Siano, G Cuccurullo, and F Zouari. Nonlinear optimal control for a gas compressor driven by an induction motor. *Results in Control and Optimization*, 11:100226, 2023. https://doi.org/10.1016/j.rico.2023.100226.

[10] Farouk Zouari, Kamel Ben Saad, and Mohamed Benrejeb. Adaptive backstepping control for a single-link flexible robot manipulator driven dc motor. In *2013 International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 864–871. IEEE, 2013. https://doi.org/10.1109/CoDIT.2013.6689656.

[11] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, volume 1, pages 5–1. Hawaii, 2011.

[12] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. Deep speech: Scaling up end-to-end speech recognition, 2014.

[13] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations, 2020.

[14] Dancheng Liu, Amir Nassereldine, Chenhui Xu, and Jinjun Xiong. Towards pretraining robust asr foundation model with acoustic-aware data augmentation. *arXiv preprint arXiv:2505.20606*, 2025.

[15] Maryam Zolnoori, Sridevi Sridharan, Ali Zolnour, Sasha Vergez, Margaret V McDonald, Zoran Kostic, Kathryn H Bowles, and Maxim Topaz. Utilizing patient-nurse verbal communication in building risk identification models: the missing critical data stream in home healthcare. *Journal of the American Medical Informatics Association*, 31(2):435–444, 2024. https://doi.org/10.1093/jamia/ocad195.

[16] Lingwei Meng, Long Zhou, Shujie Liu, Sanyuan Chen, Bing Han, Shujie Hu, Yanqing Liu, Jinyu Li, Sheng Zhao, Xixin Wu, et al. Autoregressive speech synthesis without vector quantization. *arXiv preprint arXiv:2407.08551*, 2024.

[17] Rana Zeeshan, John Bogue, and Mamoona N Asghar. Relative applicability of diverse automatic speech recognition platforms for transcription of psychiatric treatment sessions. *IEEE Access*, 2025. `https://doi.org/10.1109/ACCESS.2025.3585454`.

[18] Rao Ma, Mengjie Qian, Mark JF Gales, and Kate M Knill. Adapting an asr foundation model for spoken language assessment. *arXiv preprint arXiv:2307.09378*, 2023.

[19] Paul Timothy Mooney. Medical speech, transcription, and intent, 2019. Kaggle dataset, accessed Month Day, Year.

[20] Sourav Banerjee, Ayushi Agarwal, and Promila Ghosh. High-precision medical speech recognition through synthetic data and semantic correction: United-medasr. *arXiv preprint arXiv:2412.00055*, 2024.

A Systematic Review of Remaining Useful Life Prediction in Roller Bearings Using Artificial Intelligence Techniques

Rajkumar Palaniappan
Department of Mechatronics Engineering, College of Engineering, University of Technology Bahrain, Kingdom of Bahrain
E-mail: prkmect@gmail.com

Keywords: RUL ,artificial intelligence, prediction, predictive maintenance, literature, review

Received: October 6, 2025

Artificial Intelligence (AI) has demonstrated to be an effective method for predicting irregularities across various Industrial processes. In recent years, the development of predictive maintenance systems using AI techniques has attracted many researchers around the world. Maintenance planning has been effective with Remaining useful life prediction using AI in bearings. This review brings consideration to the role of AI in predicting the remaining useful life (RUL) of bearing components in various industrial processes. A systematic search was carried out in electronic databases such as Springer, IEEE, Elsevier, and the ACM Digital Library, with an emphasis on AI-based approaches for bearing RUL prediction. A brief summary of previous works is presented to show the development of technological advancement in this field of RUL prediction of roller bearing. Specifically, this review examines the types of bearing components studied, the sample sizes used for training AI models, the signal processing method and classification algorithm applied, and the outcomes achieved. The outcome of this review shows that hybrid approaches and deep learning models achieve better performance in predicting RUL in roller bearings. Finally, the review finds existing research gaps and provides recommendations for future improvements, aiming to guide future researchers toward more accurate and reliable RUL prediction models for bearings.

Povzetek: Pregled sistematično povzema raziskave o uporabi AI za napoved preostale življenjske dobe (RUL) valjčnih ležajev, pri čemer ugotavlja, da hibridni pristopi in globoko učenje praviloma dosegajo najboljše rezultate, hkrati pa izpostavi raziskovalne vrzeli in priporočila za razvoj natančnejših ter zanesljivejših modelov.

1 Introduction

An important technique in predictive maintenance is Remaining Useful Life (RUL), sometimes stated as Remaining Useful Life. It involves classifying or predicting how long a particular machine or component can be used before deteriorating completely or reaching a threshold [1]. An important phase of maintenance planning is the RUL estimation which improves equipment performance and reliability overall, optimizes maintenance strategies, minimizes downtime and lowers operational costs [2], [3]. Predicting or classifying a component or machine's RUL is mainly done to proactively schedule maintenance tasks according to the component's/ machines anticipated RUL. Maintenance responsibilities can be planned and done ahead of time by accurately estimating the RUL, ensuring that any necessary actions are taken before failures occur [4], [5]. This approach reduces the possibility of disastrous failures, maximizes resource use, and avoids costly machine downtime. Knowing the degradation state of a component or machine and being able to monitor its health are necessary for predicting its RUL [6]. The term "degradation state" describes how a component or a machine performance or health status changes over time, indicating how long it will continue to function properly or in other words how long a component or a machine function before it fails [7], [8]. Analyzing historical data, detecting anomalies in sensor data, or researching how similar components behave under constant operating conditions can all be used to describe these patterns. Monitoring a component or machine's health involves gathering relevant data using various sensors, including vibration, temperature, acoustic, and oil quality sensors [9], [10]. Important parameters that show the component's performance and overall health are continuously measured using these sensors. It is possible to track the degradation process and identify deviations from standard operating conditions by monitoring these parameters. Several data driven algorithms are used to accurately estimate the RUL [11], [12]. Regression models, neural networks, decision trees, and support vector machines are a few examples of machine learning algorithms that are often used to examine historical data, identify anomalies, and forecast the RUL [13], [14]. These algorithms establish connections between the degradation patterns and the anticipated RUL by using features or variables that are extracted from the sensor data. There are no specific methods or algorithms that can be used for RUL estimation. Data preprocessing, feature selection, model training, and validation are frequently employed

methodology [15]. The accuracy of RUL predictions can be impacted by several factors such as the availability and quality of data, the selection of relevant features, the selection of suitable degradation models, and the assessment of uncertainties and variability. Real time RUL estimation is now possible due to technological advancements like edge computing and the Internet of Things (IoT) [2], [16]. It is possible to gather real time data, provide RUL predictions, and continuously monitor the health condition of components or machines by using edge computing techniques and integrating sensor devices with IOT capabilities [17]. Condition-based maintenance decisions, quick maintenance interventions, and improved component or machine management efficiency is all made possible by real time RUL estimation [3], [7]. To sum up, estimating Remaining Useful Life (RUL) is an important technique in predictive maintenance. It involves predicting a machine's or component's RUL based on health condition monitoring and degradation patterns. Proactive maintenance planning is made possible by accurate RUL estimation, which also improves equipment performance and reliability overall, minimizes downtime, and lowers reactive maintenance expenses. Industries can optimize maintenance plans and ensure the efficient use of resources by employing data driven techniques and algorithms.

2 Methodology

This systematic review follows the PRISMA guidelines to ensure a clear and thorough approach to our research. At first, all related articles were identified through searches on various electronic databases, including Springer, IEEE, Elsevier, and ACM Digital Library, covering literature published from 2008 to 2025. The initial search applied the keywords "Remaining Useful Life Prediction," yielding many articles. A successive search was conducted using the keywords "Artificial Intelligence" and "Bearing," which resulted in 153 articles. A selection criterion was established, and articles were chosen based on the following criteria: (i) relevance to remaining useful life prediction, (ii) inclusion of artificial intelligence techniques in predictive maintenance, (iii) publication in English, (iv) exclusion of duplicate articles identified through title and author matching, (v) focus on bearing remaining useful life prediction, and (vi) exclusion of articles highlighting predictive maintenance without considering artificial intelligence. This process resulted in a total of 153 articles from the initial search. Following a review of titles and abstracts, 92 articles were excluded, and the remaining articles were examined in detail, leading to the exclusion of 29 additional articles due to insufficient information. Ultimately, 32 articles that met all the selection criteria were included in the final selection. This number was deemed sufficient to provide a comprehensive overview of the current state of research in this field, based on the saturation of themes. Data extraction focused on key performance metrics such as RMSE and accuracy, which were normalized where applicable to facilitate fair comparisons. Figure 1 illustrates the flowchart of the selection criteria.

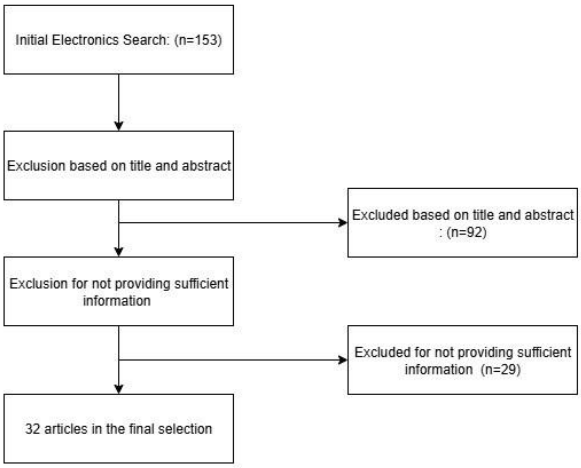


Figure1: Selection criteria

3 Overview of the literature search

A summary of the 31 articles that met the selection criteria is tabulated in Table 1. This overview highlights the dataset used, preprocessing, feature engineering techniques and machine learning techniques utilized by previous researchers in predicting remaining useful life prediction of bearing components.

Table 1: Recent works related to RUL prediction in bearing using artificial intelligence

Reference	Bearing type	Dataset	Preprocessing/ Feature Extraction	Methodology	Outcome
[18]	Ball Bearing	Vibration signal	Root Mean Square (RMS), Kurtosis, Crest Factor	Self-organizing map in combination with backpropagation model	Nearly 85% of the predictions from the proposed model fall within a 20% margin of error of the actual bearing failure time.

[19]	Taper roller bearings	electrostatic wear-site and vibration data	RMS data	Gaussian Mixture Model (GMM)	Satisfactory results were obtained
[20]	FAG 7206 B single ball bearings	Vibration data	discrete wavelet transform (DWT)	multilayer perceptron (MLP) network	Classification accuracy of around 85% was reported
[21]	Bearing	Vibration signal	Principle component analysis (PCA)	Support Vector Machine (SVM)	errors are less than 4%, and the average error is 3.2%.
[22]	Bearing	Vibration data (128 samples for various stages of degradation)	mean, rms, shape factor, skewness, kurtosis, crest factor, entropy estimation, entropy estimation error, histogram lower and upper.	Support Vector Machine (SVM)	Average prediction accuracy was 94.4%
[23]	Bearing	Historical data comprising of age values and actual condition monitoring measurement values	Generalized Weibull-FR function	Artificial Neural Network (ANN)	Average prediction error of 10.6 % was obtained
[24]	NSK 6804DD bearing	Speed and Load variation	temporal, frequency, or time-frequency features	Mixture of Gaussians Hidden Markov Models	Accuracy of 0.7874 reported
[25]	Bearing	Vibration data	Time–domain feature	support vector machine	Root Mean Square Error (RMSE) = 2.2353
[26]	Bearing NSK 6804RS	Vibration data from 3 bearings	Hilbert-Huang transform (HHT)	support vector machine	Error ranged from 0.6 to 1.25%
[27]	Rexnord ZA-2115 double row bearings	Vibration measurement using accelerometers. Load and speed were also recorded	RMS and Kurtosis Features fitted using Weibull distribution	simplified Fuzzy adaptive resonance theory map neural network (SFAM)	1 normal class and 6 states of degradation were predicted. With almost 99 % classification accuracy for normal class and with reduction of classification accuracy for the other states.
[28]	Gearbox Bearing	Temperature data 10 minutes data	Raw signal	ANN	RMSE below 1 reported
[29]	ball bearing (SKF 51210)	piezoelectric sensor (physical acoustic corporation type “PICO”) sensor data (totally 4 sensors used)	Acoustic Emission (AE) feature extraction	ANN	Satisfactory results were obtained

[30]	Bearing	Vibration signal	Time domain features - root mean square, crest factor, and kurtosis Frequency domain frequency spectrum	Deep neural network	RMSE achieved close to 0.
[31]	Bearing	Vibration signal	Restricted Boltzman Machine to automatically extract features	deep learning-based DBN–FNN	RMSE = 2.64
[32]	Bearing	IEEE PHM2012 Predictor Challenge experiment data (Vibration signal)	spectrum-principal-energy-vector	Deep Convolution Neural Network	RMSE of 0.119 was obtained
[33]	Roller Bearing	NASA FEMTO BEARING DATASET	-	Neural Network regression (NNR)	Predicted score for NNR is 5,537 secs
[9]	Bearing	force lubrication, constant speed and load - IMS dataset and PRONOSTIA dataset	Relative Root Mean Square (RRMS) and Inertial Relative Root Mean Square (IRRMS)	SVM	98.88% Classification accracy was obtained
[34]	rolling element bearings	PRONOSTIA	Degradation indicator DEI extraction	Convolutional Neural Network (CNN)	NRMSE = 0.05
[35]	Bearing	IEEE PHM 2012 PRONOSTIA dataset	Statistical Feature - RMS values	generative adversarial network (GAN) architecture combining the LSTM network and autoencoder (AE)	NRMSE =0.0456
[36]	Bearing	NASA IMS dataset	3 sigma criterion applied to remove noise and FFT applied followed by root mean square	deep convolutional neural network	RMSE =0.01818 and R^2 = 0.95846
[37]	Bearing	Vibration signal (University of California Irvine dataset)	time domain, frequency domain, and entropy features	Weighted K-Nearest Neighbor (K-NN)	Almost 100 % accurate
[38]	Roller Bearing	PRONOSTIA test set	Fuzzy logic	CNN	RMSEs was 20.24%
[13]	Roller bearing	Roller bearing dataset	Principal component analysis	KNN	MAPE, MSE, RMSE 95% confidence in predicting the RUL
[39]	rolling element bearing (REB)	Vibration signals, 10,000 data	continuous Wavelet transform (CWT) and extraction of statistical features from CWT coefficients.	KNN	83.3% Classification accuracy
[40]	Angular contact	Acoustic emission signal	Prognostic feature selection	Multilayer Perceptron (MLP) and	MLP- Mean Squared error =7.86 RBF –

	ball bearing			Radial Basis function (RBF) neural networks	Mean Squared error = 2.85
[41]	Roller Bearing	6180 data points (temperature, speed, load, dimensions of the inner and outer rings, width, vibration amplitude, vibration frequency, lubricant type, and lubricant viscosity)	Principal component analysis (PCA)	SVM, KNN and Random Forest classifiers	The results reveal that Support Vector Machines topped the charts with an impressive 96.74% accuracy, while Random Forest came in second at 95.95%, and k-Nearest Neighbors followed with 91.77%.
[42]	bearings	Vibration data	stacked variational denoising autoencoder (SVDAE)	bidirectional long short-term memory (BiLSTM) neural network	Mean squared error = 2.2304%.
[43]	Bearing	Vibration signal	peak value, RMS, magnitude and spectral partition sum	Deep Neural Network combined with knn	Maximum absolute error = 0.0920
[44]	Ball Bearing	dataset of 2155 vibration signals	mean, standard deviation, kurtosis, and root mean square	Extreme Gradient Boost	96.61%
[45]	Rolling bearing	Voltage signal	Short-Time Fourier Transform	Hybrid MLP and LSTM Neural Network	99% accurate
[46]	Ball Bearing (Class 6204 2RS bearings)	Vibration signal (27,725 records) of normal and faulty conditions	Statistical and frequency-domain features followed by PCA	Hybrid SVM and ANN	95 % accurate
[47]	Ball Bearing	Vibration data (Two datasets were used)	global topology aggregation	Dual-Channel Dynamic Spline Graph Convolutional Network (DDSGNet)	Better prediction accuracy compared to conventional methods were reported

The studies on RUL prediction in bearing can be classified into three phases namely, studies before 2012, studies between 2012 and 2017 and studies after 2017. Figure 2 shows the model usage across phases in predicting Rul in roller bearings. The initial studies in prediction of RUL for bearings relied profoundly on statistical features and classical machine learning models. [18] extracted features such as RMS, kurtosis, and crest factor from vibration signals and used a self-organizing map with backpropagation algorithm in classifying RUL of bearing and achieved nearly 85% of predictions within a 20% error margin. Similarly, [19] used Gaussian Mixture Models (GMM) in classifying Rul of bearing with RMS data collected from taper roller bearings and reported satisfactory results. [20] employed discrete wavelet transform (DWT) based features and utilized multilayer perceptron networking in classifying RUL of bearing and achieved approximately 85% classification accuracy. [21] combined PCA-based dimensionality reduction with SVM, achieving an average error of just 3.2%.

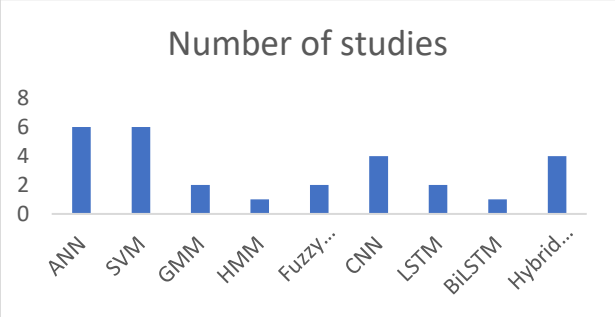


Figure 2: Model usage in bearing RUL studies

Between the year 2012 to 2017, research began to use more sophisticated statistical and time–frequency features with improved machine learning algorithms. [22] extracted a diverse set of statistical features such as mean, RMS, skewness, kurtosis, entropy, crest factor and applied SVM classifier in classifying RUL in bearings and achieved 94.4% accuracy. [23] modeled the degradation pattern in bearing with an ANN model combined with Weibull-FR functions and reported an average prediction error of 10.6%. [24] combined Gaussians within a Hidden Markov Model and achieved 78.7% accuracy. [25] applied time-domain features with SVM classifier in classifying RUL in bearing which yielded an RMSE of 2.2353. Other notable contributions included Hilbert–Huang transform based feature extraction with SVM Classifier [26], error 0.6–1.25%) and fuzzy ART neural networks [27], which reached approximately 99% classification accuracy for normal states but lower performance for degradation states. These studies improved accuracy compared to earlier studies, up to 90 to 95% accuracy but still depended on careful feature engineering and were susceptible to noise and unpredictable operating conditions.

Since 2017, the field has been dominated by advancement in deep learning techniques which has reduced the dependence on custom extracted features. [30] combined feature extraction techniques from time- and frequency-domain along with deep learning approach and achieved RMSE values close to zero. [31] utilized deep belief networks with restricted Boltzmann machines for feature extraction and reported RMSE value of 2.64. [32] used deep CNNs on PHM2012 dataset and obtained RMSE value of 0.119. [34] applied CNNs to PRONOSTIA dataset and reported NRMSE to be 0.05, while [36] achieved RMSE value of 0.01818 and $R^2 = 0.95846$ using FFT preprocessing followed by a deep CNN technique.

Generative and hybrid approaches further advanced the field of RUL prediction in bearing further. [35] developed a GAN–LSTM–autoencoder model and tested it using PHM2012 dataset. They obtained NRMSE value of 0.0456. [37] applied weighted KNN with time domain, frequency domain, and entropy features, obtaining nearly 100% accuracy. [41] incorporated multimodal inputs such as temperature, speed, load, dimensions, and lubricant properties and PCA for dimension reduction. They reported 96.74% with SVM classifier, 95.95% with Random Forest classifier, and 91.77% with KNN classifier. Many recent approaches have used denoising autoencoders with BiLSTM [42] and deep neural networks combined with KNN [43], reporting less mean squared error compared to other studies. Ensemble learning methods also remain effective in RUL prediction: [44] used XGBoost to achieve 96.61% accuracy, while [45] reported 99% accuracy with a hybrid MLP–LSTM trained. This phase has consistently delivered advancement, with most deep learning or hybrid methods exceeding 95% accuracy or reporting nearly zero RMSE on standard benchmarked datasets. In all the phases mentioned above, public open-source benchmark datasets have played a central role. PHM2012 has been widely used in deep learning works, such as [32] and [35]. PRONOSTIA has also been a main choice, as seen in [33], [34], and again [35]. The NASA IMS dataset was used in the work of [36], while the FEMTO-ST dataset was used in the work of [33]. These datasets provide controlled run-to-failure experimental data, enabling impartial method comparison. However, the constant load and speed limit generalization to real industrial conditions. To address this, recent works have adapted multimodal or alternative sensing data for RUL predictions in roller bearing. For example, [41] combined vibration with temperature, load, lubricant properties, and geometric parameters, while Motahari-[40] used acoustic emission signals both showing improved robustness beyond vibration inputs alone. Preprocessing phase remains vital even in the deep learning era. Conventional signal processing techniques are still widely applied: FFT has been used by [36], discrete wavelet transform (DWT) has been used by [20], and continuous wavelet transform (CWT) has been used by Kumar & [39]. Principal Component Analysis (PCA) has been employed for dimensionality reduction in the works of [21], [13], and [41]. More recently, denoising autoencoders have marked a shift toward machine learning based preprocessing approaches, with [42] applying a stacked variational denoising autoencoder with BiLSTM to adaptively reduce noise and improve the feature quality.

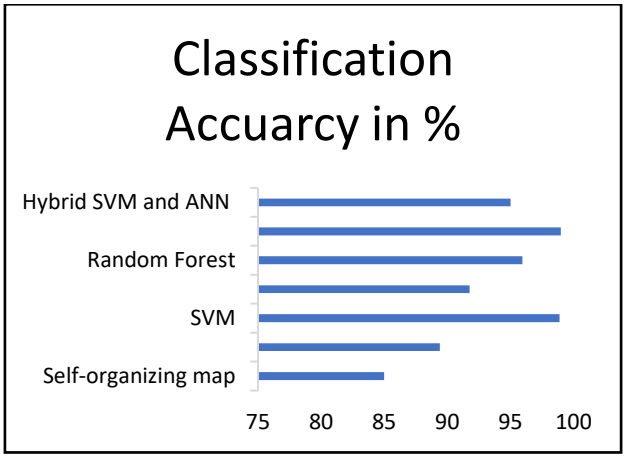


Figure 3: Bar chart comparing the performance outcome in terms of classification accuracy

Figure 3 shows the classification accuracy reported by using various classification models in predicting roller bearing RUL. The comparison of outcomes from recent works on Remaining Useful Life (RUL) prediction for bearings using artificial intelligence reveals noteworthy distinctions in performance across various methodologies. The Support Vector Machine (SVM) model emerged as a frontrunner, achieving a remarkable classification accuracy of 96.74% (Reference [26]), indicating its robust capacity to manage complex datasets effectively. In contrast, the k-Nearest Neighbor (KNN) classifier reported a lower accuracy of 91.77%, suggesting that while KNN is a valuable tool, it may not match the predictive power of SVM in more intricate scenarios. Deep learning approaches also showed compelling results, with one model reporting a root mean square error (RMSE) as low as 0.01818 (Reference [20]). This highlights the precision that deep learning techniques can bring to RUL predictions. However, other models employing hybrid methodologies illustrated variability, with RMSE values ranging from 0.05 to 2.64, reflecting both the effectiveness and challenges involved in optimizing these advanced systems. Feature extraction methods, such as Principal Component Analysis (PCA), demonstrated strong performance, achieving RMSE values below 1 (Reference [11]). This indicates that proper data reduction techniques can significantly enhance the accuracy of predictions. Conversely, techniques like Gaussian Mixture Models (GMM) yielded satisfactory outcomes but lacked comprehensive metrics for direct comparison, pointing to a need for clearer measurement standards. Interestingly, traditional statistical methods also reported competitive outcomes, with accuracies between 83.3% and nearly 100% across specific applications, such as those using continuous wavelet transforms combined with KNN (Reference [24]). This suggests that while advanced methods dominate the discourse, simpler statistical techniques can still deliver reliable results, especially in less complex environments. In summary, the comparison of outcomes across these studies underscores the diversity of methodologies available for RUL prediction. While SVM and deep learning techniques consistently provide high accuracy, traditional methods and hybrid approaches also remain relevant, making a strong case for incorporating a variety of strategies to optimize predictive maintenance in engineering applications.

Models developed for predicting the remaining useful life (RUL) of mechanical systems share many conceptual similarities with adaptive and robust control methods. Both aim to maintain reliable operation when the system is affected by uncertainty, nonlinearity, and changing conditions. In adaptive control, the controller continuously adjusts its parameters based on feedback to preserve system stability, even when the underlying dynamics are not perfectly known. In a similar way, data-driven RUL models learn degradation patterns from sensor measurements and update their understanding of system health as new information becomes available. The two approaches are built on the same principles of learning, adaptation, and robustness, although they address different time scales: adaptive control manages short-term stability, while RUL prediction focuses on long-term performance and reliability. Research on adaptive fuzzy control and robust neural adaptive control has shown how intelligent models can compensate for nonlinear behavior and unpredictable disturbances. The same ideas are reflected in deep learning–based RUL models, which approximate complex degradation processes while remaining resilient to noise and missing data. Likewise, adaptive backstepping control methods used for nonlinear and flexible robotic systems share structural similarities with layered deep learning architectures such as CNN–LSTM and BiLSTM models, where each stage refines information and reduces uncertainty to improve prediction accuracy and stability.

Bringing these two domains together offers the possibility of turning predictive maintenance into an active and self-correcting control process. Concepts from nonlinear optimal control and model-based predictive control closely mirror the goals of RUL estimation, where optimization seeks to minimize energy use, error, or component wear based on the predicted health of the system. In an integrated framework, the RUL predictor acts as a health observer that anticipates future degradation, while the adaptive controller serves as the decision-maker, adjusting parameters such as torque, load, or speed to prevent further damage and extend service life. This interaction creates a closed feedback loop in which prediction and adaptation work together to sustain performance under uncertain conditions. Embedding control-theoretic stability ideas, such as Lyapunov convergence, within RUL modeling can also provide a stronger theoretical foundation, allowing models to be assessed not only for accuracy but also for stability and reliability. Combining predictive modeling with adaptive control therefore points toward the next generation of intelligent maintenance systems—machines that can

monitor their own condition, adapt their operation in real time, and continue functioning safely even in the presence of faults or changing environments.

4 Discussion

Remaining Useful Life (RUL) can be viewed in two ways: as a regression problem where we predict a continuous value, or as a classification problem when we break it down into specific categories. The decision to use one approach over the other often depends on the goals of the study and how researchers decide to define the problem. The The journey of predicting the Remaining Useful Life (RUL) of bearings has transformed significantly, especially with the emergence of deep learning. In the early stages, researchers depended on statistical features and traditional machine learning methods, which often struggled with noise and changing conditions. From 2012 to 2017, techniques like Support Vector Machines (SVM) and Artificial Neural Networks (ANN) began to enhance accuracy by utilizing more complex features, though they still relied heavily on manual feature engineering. Today, deep learning allows us to create models that can learn directly from raw data, resulting in impressive outcomes. However, this advancement also makes it more challenging to understand how these models arrive at their predictions, raising concerns about interpretability. It's essential to establish a stronger theoretical framework by comparing these models with adaptive and robust control techniques like adaptive fuzzy control and robust neural adaptive control that effectively manage uncertainty in systems. Despite these advancements, several challenges remain. Many datasets lack the diversity needed, which limits how well our models can generalize to real-world situations. For instance, while some recent models perform well on benchmark datasets like PHM2012, they often struggle to transfer their effectiveness to real-world applications, where conditions can vary significantly. Future studies should incorporate cross-dataset evaluations to improve generalizability. Developing multimodal datasets that include various types of sensor data is vital for boosting robustness. Reproducibility is a concern due to missing hyperparameters and unclear preprocessing steps, so future studies should report details like model size, training epochs, and optimizer settings. We also need to standardize preprocessing techniques to improve reproducibility in research. Methods such as discrete wavelet transform (DWT) and Hilbert-Huang transform have been effective, but more comparative analyses are needed to determine the best preprocessing methods for different scenarios. Additionally, adopting probabilistic forecasting methods can help provide clearer insights into risks, facilitating better decision-making. Connecting RUL estimation to concepts like adaptive backstepping control and nonlinear optimal control can significantly enhance our predictive maintenance strategies. These approaches can aid in real-time decision-making and improve fault tolerance. For example, if a model predicts a potential failure, adaptive control can adjust operational parameters or schedule maintenance in advance, helping to reduce downtime and ensure system reliability. In practical terms, lightweight models that maintain accuracy are crucial for real-world applications. Techniques like model pruning and quantization can make these models more feasible in industrial environments. Moreover, it's important to address the risks of overfitting on benchmark datasets and to discuss the comparative weaknesses in interpretability or robustness to noise across different architectures. Additionally, we should thoroughly examine the challenges of deploying these models in real-world scenarios. In summary, while we've made great progress in RUL prediction, it's essential to tackle issues like generalization, preprocessing, and practical deployment. Additionally, integrating control strategies and ensuring that our models are interpretable and robust will better position RUL prediction techniques to meet the needs of modern industry.

5 Future direction

Future progress in the field of RUL prediction for bearing depends on addressing these gaps completely. Foremost thing is the development dataset which includes multimodal sensor data with varying conditions. Collaborative efforts between academia and industry could lead to development of open-access datasets that capture variable speeds, loads, lubrication regimes, vibration, temperature and other environmental conditions. Extending the work of studies like [40] [41], which incorporated multiple sensor types, will be essential for building models that generalize beyond vibration-only signals. Such datasets should also be developed to overcome imbalance, ensure sufficient representation of various forms of failure that might occur. Secondly, the research of preprocessing and feature engineering should advance simultaneously. While deep learning reduces dependence on feature engineering, preprocessing is still important for filtering noise and enhancing the signal. Comparative analyses are needed to standardize conventional methods such as WPD and EMD against other alternatives such as denoising autoencoders and so on [37]. Preprocessing techniques implemented should be reported properly in publications, including denoising, normalization, windowing, and feature selection to improve reproducibility and allow fair cross study comparison. Thirdly, research on modeling should advance beyond testing with only one dataset. It's important to test models with different datasets, like training on one benchmark (e.g., PHM2012) and testing on another (e.g., PRONOSTIA or NASA IMS). This should be standard practice to see how well models can work in different operating conditions. Methods such as domain adaptation and transfer learning can bridge the gap between academic research and industry with real world

data. The model should also provide probabilistic forecasts instead of giving exact predictions alone. Methods like Bayesian neural networks, deep ensembles, and quantile regression can provide a wide range of alternatives for predictions of RUL. This helps researchers to compare the replacement cost of the component against the risks of unexpected failures. Risk aware modeling is especially important for applications where safety is critical. Fourthly, interpretability and hybrid modeling require more consideration. Tools such as SHAP (SHapley Additive exPlanations) values, attention mechanisms, and saliency maps can clarify which features or time windows influence the RUL predictions most. Hybrid network architectures that incorporate degradation models within machine learning algorithms (as seen in [30]) offer another way toward greater clarity and estimation ability. These approaches can make prognostics not only accurate but also explainable and trustworthy to researchers. Future research should address the current lack of uncertainty quantification and limited interpretability in bearing RUL prediction. Integrating probabilistic methods such as Monte Carlo dropout, Bayesian inference, or ensemble variance can provide confidence intervals to improve decision reliability, while applying interpretability techniques like SHAP and attention mechanisms—as demonstrated in recent studies can enhance model transparency and tackle the open challenge of feature attribution in time-series data. Finally, Industrial implementation will require lightweight devices capable of operating in real time. To make machine learning models lighter and faster without losing accuracy, we should explore techniques such as knowledge distillation, quantization, and pruning. It's also important to test these models in various real-world conditions, like dealing with noisy signals, sensor failures, or missing data, to confirm they work consistently. When evaluating these models, we should look at how they affect real-world outcomes, like improving reliability, reducing downtime, and saving on maintenance costs, in addition to just technical performance. This approach will help us move these predictive models from the lab to actual industry use, showcasing their real value.

A promising direction is to couple AI-based RUL estimation with adaptive backstepping or nonlinear optimal control frameworks, enabling machines to automatically compensate for degradation effects in real time. Such an integrated architecture would transform traditional predictive maintenance into an adaptive, self-correcting process — improving reliability, safety, and operational efficiency in industrial automation. In summary, the reviewed studies demonstrate notable advances in RUL prediction of bearings, with deep learning and hybrid approaches achieving high accuracy on standard datasets. Nevertheless, the field still faces challenges of generalization, reproducibility, interpretability, and deployment readiness. By investing in developing diverse datasets, advancing preprocessing techniques, adopting probabilistic and understandable models, and prioritizing deployment-focused research, the researchers can transform promising study into reliable, real-world prognostic systems.

6 Conclusion

This review shows how research on predicting RUL of bearings has advanced over the past two decades. The review shows that the research on RUL of Bearing using AI started with the use of conventional machine learning methods that depend on carefully extracted vibration features has now moved toward deep learning and hybrid models capable of extracting features or patterns directly from raw signals itself. This change has brought significant improvements in predictive maintenance plan with early models such as SVMs and multilayer perceptrons generally reached only 85–90% accuracy, while more recent CNN–LSTM hybrids and BiLSTM networks regularly exceed 95%, with some reporting almost error-free performance on benchmark datasets. Approaches based on generative AI models, including GANs and autoencoders, have further improved the classification by handling challenges of noisy data and the shortage of samples from various failure conditions. Nevertheless, progress on open-source datasets does not fully resolve the challenges of applying these algorithms in real world conditions. A substantial dependance on PHM2012, PRONOSTIA, NASA IMS, and FEMTO-ST has created a testing environment that, while useful, does not reflect the complexity of real-world conditions, where load, speed, lubrication, and environmental conditions differ significantly. Many studies miss significant aspects, like the imbalance in datasets and the lack of complete examples of how equipment fails in real world scenario. The inconsistency in reporting how data is processed, along with different ways of measuring success, makes it hard to compare results across studies. There are also challenges with understanding how deep learning models work since they often act like black boxes. Plus, not enough focus is given to how uncertain the predictions are, how efficiently these models can be deployed, and the limitations of using them in real-time conditions. Overall, we see both successes and gaps. On the positive side, modern models show that artificial intelligence can greatly improve how we predict bearing failures. However, for these technologies to be used in the real world, we need to invest in diverse datasets, standardize how we process and evaluate data, create models that are easier to understand, and develop solutions that work reliably in real-time. By tackling these challenges, we can connect promising research with practical applications in industry. This review highlights how far we've come and what needs to be done to ensure that our tools for predicting bearing failures are useful, trustworthy, and scalable in real engineering environments.

Authors' contribution

Rajkumar Palaniappan is the solely responsible for the conception, searching relevant articles, writing and finalization of the manuscript

Acknowledgment

I would like to thank the University of Technology Bahrain management for providing adequate resources and time for preparing this review article.

References

[1] R. Palaniappan, "Predictive manufacturing: State of the art, design standards, limitations, challenges, and future perspectives," 2024, p. 020004. doi: 10.1063/5.0235999.

[2] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, "Machinery health prognostics: A systematic review from data acquisition to RUL prediction," *Mech Syst Signal Process*, vol. 104, pp. 799–834, 2018, doi: https://doi.org/10.1016/j.ymssp.2017.11.016.

[3] I. Nejjar, F. Geissmann, M. Zhao, C. Taal, and O. Fink, "Domain adaptation via alignment of operation profile for Remaining Useful Lifetime prediction," *Reliab Eng Syst Saf*, vol. 242, p. 109718, 2024, doi: https://doi.org/10.1016/j.ress.2023.109718.

[4] Z. Zhao, Bin Liang, X. Wang, and W. Lu, "Remaining useful life prediction of aircraft engine based on degradation pattern learning," *Reliab Eng Syst Saf*, vol. 164, pp. 74–83, 2017, doi: https://doi.org/10.1016/j.ress.2017.02.007.

[5] L. Guo, N. Li, F. Jia, Y. Lei, and J. Lin, "A recurrent neural network based health indicator for remaining useful life prediction of bearings," *Neurocomputing*, vol. 240, pp. 98–109, 2017, doi: https://doi.org/10.1016/j.neucom.2017.02.045.

[6] C. Ferreira and G. Gonçalves, "Remaining Useful Life prediction and challenges: A literature review on the use of Machine Learning Methods," *J Manuf Syst*, vol. 63, pp. 550–562, 2022, doi: https://doi.org/10.1016/j.jmsy.2022.05.010.

[7] L. Zhang, Z. Mu, and C. Sun, "Remaining Useful Life Prediction for Lithium-Ion Batteries Based on Exponential Model and Particle Filter," *IEEE Access*, vol. 6, pp. 17729–17740, 2018, doi: 10.1109/ACCESS.2018.2816684.

[8] Y. Fan, S. Nowaczyk, and T. Rögnvaldsson, "Transfer learning for remaining useful life prediction based on consensus self-organizing models," *Reliab Eng Syst Saf*, vol. 203, p. 107098, Nov. 2020, doi: 10.1016/j.ress.2020.107098.

[9] M. Yan, X. Wang, B. Wang, M. Chang, and I. Muhammad, "Bearing remaining useful life prediction using support vector machine and hybrid degradation tracking model," *ISA Trans*, vol. 98, pp. 471–482, Mar. 2020, doi: 10.1016/j.isatra.2019.08.058.

[10] M. Baptista, E. M. P. Henriques, I. P. de Medeiros, J. P. Malere, C. L. Nascimento, and H. Prendinger, "Remaining useful life estimation in aeronautics: Combining data-driven and Kalman filtering," *Reliab Eng Syst Saf*, vol. 184, pp. 228–239, Apr. 2019, doi: 10.1016/j.ress.2018.01.017.

[11] Z. Chen, Y. Li, T. Xia, and E. Pan, "Hidden Markov model with auto-correlated observations for remaining useful life prediction and optimal maintenance policy," *Reliab Eng Syst Saf*, vol. 184, pp. 123–136, 2019, doi: https://doi.org/10.1016/j.ress.2017.09.002.

[12] M. Elmahallawy, T. Elfouly, A. Alouani, and A. M. Massoud, "A Comprehensive Review of Lithium-Ion Batteries Modeling, and State of Health and Remaining Useful Lifetime Prediction," *IEEE Access*, vol. 10, pp. 119040–119070, 2022, doi: 10.1109/ACCESS.2022.3221137.

[13] M. S. Rathore and S. P. Harsha, "Prognostic Analysis of High-Speed Cylindrical Roller Bearing Using Weibull Distribution and *k* - Nearest Neighbor," *J Nondestruct Eval Diagn Progn Eng Syst*, vol. 5, no. 1, Feb. 2022, doi: 10.1115/1.4051314.

[14] Y. Zhou, M. Huang, and M. Pecht, "Remaining useful life estimation of lithium-ion cells based on k-nearest neighbor regression with differential evolution optimization," *J Clean Prod*, vol. 249, p. 119409, 2020, doi: https://doi.org/10.1016/j.jclepro.2019.119409.

[15] R. and N. S. K. and N. N. M. and V. V. and N. F. G. Abdalla Muhamed Abdelhamed Bakhet and Palaniappan, "An Overview of Industry 5.0: Key Features, Advantages, Limitations and Future Directions," in *Tech Fusion in Business and Society : Harnessing Big Data, IoT, and Sustainability in Business: Volume 1*, R. K. Hamdan, Ed., Cham: Springer Nature Switzerland, 2025, pp. 633–643. doi: 10.1007/978-3-031-84628-1_53.

[16] S. Khan, T. Yairi, S. Tsutsumi, and S. Nakasuka, "A review of physics-based learning for system health management," *Annu Rev Control*, vol. 57, p. 100932, 2024, doi: https://doi.org/10.1016/j.arcontrol.2024.100932.

[17] R. Palaniappan, "Cyber-Physical Systems: Design Standards, Applications, Limitations, Challenges, and Future Perspectives," 2025, pp. 17–30. doi: 10.1007/978-3-031-77078-4_2.

[18] R. Huang, L. Xi, X. Li, C. Richard Liu, H. Qiu, and J. Lee, "Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods," *Mech Syst Signal Process*, vol. 21, no. 1, pp. 193–207, Jan. 2007, doi: 10.1016/j.ymssp.2005.11.008.

[19] S. L. Chen, M. Craig, R. Callan, H. Powrie, and R. Wood, "Use of Artificial Intelligence Methods for Advanced Bearing Health Diagnostics and

Prognostics," in *2008 IEEE Aerospace Conference*, IEEE, Mar. 2008, pp. 1–9. doi: 10.1109/AERO.2008.4526604.

[20] C. Castejón, O. Lara, and J. C. García-Prada, "Automated diagnosis of rolling bearings using MRA and neural networks," *Mech Syst Signal Process*, vol. 24, no. 1, pp. 289–299, Jan. 2010, doi: 10.1016/j.ymssp.2009.06.004.

[21] C. Sun, Z. Zhang, and Z. He, "Research on bearing life prediction based on support vector machine and its application," *J Phys Conf Ser*, vol. 305, p. 012028, Jul. 2011, doi: 10.1088/1742-6596/305/1/012028.

[22] H.-E. Kim, A. C. C. Tan, J. Mathew, and B.-K. Choi, "Bearing fault prognosis based on health state probability estimation," *Expert Syst Appl*, vol. 39, no. 5, pp. 5200–5213, Apr. 2012, doi: 10.1016/j.eswa.2011.11.019.

[23] Z. Tian, "An artificial neural network method for remaining useful life prediction of equipment subject to condition monitoring," *J Intell Manuf*, vol. 23, no. 2, pp. 227–237, Apr. 2012, doi: 10.1007/s10845-009-0356-9.

[24] K. Medjaher, D. A. Tobon-Mejia, and N. Zerhouni, "Remaining Useful Life Estimation of Critical Components With Application to Bearings," *IEEE Trans Reliab*, vol. 61, no. 2, pp. 292–302, Jun. 2012, doi: 10.1109/TR.2012.2194175.

[25] X. Chen, Z. Shen, Z. He, C. Sun, and Z. Liu, "Remaining life prognostics of rolling bearing based on relative features and multivariable support vector machine," *Proc Inst Mech Eng C J Mech Eng Sci*, vol. 227, no. 12, pp. 2849–2860, Dec. 2013, doi: 10.1177/0954406212474395.

[26] A. Soualhi, K. Medjaher, and N. Zerhouni, "Bearing Health Monitoring Based on Hilbert–Huang Transform, Support Vector Machine, and Regression," *IEEE Trans Instrum Meas*, vol. 64, no. 1, pp. 52–62, Jan. 2015, doi: 10.1109/TIM.2014.2330494.

[27] J. Ben Ali, B. Chebel-Morello, L. Saidi, S. Malinowski, and F. Fnaiech, "Accurate bearing remaining useful life prediction based on Weibull distribution and artificial neural network," *Mech Syst Signal Process*, vol. 56–57, pp. 150–172, May 2015, doi: 10.1016/j.ymssp.2014.10.014.

[28] P. Bangalore and L. B. Tjernberg, "An Artificial Neural Network Approach for Early Fault Detection of Gearbox Bearings," *IEEE Trans Smart Grid*, vol. 6, no. 2, pp. 980–987, Mar. 2015, doi: 10.1109/TSG.2014.2386305.

[29] M. Elforjani, "Estimation of Remaining Useful Life of Slow Speed Bearings Using Acoustic Emission Signals," *J Nondestr Eval*, vol. 35, no. 4, p. 62, Dec. 2016, doi: 10.1007/s10921-016-0378-0.

[30] L. Ren, J. Cui, Y. Sun, and X. Cheng, "Multi-bearing remaining useful life collaborative prediction: A deep learning approach," *J Manuf Syst*, vol. 43, pp. 248–256, Apr. 2017, doi: 10.1016/j.jmsy.2017.02.013.

[31] J. Deutsch and D. He, "Using Deep Learning-Based Approach to Predict Remaining Useful Life of Rotating Components," *IEEE Trans Syst Man Cybern Syst*, vol. 48, no. 1, pp. 11–20, Jan. 2018, doi: 10.1109/TSMC.2017.2697842.

[32] L. Ren, Y. Sun, H. Wang, and L. Zhang, "Prediction of Bearing Remaining Useful Life With Deep Convolution Neural Network," *IEEE Access*, vol. 6, pp. 13041–13049, 2018, doi: 10.1109/ACCESS.2018.2804930.

[33] H. O. Omoregbee, M. U. Olanipekun, and A. B. Edward, "Non Coding of Big Dataset and the use of Neural Network Regression Artificial Intelligence Model in Azure for Predicting the Remaining Useful Life (RUL) of Bearing," in *2019 IEEE AFRICON*, IEEE, Sep. 2019, pp. 1–7. doi: 10.1109/AFRICON46755.2019.9133738.

[34] C. Cheng *et al.*, "A Deep Learning-Based Remaining Useful Life Prediction Approach for Bearings," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 3, pp. 1243–1254, Jun. 2020, doi: 10.1109/TMECH.2020.2971503.

[35] B.-L. Lu, Z.-H. Liu, H.-L. Wei, L. Chen, H. Zhang, and X.-H. Li, "A Deep Adversarial Learning Prognostics Model for Remaining Useful Life Prediction of Rolling Bearing," *IEEE Transactions on Artificial Intelligence*, vol. 2, no. 4, pp. 329–340, Aug. 2021, doi: 10.1109/TAI.2021.3097311.

[36] H. Ding, L. Yang, Z. Cheng, and Z. Yang, "A remaining useful life prediction method for bearing based on deep neural networks," *Measurement*, vol. 172, p. 108878, Feb. 2021, doi: 10.1016/j.measurement.2020.108878.

[37] Q. Wang, S. Wang, B. Wei, W. Chen, and Y. Zhang, "Weighted K-NN Classification Method of Bearings Fault Diagnosis With Multi-Dimensional Sensitive Features," *IEEE Access*, vol. 9, pp. 45428–45440, 2021, doi: 10.1109/ACCESS.2021.3066489.

[38] J. A. de Leon, X. Batiller, K. C. Fernandez, M. M. Vaay, C. R. Renosa, and R. Concepcion, "Remaining Useful Life Estimation for Roller Bearings through a Takagi-Sugeno Fuzzy Logic Model with Convolutional Neural Networks Consequents," in *2022 IEEE 14th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, IEEE, Dec. 2022, pp. 1–6. doi: 10.1109/HNICEM57413.2022.10109404.

[39] H. S. Kumar and G. Upadhyaya, "Fault diagnosis of rolling element bearing using continuous wavelet transform and K- nearest neighbour," *Mater Today Proc*, vol. 92, pp. 56–60, 2023, doi: https://doi.org/10.1016/j.matpr.2023.03.618.

[40] M. Motahari-Nezhad and S. M. Jafari, "Comparison of MLP and RBF neural networks for bearing remaining useful life prediction based

on acoustic emission," *Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology*, vol. 237, no. 1, pp. 129–148, Jan. 2023, doi: 10.1177/13506501221106556.

[41] R. Palaniappan, "Comparative Analysis of Support Vector Machine, Random Forest and k-Nearest Neighbor Classifiers for Predicting Remaining Usage Life of Roller Bearings," *Informatica*, vol. 48, no. 7, Apr. 2024, doi: 10.31449/inf.v48i7.5726.

[42] L. Magadán, J. C. Granda, and F. J. Suárez, "Robust prediction of remaining useful lifetime of bearings using deep learning," *Eng Appl Artif Intell*, vol. 130, p. 107690, Apr. 2024, doi: 10.1016/j.engappai.2023.107690.

[43] Y. Ma, "Bearing Life Prediction Model for Electromechanical Equipment by Integrating Deep Neural Network and K-Nearest Neighbor Algorithm and Its Application," *Journal of Theoretical and Applied Mechanics*, vol. 62, no. 4, pp. 721–735, Oct. 2024, doi: 10.15632/jtam-pl/194237.

[44] U. Farooq, M. Ademola, and A. Shaalan, "Comparative Analysis of Machine Learning Models for Predictive Maintenance of Ball Bearing Systems," *Electronics (Basel)*, vol. 13, no. 2, p. 438, Jan. 2024, doi: 10.3390/electronics13020438.

[45] K. Bharatheedasan, T. Maity, L. A. Kumaraswamidhas, and M. Durairaj, "Enhanced fault diagnosis and remaining useful life prediction of rolling bearings using a hybrid multilayer perceptron and LSTM network model," *Alexandria Engineering Journal*, vol. 115, pp. 355–369, Mar. 2025, doi: 10.1016/j.aej.2024.12.007.

[46] V. I. Vlachou *et al.*, "Intelligent Fault Diagnosis of Ball Bearing Induction Motors for Predictive Maintenance Industrial Applications," *Machines*, vol. 13, no. 10, p. 902, Oct. 2025, doi: 10.3390/machines13100902.

[47] Y. Jin, D. Liu, Y. Xiao, and L. Cui, "Dual-channel dynamic spline graph convolutional network for bearing remaining useful life prediction," *Reliab Eng Syst Saf*, vol. 266, p. 111731, Feb. 2026, doi: 10.1016/j.ress.2025.111731.

Hybrid Anomaly Detection in OS Kernel Interfaces for Power Monitoring Systems Using Fuzzy Logic and Deep Belief Networks

Tianyu An[1*], Changlin Lv[1], Manpo Li[1], Shugui Zhang[2]
[1]Northeast Branch of State Grid Corporation, 110180, Liaoning, China
[2]Sichuan Energy Internet Research InstituteTsinghua University, Chengdu 610218, Sichuan, China
E-mail: antianyunesgcc@163.com, lvchanglin@ne.sgcc.com.cn, limanpo@ne.sgcc.com.cn, aix@zhutkj.cn
*Corresponding author

Keywords: Operating system (OS), kernel interfaces, power monitoring systems, anomaly detection, embedded system, security, Data analytics for energy-cost efficient system operation.

Received: June 6, 2025

Anomaly detection in operating system (OS) kernels is critical for the stability and security of embedded systems, particularly in power monitoring applications. OS kernel behavior is complicated, and typical anomaly detection algorithms frequently fail to detect smaller anomalies, especially in power-sensitive applications where energy efficiency is critical. The goal of this research is to create an effective anomaly detection framework capable of reliably identifying abnormalities in the Chinese OS kernel's behavior through power monitoring, assuring consistent system performance and security. The framework includes several critical steps: First, gather a dataset of system call sequences and power usage logs from the OS kernel. Data pre-processing is utilized to clean and normalize the dataset, ensuring it is formatted consistently for investigation. Feature data extraction is then carried out via the Kernel Principal Component Analysis (Kernel PCA) method that uses such important kernel interaction characteristics as the frequency of system calls and the power consumption behaviour. A novel technique, Fire Hawk Optimizer Fused Fuzzy Logic-Based Deep Belief Networks (Fire-Fuzzy DBN) is a hybrid approach that combines FHO to optimize system parameters, Fuzzy Logic to handle uncertainty in system behavior, and DBNs to extract complex patterns, resulting in a robust, adaptive, and effective solution for detecting kernel anomalies. The outcomes reveal that the proposed Fire-Fuzzy DBN strategy, which was implemented in Python, significantly improves kernel anomaly detection accuracy by 99% over previous techniques. The research data analytics for energy-cost efficient system operation establishes the efficacy of fuzzy testing technology in detecting anomalies in OS kernel interfaces for power monitoring systems, therefore improving embedded system dependability and security.

Povzetek: Prispevek obravnava zaznavanje anomalij v jedrnih vmesnikih operacijskega sistema v energijsko občutljivih vgrajenih sistemih za nadzor porabe. Avtorji združijo dnevniške zapise sistemskih klicev in meritve porabe energije, iz katerih s Kernel PCA razvijejo hibridni model Fire-Fuzzy DBN, ki povezuje mehko logiko, DBN ter Fire Hawk Optimizer za optimizacijo parametrov.

1 Introduction

Strong anomaly detection methods are gaining more and more significance to provide cell stability and dependability, as power monitoring systems see an explosive proliferation to the current electronic devices [1]. Hardware interface and the OS kernel are vital when it comes to the movement of data and system performance monitoring on Chinese systems. Anomalies can be generated by complex interactions, and this can prejudice the performance of the system. Such anomalies can be collected to improve the efficiency and reliability of system interface points at the kernel level [2]. Fuzzy testing technology is one of the most common techniques that are adapted to locate vulnerabilities, detect abnormal behaviors of a system. Fuzzy testing is distinct to existing or rather conventional systems of testing since they use known inputs and outputs whereas fuzzy systems test defaults to introduce anomalies through the use of corrupted data sources [3]. Energy-cost efficiency of operation of a system using data analytics would allow the identification of obscure strengths, edge cases that might not be addressed in conventional methods, and would be able to solve anomalies in power usage within a system through fuzzy testing of the Chinese OS kernel interface with a power monitoring system [4]. The operations in kernels are dynamic which can more often than not leave the traditional anomaly detection methods unusable due to

unexpected events that occur within a kernel. In the traditional approaches, sensitivity or freedom of degree was lacking; in terms of energy savings, the way systems or monitoring power is sustained is of essence in systems of large-scale and devices that are embedded [5]. The fuzzy testing technology is more precise and effective due to the anomaly detection which provides random and mutated input data that explores the unusual behaviors, which resolves the historical obstacle of scarce anomaly detection and improves efficiencies of the system by resolving the unanticipated situations [6]. Fuzzy testing helps make the anomaly detecting mechanisms more sensitive by simulating a wide variety of scenarios. The importance of these kinds of systems as that of the Chinese OS kernel interfaces is that when the testing protocol is carried, it does not usually test them against all the anomalies. By identifying minor weak points early enough, fuzzy testing of the system increases reliability and prevents any potential latent faults that may lead to performance or efficiency degradation [7]. The aim of the research would be based on fuzzy testing technologies to develop an anomaly detection system on power monitoring systems. In a bid to enhance the dependability and the power efficiency of the system, the analysis tries to identify and correct the mismatches at OS kernel interface level. It aims to offer this type of a robust solution that will increase the overall reliability and effectiveness of the power monitoring systems.

The motivation behind this work stems from the increasing complexity of OS kernel behavior in embedded power monitoring systems, where even minor anomalies can lead to significant security risks or system failures. Traditional detection methods often overlook subtle deviations, especially under energy constraints. Therefore, developing a more accurate and adaptive anomaly detection framework is essential to ensure system reliability, efficiency, and security. This Fire-Fuzzy DBN addresses a critical gap by introducing a robust solution tailored for power-sensitive, kernel-level anomaly detection. The key contributions of this work are as follows.

Key contribution

- Realistic Kernel Dataset Utilization: The performance of embedded systems in power-sensitive contexts was accurately modeled by using a real-world OS kernel anomaly collection that included power logs and system call traces.
- Strong Pipeline for Preprocessing Data: To guarantee high-quality input for analysis, extensive preparation was used, including data cleaning, management of missing values, and Z-score normalization.
- Effective Kernel PCA Feature Extraction: Through reducing high-dimensional data while maintaining crucial characteristics like call frequency and power consumption, kernel PCA improved the system's sensitivity to minute irregularities.
- Proposed Fire-Fuzzy DBN Model: Developed a hybrid anomaly detection model that handles uncertainty, learns deep features, and improves detection precision by merging Fire Hawk Optimizer, Deep Belief Networks, and Fuzzy Logic.
- Cost-Effective Energy Use: The framework is ideal for power-sensitive and real-time integrated technologies in industrial or utility-based uses since it was designed to identify abnormalities while consuming the least amount of energy possible.

The following sections comprised the remainder of the section: In Part 2, the relevant works are presented. Part 3 covers the material and methods, Part 4 includes the findings and discussion, and Part 5 provides a conclusion.

2 Related work

The core network's attack surface was increased by the 5G network, raising security risks. Intrusion Detection Systems (IDSs) were utilized to identify anomalies to safeguard the network. However, conventional techniques have trouble identifying the relationship between attack behaviors and traffic characteristics [8]. A novel intrusion detection technique based on multiple-kernel clustering (MKC) methods was presented to overcome problems. The approach provided more tolerance for low-quality traffic sampling data, and decreased sensitivity to increase classification accuracy. The growth of Internet of Things (IoT) devices raised security and privacy issues, making anomaly detection essential. Manufacturers frequently refuse to correct vulnerable devices because of vulnerabilities. A useful last line of security was provided by the host-based anomaly detection system for IoT devices (HADES-IoT), which prevented malicious programs from being launched [9]. It is compatible with Linux-based IoT devices and offers impermeable security. Through a memory usage of about 5.5% and minimal Control Process Unit (CPU) resources, HADES-IoT proved to be an efficient method for identifying IoT malware. Although the Internet of Medical Things (IoMT) has transformed healthcare, there remain risks. Anomaly-based Intrusion Detection Systems (AIDS) were among the innovative security measures required to handle. The research used novelty and outlier detection techniques to provide a revised hybrid AIDS design for IoMT networks [10]. The findings of performance evaluations indicate that all integrated machine learning (ML) algorithms have low computational expenditures and that certain algorithms have strong detection performance. The effectiveness of the strategy was illustrated by the details of implementation on devices. Table 1 presents a comparison of existing anomaly detection methods, highlighting their

outcomes, benefits, and relevance to kernel interface monitoring.

Table 1: Comparative overview of advanced anomaly detection techniques relevant to kernel interface monitoring

Reference	Method	Result	Advantage	Limitation
Nalini et al., [11]	GSMPSO-MM-RCNN (Grid Search-based Multi-population PSO with Regional CNN)	Achieved 90% accuracy across four high-dimensional datasets	Reduces overfitting, improves pattern detection, lowers estimation time	Might need high computational resources; lacks external validation.
Wang etal., [12]	Ensemble Learning for System Logs	Improved precision, recall, and F1-score in anomaly prediction	Effective in handling large-scale logs, it enhances prediction accuracy	Be limited to cloud log context; generalizability unclear.
Zacaron et al., [13]	DL-based IDS for SDN	Eliminated 89% of abnormal flows	Detects and blocks threats in real-time, improves SDN network security	Only tested on two benchmark datasets; simplicity might limit scope.
Ahmadi et al., [14]	Super Learning Ensemble for ICS	Achieved high abnormal recall and maximum F1-score	Low error rate, adaptable to smart city/ICS domains	Focused on one-class classification; might not generalize to multiclass scenarios.
Soni et al., [15]	NelPareto AVOA + CDAD (ITAO System)	Detected complex traffic anomalies, improved energy efficiency	Multi-objective optimization enhances traffic flow and safety	Domain-specific, traffic-centric system fuly not generalize outside urban contexts.

3 Materials and methods

The proposed anomaly detection methodology for power monitoring systems involving Chinese OS kernel interfaces proposes novel strategies, such as data preprocessing, Z-score normalization, and Fire-Fuzzy DBN. These strategies improve the ability to identify anomalies under dynamic Chinese OS kernel interfaces with greater accuracy, flexibility, and robustness. The KPCA dimensions are reduced to information, and the Fire-Fuzzy DBN strengthens the robustness of the system. Figure 1 displays the flow of the method used in the proposed system. Figure 1 displays the flow of the method used in the proposed system.

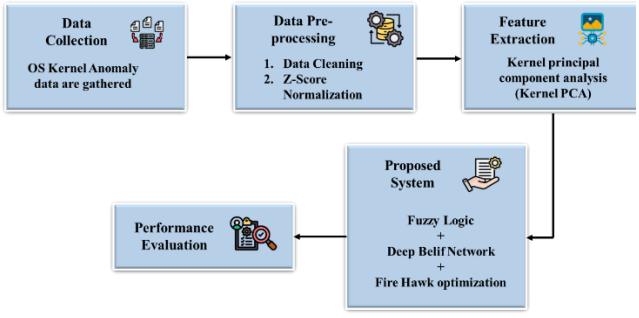


Figure 1: Flow of the recommended technique

3.1 Data collection

The dataset was collected from the open-source Kaggle website https://www.kaggle.com/datasets/ziya07/os-kernel-anomaly-dataset/data. The dataset is created for assessing OS kernel anomaly detection within the context of embedded power monitoring systems. It contains many variables that could be useful for the examination of hybrid models since it represents the relation of power consumption patterns with the increased frequency of system-level events. Data size is 65.77 kB, and a total of 1000 were recorded. Table 2 shows the Dataset Structure and Description.

Table 2: Overview of dataset structure and description

Attribute Name	Description	Data Type	Example Value
timestamp	UNIX timestamp of the system event	Integer	1624982374
process_name	Name of the kernel or user-level process generating the event	String	kernel_thread, pm_monitor

cpu_ usage	CPU utilization at the time of the event (%)	Float	35.72
memory_ usage	Memory usage in MB	Float	412.5
power_ usage	Instantaneous power consumption in Watts	Float	5.61
Syscall _ sequence	Encoded sequence of recent system calls prior to the event	String (encoded)	open, read, mmap, close
event_ type	Type of system event (e.g., syscall, I/O, memory operation)	Categorical	syscall, memory, I/O
anomaly _ label	Class label indicating anomaly presence (0 = normal, 1 = anomaly)	Integer (binary)	0 or 1

3.2.1 Data exploration

Fuzzy testing technology was used to develop an anomaly detection method for the Chinese OS kernel interface in a power monitoring system, focusing on CPU, memory, and power consumption features, enhancing detection accuracy and reliability, particularly in extreme operating conditions. The feature importance is shown in Figure 2.

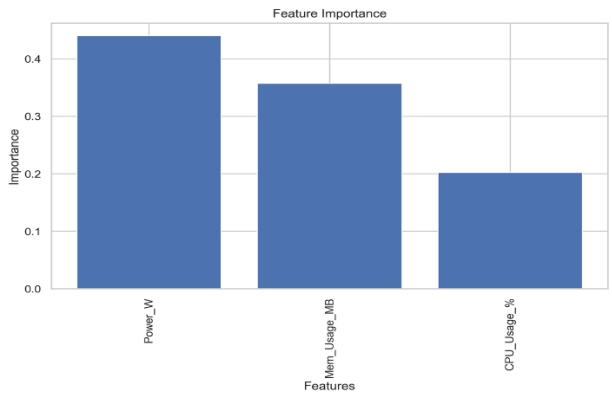


Figure 2: Feature importance scores for CPU, memory, and power consumption in detecting anomalies in the OS kernel power monitoring system.

3.2.2 Data preprocessing

Data preprocessing involved cleaning to remove noise and irrelevant entries, ensuring dataset integrity. Z-score normalization was applied to standardize features, centering them around a mean of zero with unit variance. This choice improved model convergence and highlighted subtle anomalies by equalizing feature scales, which is essential for effective feature extraction and accurate kernel anomaly detection.

Data cleaning: Data cleansing is a very important preprocessing phase in which the errors in data is being identified & correct as well as data become more consistent & more accurate. It is crucial in statistical instrumental control, anomaly detection and energy monitoring systems as anomaly finding and pattern finding is dangerous. Fuzzy testing technology enhances system robustness by allowing flexible approximation reasoning. Integrating data cleansing and anomaly detection improves the reliability and efficiency of power monitoring systems.

3.2.3 Z-score Normalization: Normalization is crucial in the preprocessing stage for power monitoring and anomaly discovery using the Chinese OS kernel. It ensures uniformity across features by transforming information into a standardized format. Z-score normalization is applied for scaling feedback qualities like power consumption readings and system performance metrics. The method allows the fuzzy testing-based anomaly detection system to analyze the data by converting the original values to a normalized form, where the standard deviation equals 1 and the mean value equals 0. Eq 1 presents the formula for Z-score normalization:

$$x' = \frac{x_j - G_j}{std\ (G)} \quad (1)$$

Description: x' = Normalization value results, x = The attribute's value that has to be normalized, G_j = The attribute's mean value, $and\ std\ (G)$ = Attribute G for the standard deviation.

3.3 Kernel principal component analysis (Kernel PCA)

To increase detection accuracy and system responsiveness in the context of the Chinese OS kernel for anomaly detection for kernel interfaces in power monitoring systems, effective data handling is essential. Datasets can be made less dimensional by using the widely utilized KPCA method before being fed to anomaly detection techniques. KPCA performs most effectively in linear systems. The KPCA combines a nonlinear kernel technique with the traditional approach to provide better performance in complicated, nonlinear situations like

kernel interface procedures, and data analytics for energy-cost efficient system operation. The primary components of the updated dataset created by KPCA comprise variables that preserve most of the information from the initial collection of data. By removing components with inadequate data, KPCA lowers the dimensionality of the dataset without significantly sacrificing any important features. The first step is to standardize the initial dataset W_{std}. Eq (2) is used to calculate the primary components that represent the new dataset W_{new}.

$$W_{new} = W_{std}R \tag{2}$$

The standardized original dataset is denoted by W_{std}, and the matrix of eigenvectors sorted by the covariance matrix of W_{std} is R. KPCA expands on this technique by implementing a linear with the higher-dimensional feature space after first mapping the input data to it using nonlinear transformations.

3.4 Proposed system

A hybrid anomaly detection architecture called the Fire-Fuzzy DBN was created to enhance stability and identify issues in Chinese OS kernel interfaces used in power monitoring systems. It blends the FHO, DBNs, and fuzzy logic. By converting input data, such as CPU, memory, and power consumption, into fuzzy values, fuzzy logic controls system behaviour uncertainty. Then, to distinguish between normal and aberrant activity, DBNs learn intricate, hidden patterns. By effectively looking for anomaly indicators and adjusting DBN parameters, FHO improves the model by imitating the fire hawk's hunting technique. In dynamic embedded contexts, this connection facilitates adaptive decisions in real time, enhances detection precision, and encourages energy-effective execution, guaranteeing dependable power management and system safety.

3.4.1 Fuzzy Logic

The initial stage of the anomaly detection algorithm uses fuzzy logic to deal with input data unpredictability, such as fluctuations in CPU activity or power consumption. Using participation processes, it transforms numerical values into degrees of belonging to various categories (such as "low," "medium," or "high") in place of employing physical thresholds. After that, a rule-based system assesses the likelihood that an input may depict anomalous behavior. Even in cases when the input is chaotic or incompatible, this enables the system to make more intelligent and adaptable judgments.

Fuzzy logic is a systematic method for managing inconsistencies and imprecision in Chinese OS kernel interfaces of power monitoring systems. It comprises four components: input fuzzification, fuzzy inference engine, fuzzy rule base, and output defuzzification. The research uses a Mamdani-type fuzzy logic system with a linguistically expressed rule basis. This is how a general fuzzy rule Q_l can be explained as follows by eq (3)

$$Rule\ Q_l: \boldsymbol{IF} w_j\ is\ B_1^l \boldsymbol{AND}\ \dots\ \boldsymbol{AND} w_m\ is\ B_m^l \boldsymbol{THEN} z_l\ is\ A^l \tag{3}$$

The fuzzy sets associated with the j^{th} input and output of the l^{th} rule are represented by A_1^l and A^l, appropriately; m indicates the dimensionality of the input vector w, while z_l is the corresponding output variable. Utilizing a suitable classification function (such as Gaussian, triangular, or trapezoidal), each input w_j is first fuzzified. Fuzzification gives a membership grade $\mu_{A_1^l}(w_j)$ to every input. For rule Q_l, the minimum is used to calculate the degree of disciplining in eq (4).

$$\mu_{Q_l}(\vec{w}) = min\left\{\mu_{A_1^l}(w_j)\right\}, j = 1 \dots, m \tag{4}$$

The output fuzzy set A can be obtained by aggregating the consequents of all terminating rules using an operator, such as the maximum operator. A thorough description of the fuzzy inference procedure was identified. Defuzzification techniques like the centroid technique are used to get an obvious outcome for anomaly detection decision-making. The crisp output z is computed by discretizing the output domain to M samples, as follows in eq (5):

$$y = \frac{\sum_{j=1}^{M} z_j \mu_A(z_j)}{\sum_{j=1}^{M} \mu_B(z_j)} \tag{5}$$

In the dynamic and frequently random nature of kernel interface operations in power monitoring systems, this fuzzy logic framework makes it possible the Chinese OS kernel to identify anomalies more reliably and adaptively, contributing to energy-cost efficient system operation.

3.4.2 Deep Belief Networks (DBN)

The DBN, a type of DL model composed of multiple layers of Restricted Boltzmann Machines (RBMs), receives input data that has been preprocessed using fuzzy logic. By identifying patterns and relationships among various system activities, the DBN learns from this fuzzified data. Through its layer-by-layer learning approach, the DBN is capable of detecting complex and hidden anomalies that might be overlooked by simpler methods.

Precise portrayal of the uncertainty with the input domain is dynamic to facilitate effective anomaly detection in the Chinese OS kernel interface of power monitoring systems about energy-cost efficient system operation. The domain values are converted to fuzzy representations with the

correct membership functions to receive as inputs to the DBN, retaining the system's obscurity using eq (6).

$$\mu_{AD}(W) = \begin{Bmatrix} 0 & (w < b) or (w > c) \\ \frac{w-b}{a-b} & b \le w \le a \\ 1 & b \le w \le d \\ \frac{c-w}{c-d} & d \le w \le c \end{Bmatrix} \quad (6)$$

A DBN is a type of deep neural network (DNN) comprising multiple layers of belief networks, which are generally implemented as RBMs in each level. The sequential stacking of the RBMs in the DBN structure establishes a robust deep structure that can learn hierarchical features. Belief networks and RBMs are the two primary network types that are integrated by DBNs. Layers of random binary units combined by weighted, directed boundaries constitute a belief network. The network can enhance its representation capabilities by learning patterns in the input information by modifying the inter-unit weights according to the belief networks. $\mu_{AD}(W)$ represents the anomaly detection in the investigation; the binary units are assumed to be in 0 or 1.

A DBN is a divided undirected graphical model that incorporates visible and hidden binary units. The only connections between the units are between different categories. The distributions of probabilities across the hidden (Hd) and visible Vs layers are characterized by energy functions, which are expressed as follows in the eqs (7-9):

$$Prob\ (Vs, Hd) = \frac{1}{y} \exp\big(-Eg(Vs, Hd)\big) \quad (7)$$

$$Y = \sum_{Vs,Hd} \exp(-Eg(Vs, Hd)) \quad (8)$$

$$Eg\ (Vs, Hd) = -\sum_j va_j\, Vs_j - \sum_i hb_j\, Hd_j - \sum_j \sum_i Vs_j\, , wt_{j,i}, Hd_i \quad (9)$$

Vs represents the visible elements, (Hd),the implicit components, wt the weight ratio among the present phase and the previous level, hb the hidden biased of the concluding level, and va the visible bias. According to logistic regression, the consequent probability $Prob\ (Vs_i = 1|Hd)$ and $Prob\ (Hd_i = 1|Vs)$ have activating when the unobserved matrix $Hd(Hd_1, \ldots, Hd_i, .., Hd_n)$ is identified. The expected number of the j^{th} visible unit may be determined using the following formula, which is followed by equation (10):

$$Prob\ (Vs_i = 1|Hd) = \sigma\big(va_i + \sum_{i=1}^{m} wt_{j,i}\, Hd_j\big) \quad (10)$$

The stimulation frequency of the i^{th} hidden units may also be determined by using the following expression for a collection of visible vectors $vs\ (vs_1, \ldots, vs_j, \ldots, vs_m)$ followed by (11):

$$Prob\ (Hd_i = 1|Vs) = \sigma\big(hb_i + \sum_{j=1}^{n} wt_{j,i}\, , Vs_j\big) \quad (11)$$

where $wt_{j,i}$ represents the association strength between a j^{th} exposed units and the i^{th} hiding device, hb_i is the hiding perception connection, and σ is the sigmoid activation function. Maximizing the joint probabilities among the set of training inputs is essential for anomaly detection for a power monitoring system's kernel interface to properly learn the typical operational behavior using eq (12).

$$\arg max_{wt} \prod_{vs \in Vs} Prob(Vs) \quad (12)$$

Where the set of all training input samples obtained from the kernel interface data is represented by Vs. Using fuzzy testing technology for energy-cost efficient system operation, the system can efficiently differentiate between typical and unusual behavior patterns by maximizing this probability, improving the security and dependability of power monitoring activities.

3.4.3 Fire Hawk Optimizer (FHO)

The FHO improves the DBN's performance by adjusting its settings automatically. Inspired by how fire hawks search for prey, this method searches for the best values for learning, such as weights and biases, so the DBN can detect anomalies more accurately. It tries different combinations, compares their results, and moves toward the best solution. This optimization makes the system faster and more accurate in finding kernel-level issues.

The FHO algorithm was inspired by fire hawks' foraging habits, specifically the ability to initiate and spread flames. The FHO method initializes a set of candidate solutions (W) that reflect possible practical conditions of the system to discover anomalies in the Chinese OS kernel interface of a power monitoring system. To efficiently investigate a variety of patterns of typical and anomalous system behavior, the starting positions of the fire hawks and prey, represented as position vectors inside the search space, are allocated at random using eqs (13-14).

$$W = \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_j \\ \vdots \\ W_M \end{bmatrix} = \begin{bmatrix} W_1^1 W_1^2 & \cdots & W_1^i & \cdots & W_1^c \\ W_2^1 W_2^2 & \cdots & W_2^i & \cdots & W_2^c \\ \vdots \quad \vdots & \cdots & \vdots & \ddots & \vdots \\ W_j^1 W_j^2 & \cdots & W_j^i & \ddots & W_j^c \\ \vdots \quad \vdots & \cdots & \vdots & \cdots & \vdots \\ W_M^1 W_M^2 & & W_M^i & & W_M^c \end{bmatrix} \begin{cases} j == 1,2, \ldots, M. \\ i = 1,2, \ldots, c. \end{cases} \quad (13)$$

$$x_i^j(0) = x_{i,min}^j + rand.(x_{i,max}^j - x_{i,min}^j)\begin{cases} j == 1,2,\dots,M. \\ i = 1,2,\dots,c. \end{cases} \quad (14)$$

Here M is the overall number of solution individuals that represent the various Chinese OS kernel interface operating asserts in the power monitoring system; W_j is the j^{th} solution candidate in the search space; c is the dimensionality of the anomaly detection problem; $x_i^j(0)$ is the initial position of the i^{th} decision variable for the j^{th} solution candidate; W_j^i is the i^{th} decision variable linked to the j^{th} solution prospective; and the evenly distributed random number $rand$ is in the range $(0, 1)$; the minimum and maximum bounds for the i^{th} decision variable of the j^{th} candidate solution are represented by $x_{i,min}^j$ and $x_{i,max}^j$, respectively, guaranteeing the system's practicable operating range for Chinese OS kernel for anomaly detection, respectively.

Whereas PR_l indicates the l^{th} potential anomaly found in the search space out of a total of n potential anomalies, FH_k indicates the k^{th} candidate anomaly detector (fire hawk) with the entire search space of m fire hawk recruits. Throughout the subsequent portion of the detection method, the distance between each fire hawk (anomaly detector) and its matching prey (possible anomaly) is determined. The distance, D_k^l, is described by the following equation. Where (w_2, w_1) and (z_2, z_1) indicate the coordinates of the FHO in the search space, respectively; D_k^l indicates the total distance between the k^{th} fire hawk and the l^{th}; n and m indicate the aggregate number of FHO in the search space, followed by eqs (15-17).

$$PR = \begin{bmatrix} PR_1 \\ PR_2 \\ \vdots \\ PR_l \\ \vdots \\ PR_n \end{bmatrix}, \quad l = 1,2,\dots,n. \quad (15)$$

$$FH = \begin{bmatrix} FH_1 \\ FH_2 \\ \vdots \\ FH_k \\ \vdots \\ FH_m \end{bmatrix}, \quad k = 1,2,\dots,m \quad (16)$$

$$D_k^l = \sqrt{(w_2 - w_1)^2 + (z_2 - z_1)^2}, \quad \begin{cases} k = 1,2,\dots,m \\ l = 1,2,\dots,n \end{cases} \quad (17)$$

Where GB represent for the greatest anomaly identification solution found in the search space, which is regarded as the most important anomaly indicator; q_1 and q_2 are uniformly distributed random numbers in the range $(0, 1)$ that are used to simulate the movement of anomaly detection devices in the direction of primary anomaly and the surrounding detection categories; and FH_{Near} demonstrates a neighboring anomaly detector in the search space using eq (18).

$$FH_k^{new} = FH_k + (q_1 \times GB - q_2 \times FH_{Near}), k = 1,2,\dots,m. \quad (18)$$

The following is the mathematical expression for SP_k and SP, implementing to account that in the context Chinese OS kernel for anomaly detection, an optimal location is an ideal area where several kernel interface behaviors cluster to maintain system stability through anomalous incidents, data analytics for energy-cost efficient system operation. The l^{th} detected anomaly occurrence in the kernel interface monitoring area is indicated by PR_l, while PR_r denotes the r^{th} anomaly-related behavior reported associated with the l^{th} Chinese OS kernel for anomaly detector, followed by eqs (19-20).

$$SP_k = \frac{\sum_{r=1}^{q} PR_r}{q}, \begin{cases} r = 1,2,\dots,q. \\ k = 1,2,\dots,m. \end{cases} \quad (19)$$

$$SP = \frac{\sum_{l=1}^{n} PR_l}{n}, \quad l = 1,2,\dots,m. \quad (20)$$

The FHO successfully detects anomalies in the Chinese OS kernel interface using the data analytics for energy-cost efficient system operation and foraging behaviors of fire hawks to search for possible vulnerabilities of the system. The Fire-Fuzzy DBN enhances the power monitoring system's Chinese OS kernel interface, abnormality detection, and stability by integrating hierarchical learning from DBN, Fuzzy Logic uncertainty management, and the adaptive anomaly detection feature of FHO, thereby maximizing fire hawk distance movement. By a novel blending of techniques, this hybrid method efficiently identifies faults and optimizes system stability.

4 Experimental evaluation and discussion

Effectiveness assessment and the outcomes of the algorithm's adoption are discussed in the next part. The efficacy of the suggested Fire-Fuzzy DBN is demonstrated by the accuracy outcomes of the study's compared to the conventional and Extended Berkeley Packet Filter Intrusion Detection/Prevention Systems (iKern-IDPS) [16]. Table 3 summarizes the key components and hyperparameter configurations used in the proposed Fire-Fuzzy DBN anomaly detection framework.

Table 3: Hyperparameter configuration of Fire-Fuzzy DBN Framework

Component	Hyperparameter	Value / Description
Fuzzy Logic	Number of Input Variables	3 (CPU, Memory, Power Usage)
	Membership Function Type	Triangular
	Membership Levels per Input	3 (Low, Medium, High)
	Total Fuzzy Rules	9 (3×3 combinations)
	Defuzzification Method	Centroid method
	Rule Evaluation Method	Mamdani Inference System
Kernel PCA	Number of Components	5 (retaining 95% cumulative variance)
	Kernel Type	Radial Basis Function (RBF)
DBN (Deep Belief Network)	Input Layer Size	10 (from fuzzified + reduced features)
	Hidden Layer 1 Size	64 neurons
	Hidden Layer 2 Size	32 neurons
	Output Layer Size	2 (Normal, Anomaly)
	RBM Pretraining Epochs	10
	Learning Rate (RBM)	0.01
	Fine-tuning Optimizer	Adam
	Fine-tuning Epochs	20
	Batch Size	128
Fire Hawk Optimizer (FHO)	Population Size (N)	30 fire hawks
	Maximum Iterations	50
	Search Space Range	[-1, 1] (for weight initialization)
	Distance Metric	Euclidean distance
	Randomization Factors	Uniform [0, 1] for search variation
	Fitness Function	DBN validation accuracy
Training Dataset	Dataset Size	45,672 records
	Anomaly Ratio	~9% anomalies
	Train/Test Split	80% / 20%
	Normalization Method	Z-score

4.1 Experimental configuration

Python installations and system settings are used for Power monitoring systems that use fuzzy testing technologies to detect anomalies in the OS kernel interface. The experimental environmental configuration is shown in Table 4.

Table 4: Implementation environment details

Components	Descriptions
RAM	25 GB
CPU	Intel Xeon Processor (2.4 GHz)
GPU	NVIDIA Tesla V100
OS	Ubuntu 20.04 LTS
System Programming	Python 3
DL Framework	PyTorch 1.8

4.2 Correlation and confusion matrix

The correlation matrix assesses the covariance between power consumption, CPU usage, and memory usage. The correlation matrix indicates a low correlation between the power and memory usage. The confusion matrix analyzes the accuracy of an anomaly detection technique. It shows that the classification accuracy was better, as there were only a few misclassifications in the power monitoring system.

Correlation matrix: Fuzzy testing method is employed in the investigation to examine anomaly detection in the power monitoring system's Chinese OS kernel interface. Power consumption, CPU usage, and memory usage relations are shown on the correlation heatmap. Power and memory usage can't be closely related, evidenced by the lack of a strong association between memory usage and power consumption and CPU usage. The analysis optimizes power management and detects anomalies in the system.

Confusion matrix: The performance of the operational efficiency of the anomaly detection method using the Chinese OS kernel interface for power monitoring systems is illustrated in this confusion matrix shown in Figure 3. The model correctly labeled 180 normal instances and 8 anomaly cases, incorrectly classified 10 anomalies as normal and 2 normal instances as anomalies. The fuzzy testing technology utilized for anomaly detection is efficient in distinguishing between normal and anomalous behavior within the system, a fact that is proved by its relatively high accuracy and low rate of incorrect classification.

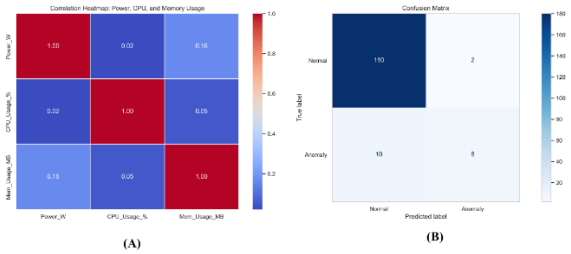


Figure 3: (A) Correlation matrix, (B) Confusion matrix evaluation

4.3 Process and label distribution for anomaly detection in power monitoring system

The procedure and label distribution for the anomaly detection task for a Chinese OS kernel interface in a power monitoring system are shown in this Figure 4. The system process distribution is dominated by kernel_ thread and pm_ monitor, but there's a significant label imbalance, with most cases classified as normal and a few anomalous, which could hinder the framework's training and classification of anomalies.

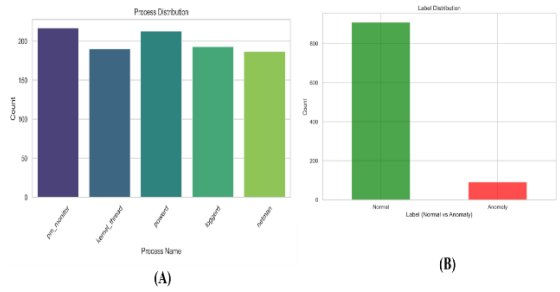


Figure 4: (A) Process distribution, (B) Label distribution outcome performance

4.4 Fuzzy testing-based anomaly detection in Chinese OS kernel interfaces for energy monitoring

The anomaly detection framework employs fuzzy testing technologies to attack Chinese OS kernel interfaces in power monitoring systems (Figure 5). Fuzzy testing evaluates system strength and vulnerabilities by detecting anomalies in CPU and power consumption. It enables early failure or inefficiency detection through pattern analysis, enhancing the protection and stability of energy-monitoring systems by detecting kernel-level interaction inefficiencies.

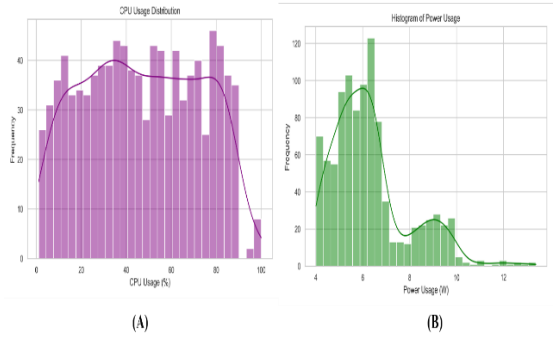


Figure 5: (A) CPU usage distribution, (B) Histogram of power usage of outcome performance

4.5 Time-series analysis of kernel power in fuzzy testing for anomaly detection

To detect anomalies in the power consumption patterns of Chinese OS kernel interfaces, the time series of employed fuzzy testing is shown in Figure 6. The system monitors the kernel's stress response using unplanned or warped inputs. Variations in power consumption are analyzed, revealing anomalous spikes and erratic patterns. Rolling mean analysis identifies deviations from normal usage patterns, enhancing the reliability and robustness of the power monitoring system.

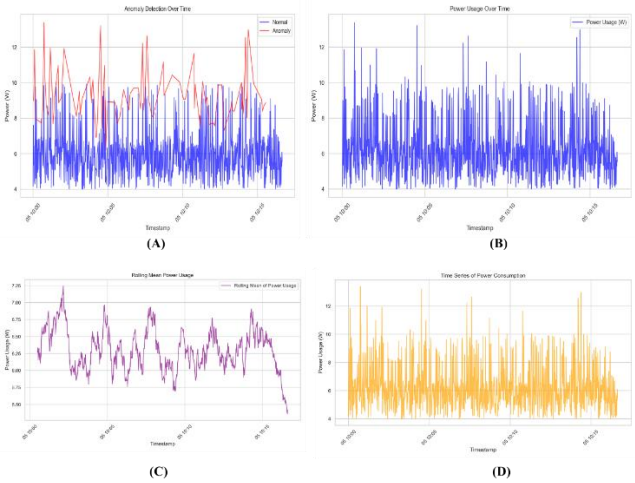


Figure 6: (A) Anomaly detection over time, (B) Power usage over time, (C) Rolling mean power usage, and (D) Time series of power consumption

4.6 Multivariate anomaly detection in kernel-level power monitoring using Fuzzy testing

The fuzzy testing-based anomaly detection method at Chinese OS kernel interfaces under power management scenarios is displayed in Figure 7. The technique uses fuzzy testing to identify unexpected behavior in power consumption and system parameters like CPU, memory usage, and time. It uses multivariate analysis to detect concealed defects and abnormal system responses, improving fault tolerance, system stability, and operating efficiency in real-time power monitoring systems.

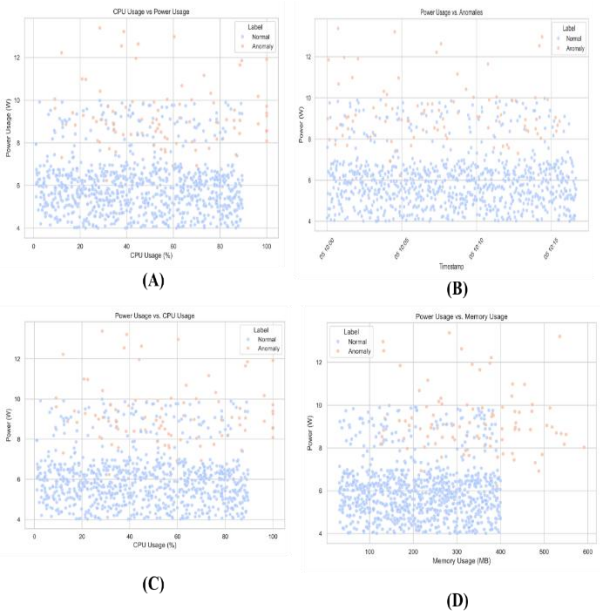


Figure 7: (A) CPU usage vs power usage, (B) Power usage vs anomalies, (C) Power usage vs CPU usage, (D) Power usage vs memory usage

4.7 Performance of Fuzzy testing-based anomaly detection using ROC analysis

A fuzzy testing-based approach for anomaly detection from Chinese OS kernel interfaces in a power monitoring system is illustrated through the use of a Receiver Operating Characteristic (ROC) curve, as shown in Figure 8. The model effectively distinguishes normal and abnormal behavior, with high sensitivity and specificity, and excellent classification performance, achieving an Area Under Curve (AUC) of 0.94, proving the effectiveness of fuzzy testing in early anomaly detection and system security enhancement.

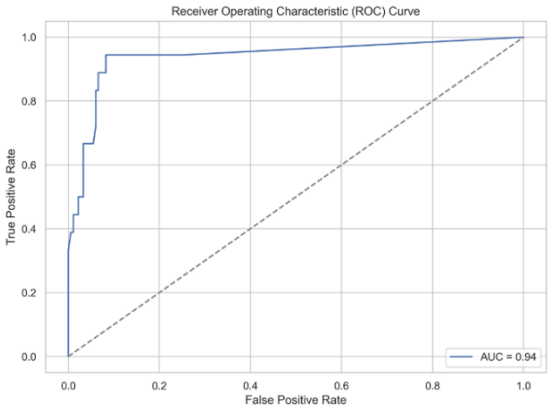


Figure 8: Evaluation performance of ROC curve

4.8 Comparative evaluation for accuracy

Accuracy in anomaly detection refers to how accurately a system detects outliers or behaves unusually in the dataset provided. The Chinese OS kernel interface of the OS fundamental structure is often tailored for certain Chinese hardware or software environments for data analytics for energy-cost efficient system operation. Fuzzy testing technology is testing systems with approximations by introducing variables and testing the system under multiple conditions to ensure increased robustness and fault tolerance. The existing technique, iKern-IDPS scored 98.84%, and the proposed Fire-Fuzzy DBN method improves accuracy to 99%. The outcomes demonstrate that the recommended strategy has the highest accuracy when compared to the current techniques. Table 5 and Figure 9 show the outcome performance of the existing and proposed techniques for accuracy.

Table 5: Comparative performance of accuracy

Methods	Accuracy (%)
iKern-IDPS [16]	98.84
Fire-Fuzzy DBN [Proposed]	99

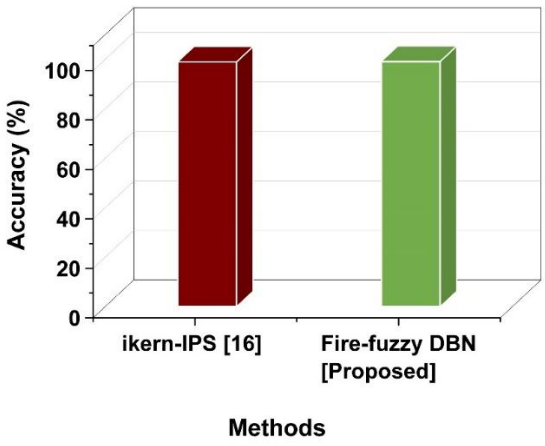


Figure 9: Outcome performance of accuracy

The method would face scalability concerns if applied to vast datasets, which is an area of research limitation. In addition to that, the effectiveness of the model under scenarios such as these with missing logs would be subject to its assumption regarding log quality data [12]. The research could investigate techniques to deal with data as making it more scalable. The drawback is that in dynamic, volatile network situations where attack signatures change continuously, the proposed IDS cannot perform as well. Using benchmark datasets adequately captures the diversity of vectors and the complexity present in the data [13]. Such issues should be addressed by the assignment with diverse real-world data and more accommodating detection mechanisms.

The challenges of this research pertain to the fact that the iKern-IDPS cannot incorporate that may or could be a real attack paradigm, as it is more oriented around anomalies at the kernel interface. Also, the ability of the fuzzy testing technology to be configurable to dynamic and evolving OS environments remains untested over a timeline of exploit activity. The limitation of the Kernel-level trac was that it relied exclusively on simulated test environments, which cannot accurately model real-world conditions. The method cannot capture all potential kernel-level anomalies, which can limit detection capabilities. To address these challenges, the Fire-Fuzzy DBN approach could incorporate the detection of dynamic attacks while adapting the current OS environment through continual learning. Incorporating combined simulation and scenarios could broaden the coverage of anomalous detection, along with continual improvements in accurate detection over time.

5 Conclusion

The anomalous behavior of the system is to determine the purpose of anomaly detection for Chinese OS kernel interfaces for Power Monitoring Systems. Fuzzy testing

technology is used to discover and detect anomalies resulting from the use of fuzziness or imprecise logic. The method helps ensure consistent operation, energy-cost efficient system operation and energy utilization monitoring in the system. The Fire-Fuzzy DBN method achieved the greatest performance and was superior in terms of performance in accuracy (99%). Fuzzy testing for anomaly detection is ultimately constrained by its limitations in determining all of the possibilities available to the system and its applicability being limited to a few platforms.

Limitation and future scope

While the proposed Fire-Fuzzy DBN framework demonstrates high accuracy, it has limitations. The integration of fuzzy logic and deep networks can lead to increased computational overhead, potentially impacting real-time performance. Additionally, scalability in large-scale or diverse kernel environments remains a concern, necessitating further optimization and testing to ensure robustness in practical, real-world deployments. To provide adaptive, self-learning anomaly detection in dynamic power monitoring systems, further improvements may include integrating reinforcement learning, extending detection capabilities across more OS components, and optimizing the model for real-time deployment.

References

[1] Wu, Y., Wang, S., Chen, H., Peng, D., & Yuan, Z. (2024). Kernelized fuzzy-rough anomaly detection. *IEEE Transactions on Fuzzy Systems*, *32*(8), 4285-4296.https://doi.org/10.1109/TFUZZ.2024.3393710

[2] Duan, G., Fu, Y., Cai, M., Chen, H., & Sun, J. (2023). DongTing: A large-scale dataset for anomaly detection of the Linux kernel. *Journal of Systems and Software*, *203*, 111745.. https://doi.org/10.1016/j.jss.2023.111745

[3] Zhang, H., Liu, Y., Gui, F., & Yang, X. (2023). A universal aquaculture environmental anomaly monitoring system. *Sustainability*, *15*(7), 5678.https://doi.org/10.3390/su15075678

[4] Shi, Y. (2025). FAT-Net: A Spectral-Attention Transformer Network for Industrial Audio Anomaly Detection Using MFCC and Raw Features. *Informatica*, *49*(26).https://doi.org/10.31449/inf.v49i26.8746

[5] Wang, R., Qiu, H., Cheng, X., & Liu, X. (2023). Anomaly detection with a container-based stream processing framework for industrial internet of things. *Journal of Industrial Information Integration*, *35*, 100507.https://doi.org/10.1016/j.jii.2023.100507

[6] Liu, J., Guogang, W. A. N. G., Zong, X., Bowei, N. I. N. G., & He, K. (2025). EfficientTransformer: A Dynamic Anomaly Detection Model for Industrial Control Networks. *IEEE Access*.https://doi.org/10.1109/ACCESS.2025.3545659

[7] Halimi, K., Hadjadj, A., Kouahla, Z., & Farou, B. (2025). A Fuzzy Logic-Driven Semantic and Binary Tree-Based Indexing Framework for Scalable IoT Data Storage and Retrieval. *Informatica*, *49*(24).https://doi.org/10.31449/inf.v49i24.8039

[8] Hu, N., Tian, Z., Lu, H., Du, X., & Guizani, M. (2021). A multiple-kernel clustering based intrusion detection scheme for 5G and IoT networks. *International Journal of Machine Learning and Cybernetics*, *12*(11), 3129-3144. https://doi.org/10.1007/s13042-020-01253-w

[9] Breitenbacher, D., Homoliak, I., Aung, Y. L., Elovici, Y., & Tippenhauer, N. O. (2021). HADES-IoT: A practical and effective host-based anomaly detection system for IoT devices (extended version). *IEEE Internet of Things Journal*, *9*(12), 9640-9658. https://doi.org/10.1109/JIOT.2021.3135789

[10] Zachos, G., Mantas, G., Porfyrakis, K., Bastos, J. M. C., & Rodriguez, J. (2025). Anomaly-Based Intrusion Detection for IoMT Networks: Design, Implementation, Dataset Generation and ML Algorithms Evaluation. *IEEE Access*.. https://doi.org/10.1109/ACCESS.2025.3547572

[11] Nalini, M., Yamini, B., Fernandez, F. M. H., & Priyadarsini, P. U. (2024). Enhancing anomaly detection Efficiency: Introducing grid searchbased multi-population particle Swarm optimization algorithm based optimized Regional based Convolutional neural network for robust and scalable solutions in High-Dimensional data. *Biomedical Signal Processing and Control*, *96*, 106651.https://doi.org/10.1016/j.bspc.2024.106651

[12] Wang, B., Hua, Q., Zhang, H., Tan, X., Nan, Y., Chen, R., & Shu, X. (2022). Research on anomaly detection and real-time reliability evaluation with the log of cloud platform. *Alexandria Engineering Journal*, *61*(9), 7183-7193.https://doi.org/10.1016/j.aej.2021.12.061

[13] Zacaron, A. M., Lent, D. M. B., da Silva Ruffo, V. G., Carvalho, L. F., & Proença Jr, M. L. (2024). Generative adversarial network models for anomaly detection in software-defined networks. *Journal of Network and Systems Management*, *32*(4), 93. https://doi.org/10.1007/s10922-024-09867-z

[14] Ahmadi-Assalemi, G., Al-Khateeb, H., Epiphaniou, G., & Aggoun, A. (2022). Super learner ensemble for anomaly detection and cyber-risk quantification in industrial control systems. *IEEE Internet of Things Journal*, *9*(15), 13279-13297.https://doi.org/10.1109/JIOT.2022.3144127

[15] Soni, R., Soni, S., & Nagwanshi, K. K. (2025). Efficient cluster-based deep anomaly detection based traffic analysis and multi-objective optimization for smarter traffic control. *Evolving Systems*, *16*(1), 5.https://doi.org/10.1007/s12530-024-09636-y

[16] Hadi, H. J., Adnan, M., Cao, Y., Hussain, F. B., Ahmad, N., Alshara, M. A., & Javed, Y. (2024). ikern: Advanced intrusion detection and prevention at the kernel level using ebpf. *Technologies*, *12*(8), 122.. https://doi.org/10.3390/technologies12080122

AI-Driven Building Energy Prediction and Multi-Objective Scheduling Using LSTM-GRU-MPC Integration

Yaqiong Dong*, Wei Guo
Henan University of Urban Construction, Pingdingshan 467036, China
E-mail: dongyaqiong@163.com, guoweivip@outlook.com
*Corresponding author

Keywords: artificial intelligence, building energy management, optimize scheduling, deep learning model

Received: August 5, 2025

With the increasing energy consumption of buildings, traditional energy management methods are challenging to cope with complex real-time demand and supply changes. Artificial intelligence-based prediction and optimal scheduling methods are urgently needed to improve energy efficiency and reduce energy consumption. The primary goal is to enhance the energy efficiency of buildings through intelligent prediction and scheduling techniques. To achieve this, we propose the use of Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models, which are well-suited for capturing the temporal dependencies inherent in energy consumption data. Secondly, in the aspect of energy optimal scheduling, a multi-objective optimal scheduling method is designed by combining model predictive control (MPC) and artificial intelligence technology, and the strategy is adjusted and optimized according to dynamic constraints. This paper integrates big data and Internet of Things technology into the building energy management system to further improve management efficiency. It uses the collaborative optimization of edge computing and cloud computing to achieve the combination of real-time scheduling and long-term planning. The temperature, humidity, and energy consumption data show a certain regularity in this building energy consumption prediction analysis. For example, the average energy consumption within the building is 23.7 kWh, and the maximum energy consumption value is 91.2 kWh, showing an apparent fluctuation in the building's energy demand. LSTM achieved a Mean Absolute Error (MAE) of 12.3 kWh, outperforming linear regression (MAE = 25.6 kWh) and GRU (MAE = 13.8 kWh) in long-term temporal dependency capture. The MPC-based scheduling reduced peak energy consumption by 13.5% (from 91.2 kWh to 78.9 kWh) and total energy cost by 18%, while maintaining an 85% thermal comfort satisfaction rate.

Povzetek: Predlagan je sistem za pametno upravljanje energije v stavbah, kjer LSTM/GRU napovedujeta porabo, MPC+AI pa optimizira razporejanje za manjše konice in stroške ob ohranjanju udobja.

1 Introduction

With the rapid development of artificial intelligence technology, primarily the breakthrough of deep learning, reinforcement learning, and intelligent optimization algorithms, building energy management has ushered in unprecedented opportunities [1, 2]. The core strength of these technologies is their ability to handle complex nonlinear relationships and quickly adapt to dynamically changing environments. Deep learning technology can train complex models based on large-scale historical data, thus achieving high-precision energy consumption prediction [3]. Specifically, deep learning models (e.g., LSTM, GRU) enable high-precision energy prediction by capturing temporal dependencies in historical data, while reinforcement learning and MPC provide adaptive real-time optimization for energy dispatching under dynamic constraints [4]. Integrating these AI tools addresses traditional methods' limitations in handling complex, time-varying building energy systems [5]. Energy consumption in buildings is influenced by various factors, such as weather conditions, occupancy patterns, and operational schedules, all of which exhibit complex temporal relationships. Traditional machine learning models, such as linear regression or decision tree-based models, often struggle to capture these time-varying patterns effectively. In contrast, LSTM and GRU, both types of recurrent neural networks (RNNs), are specifically designed to handle sequential data with long-term dependencies. These models excel at learning temporal patterns, making them ideal candidates for energy consumption prediction tasks. The LSTM network, in particular, is known for its ability to retain information over long sequences, which is crucial when dealing with energy consumption data that spans days, weeks, or even months. GRU, a simplified variant of LSTM, also provides strong performance with fewer parameters, making it computationally efficient while still capturing the essential temporal dependencies [6, 7].

Distributed intelligent sensors (deployed in offices, lobbies, and HVAC rooms) collect hourly data on key variables: indoor temperature, humidity, occupancy (via motion detectors), HVAC operation status, and real-time energy consumption. These sensors transmit data wirelessly via LoRaWAN (Long Range Wide Area Network) to edge computing nodes [8].

In terms of zero-energy buildings, researchers have made many explorations. The energy, environmental, and economic aspects of the whole life cycle of zero-energy solar housing are deeply evaluated, and the critical areas of industrialization development of zero-energy housing are analyzed [9]. To assess the prediction accuracy of our models, we use the Mean Absolute Error (MAE) as a performance metric. MAE is a widely used metric in regression tasks, as it provides an intuitive measure of prediction accuracy by calculating the average of absolute errors between predicted and actual energy consumption values. Recent lightweight AI models—such as compressed GRU variants based on TinyML (Tiny Machine Learning)—further boost edge-deployed prediction performance by reducing parameters without significant accuracy loss, making them ideal for resource-constrained building scenarios [10, 11]. Zhu and other scholars analyzed the energy performance of thermal collection and storage wall technology in zero-energy buildings. Can Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models improve the hourly prediction accuracy of building energy consumption compared to traditional regression-based methods (e.g., linear regression) [12, 13]. Deng and other scholars compared the energy supply concepts of zero-energy housing in two typical climate zones: the humid climate zone represented by Shanghai and the dry climate zone represented by Madrid. By analyzing the annual energy balance, the optimal design strategy of the HVAC system based on different climate zones is proposed [14, 15]. Energy management studies across diverse climate and building types highlight both unique strengths and universal limitations of current methods. Ahrarinouri et al. effectively reduced energy costs in residential settings using multiagent reinforcement learning, though long-term predictions and IoT integration remain challenges. Arbab et al. achieved static load forecasting accuracy with artificial neural networks but struggled with dynamic temporal dependencies. Cantemir et al. stabilized energy use in commercial buildings via static MPC but lacked real-time grid adaptability. Meanwhile, de-Paz-Centeno et al. developed lightweight ML for PV data integration, focusing narrowly on one renewable source. Overall, these approaches reveal gaps in adaptability, multi-objective optimization, and unified frameworks for real-time integration [16, 17]. Building energy consumption prediction and optimal scheduling methods based on artificial intelligence can significantly improve the intelligent level of building energy management and promote the development of green buildings such as zero-energy buildings. An in-depth exploration of this research direction will support achieving sustainable development goals and building smart cities [18, 19]. The aim of this study is to propose an AI-driven framework integrating LSTM, GRU, and MPC for improving building energy efficiency through accurate prediction and dynamic scheduling. This paper is structured as follows: Section 2 introduces deep learning models for energy prediction, Section 3 details the multi-objective optimization method, Section 4 integrates IoT and big data, Section 5 presents experimental results, and Section 6 concludes the study. Based on perceived states, the RL agent adjusts MPC control parameters (e.g., HVAC fan speed, cooling/heating setpoints) to minimize energy consumption while avoiding thermal comfort violations. For example, if indoor temperature exceeds 26°C, the agent increases MPC's cooling priority to restore comfort.

Table 1 is related works summary table.

2 Deep learning prediction model of building energy consumption

2.1 Architecture optimization of deep neural network based on time series data

The selection of activation function is of great significance in DNN optimization. As shown in equations (1) and (2), h is the output of the current layer, W is the weight, x is the output of the previous layer, b is the bias term, and f is the activation function. y is the output after activation. The nonlinear characteristics of ReLU can help the network effectively extract complex features, and its computational efficiency is high, which can significantly improve the training speed of the model.

Table 1: Related works summary

Method/Model	Dataset Used	Metrics
Multiagent Reinforcement Learning	Residential building energy consumption data	Energy cost reduction rate
Artificial Neural Networks (ANN)	Indoor illuminance data of buildings	Static load forecasting accuracy
Static Model Predictive Control (MPC)	Commercial building energy use data	Energy use stability index
Lightweight Machine Learning (ML)	Photovoltaic (PV) production data	PV data imputation accuracy
LSTM	Smart home heating-cooling and power consumption data	Not specified in detail

Table 2: Performance comparison of LSTM, GRU, Linear regression, and DNN on building energy consumption prediction

Model	MAE (kWh)	RMSE (kWh)	MAPE (%)	Training Time per Epoch (Minutes)
Linear Regression	25.6	29.5	18.2	0.8
DNN	13.2	16.8	9.0	3.0
GRU	13.8	17.2	9.5	3.5
LSTM (Proposed)	12.3	15.7	8.6	6.8

$$y = max(0, x) \tag{1}$$

$$h = f(W \cdot x + b) \tag{2}$$

Aiming at the high-dimensional characteristics of building energy consumption data, DNN gradually extracts low-dimensional to high-dimensional feature representations by stacking multiple hidden layers, so as to achieve accurate modeling of building energy consumption. As shown in equation (3), e is the error of the current layer, e' is the error of the next layer, W' is the weight of the next layer, and $f'(h)$ is the derivative of the activation function. The training process of DNN needs to be cooperated with efficient optimization algorithms, such as the Adam optimizer, which can quickly find the optimal solution in the multi-dimensional parameter space while avoiding falling into the local optimum, thus effectively reducing the risk of overfitting of the model.

$$e = e' \cdot W' \cdot f'(h) \tag{3}$$

In the specific application of building energy consumption prediction and optimal scheduling, the whole life cycle cost optimization of zero emission buildings is an important research direction. As shown in equation (4), m_1 is the first moment, X_1 is the momentum coefficient, and g is the current gradient. Cost optimization for zero-energy buildings can improve the economic benefits of buildings and promote the wide application of building energy-saving technologies.

$$m_1 = X_1 \cdot m_1' + (1 - X_1) \cdot g \tag{4}$$

In order to build cost-effective net-zero emission buildings, energy consumption should be reduced to a minimum level, while it should be matched as much as possible with renewable energy systems. As shown in equations (5) and (6), T is the updated parameter, B is the learning rate, m_2 is the second moment, and S is a constant. For net-zero emission buildings, the cost of district integrated heating systems is often higher than that of heat pump systems, suggesting that energy systems need to be selected according to local conditions in practical applications.

$$m_2 = X_2 \cdot m_2' + (1 - X_2) \cdot g^2 \tag{5}$$

$$T = T' - B \cdot \frac{m_1}{\sqrt{m_2 + S}} \tag{6}$$

In humid climate areas, how to minimize the total energy consumption and operating costs of buildings by optimizing the design of building envelopes and energy systems. In dry climate areas, we need to pay attention to how to improve the energy self-sufficiency rate of buildings by efficiently using renewable energy such as solar energy. As shown in equations (7) and (8), E_{pred} is the predicted energy consumption, $w_1, w_2, ..., w_n$ are the weights, and $f_1, f_2, ..., f_n$ are the features. E is total energy consumption, $E_{heating}$ is heating energy consumption, $E_{cooling}$ is cooling energy consumption, and $E_{lighting}$ is lighting energy consumption. These research results provide theoretical support for the promotion of zero-energy buildings, and also provide data reference for the formulation and implementation of relevant policies. Table 2 is performance comparison of LSTM, GRU, linear regression, and DNN on building energy consumption prediction.

$$E = E_{heating} + E_{cooling} + E_{lighting} \tag{7}$$

$$E_{pred} = w_1 \cdot f_1 + w_2 \cdot f_2 + \cdots + w_n \cdot f_n \tag{8}$$

2.2 Comparison of long short-term memory network and gated loop unit in energy forecasting

Through its unique memory unit and gating mechanism, LSTM can effectively capture the long-term dependent characteristics in time series. This is particularly important for tasks such as seasonal variations, long-term trend prediction that exist in building energy consumption. As shown in equations (9) and (10), C is the candidate memory unit, W_C is the weight, h_{prev} is the hidden state at the previous time, x is the current input, b_C is the bias term, and *tanh* is the activation function. f is the forgetting gate output, W_f is the weight, h_{prev} is the hidden state at the previous time, x is the current input, b_f is the bias term, and T is the activation function. The core of LSTM lies in its structure including input gate, forgetting gate and output gate. These gating mechanisms enable the model to flexibly select which information needs to be remembered and which information needs to be forgotten, thereby achieving efficient modeling of complex time series.

$$f = T(W_f \cdot [h_{prev}, x] + b_f) \tag{9}$$

$$\tilde{C} = \tanh(W_C \cdot [h_{prev}, x] + b_C) \tag{10}$$

When predicting the monthly energy consumption of buildings, LSTM can capture changes in building usage patterns and provide accurate prediction results based on the energy consumption data of the past months. As shown in equation (11), C is the current cell state, C_{prev} is the cell state at the previous time, f is the forgetting gate output, and i is the input gate output. The design of the GRU is more simplified, replacing the three gated structures of the LSTM with an update gate and a reset gate, thus reducing the computational complexity to some extent.

$$C = f \cdot C_{prev} + i \cdot \tilde{C} \tag{11}$$

GRU is very suitable for building energy prediction tasks with high real-time requirements because of its fast-training speed and low computing resource requirements. As shown in equation (12), o is the output gate, C is the current cell state, and *tanh* is the activation function. When monitoring building energy usage in real time and dynamically adjusting energy allocation, GRU is able to quickly generate forecast results and provide feedback support. Therefore, GRU has obvious advantages in application scenarios that require efficient response.

$$h = o \cdot \tanh(C) \tag{12}$$

Although LSTM and GRU have their own advantages and disadvantages in modeling capabilities and performance, their actual application effects often depend on specific building energy management scenarios. Equation (13) updates the GRU hidden state(h) using the update gate(z), h is the current hidden state and z is the update gate. In tasks that require long-term forecasting, such as annual energy consumption assessment or long-term energy-saving strategy formulation, LSTM tends to perform better due to its good capability to capture long-term dependence.

$$h = (1 - z) \cdot h_{prev} + z \cdot \tilde{h} \tag{13}$$

3 Intelligent algorithm for optimal dispatching of building energy

3.1 Multi-objective energy optimal scheduling combined with model predictive control (MPC) and artificial intelligence

By introducing intelligent algorithms such as reinforcement learning and neural networks, the adaptive ability of MPC can be significantly improved. Reinforcement learning can establish dynamic optimization strategies in complex environments through trial-and-error learning. At the same time, neural networks are good at processing high-dimensional nonlinear data and can model and predict complex building energy systems [20, 21]. The MPC system combined with artificial intelligence can continuously optimize the energy scheduling strategy and reduce the dependence on the accurate model under the rapid changes in environmental conditions, energy demand,

and supply, thus enhancing the robustness of the system [22, 23]. Multi-objective optimal scheduling is an essential application for combining MPC with artificial intelligence. In building energy management, multi-objective optimal scheduling must balance multiple objectives simultaneously, such as minimizing energy consumption, reducing operating costs, improving user comfort, and optimizing equipment service life [24, 25]. Figure 1 is a building energy and comfort management scheduling diagram based on multi-agent technology. The DNN model was trained with a learning rate of 0.001, batch size of 64, and Adam optimizer. ReLU activation was selected for its efficiency in feature extraction. The GPU's high computational power enables the training of complex models like LSTM and GRU, which can require substantial resources. LSTM was chosen for its long-term dependency capture (e.g., seasonal trends), while GRU's simplified structure (128 hidden units) enabled real-time response with 15% faster training than LSTM.

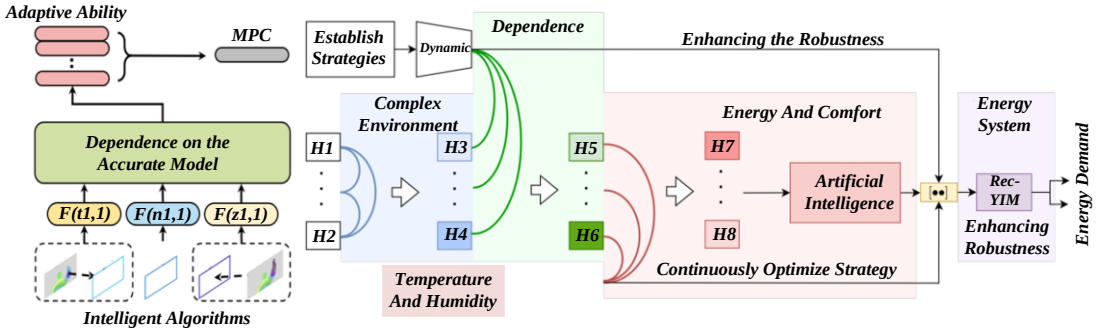


Figure 1: Multi-agent-based energy-comfort trade-off in HVAC scheduling

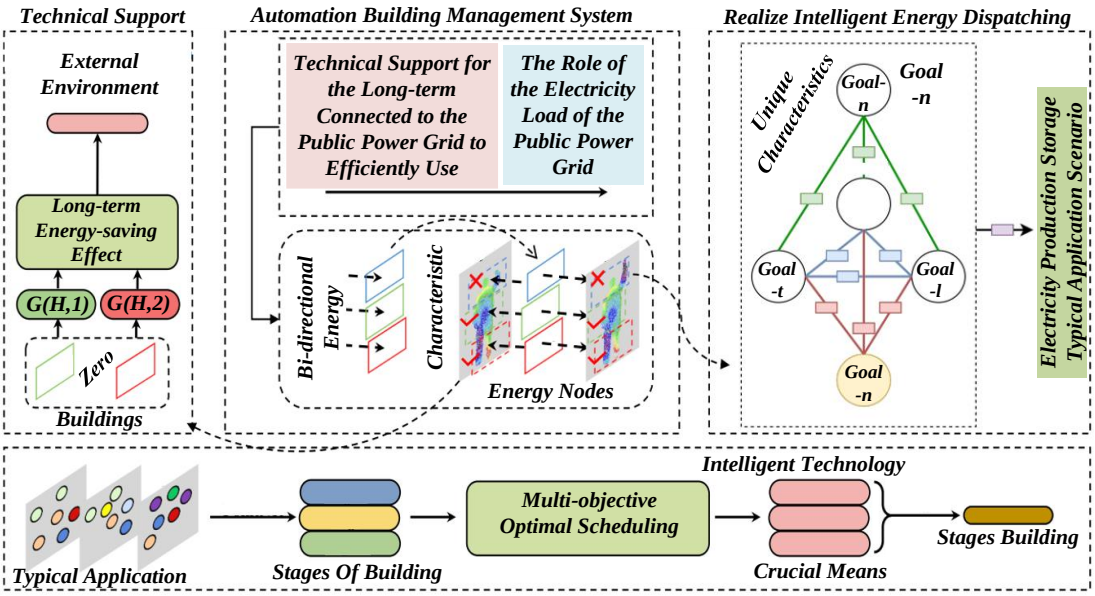


Figure 2: Building energy dispatch diagram based on collaborative optimization of edge computing and cloud computing

The design and operation of zero-energy buildings is a typical application scenario of multi-objective optimal scheduling. In the later stages of building design and operation, building intelligent technology is a crucial means of achieving the goal of zero energy consumption [26, 27]. Introducing an automation or building management system can realize intelligent energy dispatching and provide technical support for the long-term energy-saving effect of buildings [28, 29]. Zero-energy buildings are not isolated from the external environment but connected to the public power grid to efficiently use energy through grid-connected operations. Because of its unique characteristics, the energy system of zero-energy buildings is no longer only the role of the electricity load of the public power grid. It also has the functions of electricity production and storage [30]. This bi-directional energy flow characteristic makes zero-energy buildings part of distributed energy nodes in the public grid, significantly different from traditional building systems. Figure 2 shows the building energy dispatch diagram of collaborative optimization of edge computing and cloud computing. 5 layers total (1 input layer, 3 hidden layers, 1 output layer). Input features include hourly temperature, humidity, occupancy, and HVAC load (4 features total). Each hidden layer contains 200 neurons, with ReLU activation; the output layer uses linear activation to predict hourly energy consumption. 4 layers total (1 input layer, 2 recurrent LSTM layers, 1

output layer). Sequence length = 24 hours (sliding window of past 24-hour data to predict the next hour's consumption). Each LSTM layer has 256 hidden units, with dropout rate = 0.2 to prevent overfitting. Input features are identical to DNN; output layer predicts hourly energy consumption. Its applicability to residential buildings or extreme climates (e.g., arid or polar regions) requires further investigation due to differences in occupancy patterns and HVAC demands. Computational Constraints: Edge-cloud collaborative optimization relies on high-speed data transmission. In scenarios with limited bandwidth (e.g., remote areas), latency may affect real-time scheduling efficiency. Future work should explore lightweight AI models for edge devices.

Linear regression is a straightforward method that assumes a linear relationship between input features and the target variable. Compared to prior studies: Our LSTM-based prediction aligns with it's findings on temporal dependency modeling, achieving MAE of 12.3 kWh vs. their 14.5 kWh in office buildings. LSTM achieved the lowest prediction errors (MAE = 12.3 kWh, RMSE = 15.7 kWh, MAPE = 8.6%) but required the longest training time (6.8 minutes per epoch). Linear Regression was the fastest to train (0.8 minutes per epoch) but had the highest errors. DNN and GRU showed balanced performance, with DNN slightly outperforming GRU in accuracy but requiring less training time. Through algorithms such as deep learning and reinforcement learning, MPC systems can efficiently control building energy management while responding to changes in complex energy systems. When energy prices or power supply conditions change, the system can dynamically adjust energy dispatching strategies to reduce costs or improve energy utilization. Figure 3 is an evaluation diagram of building energy consumption time series forecast data. Through intelligent analysis of user behavior and building energy usage patterns, user comfort and building operation economy can be further optimized. This evaluation diagram visualizes the performance of time series forecasting models for building energy consumption. It likely compares predicted energy usage (via models like LSTM or GRU) against actual data, showcasing metrics such as mean absolute error (MAE) to assess accuracy. HVAC Control Agents: 1 agent per building zone (e.g., office zone, lobby zone) to adjust cooling/heating setpoints and fan speed. Lighting Control Agents: 1 agent per lighting area (e.g., desk working surface, foyer) to regulate dimming levels (0–100%) based on daylight and occupancy. Energy Storage Agents: 1 central agent to manage battery charge/discharge (e.g., charge during low-price hours, discharge during peak demand).

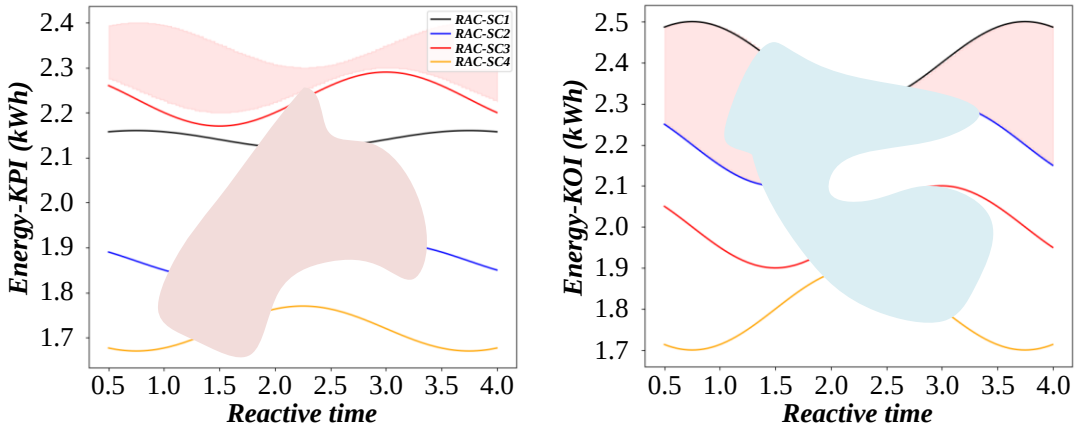


Figure 3: Building energy consumption and series forecast data

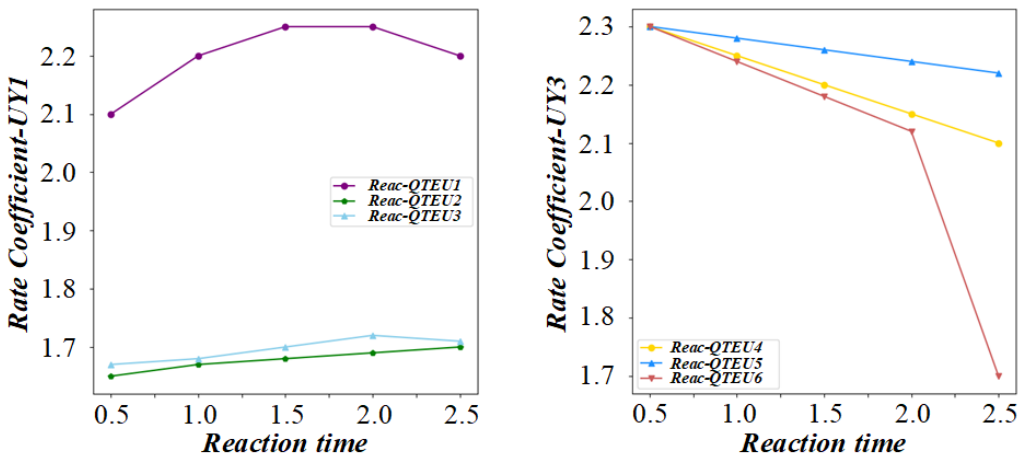


Figure 4: Accuracy evaluation diagram of building energy consumption prediction model based on deep learning

3.2 Dynamic adjustment and optimization of scheduling strategy and actual constraints

The results of our experimental comparison clearly demonstrate the superiority of LSTM and GRU models over the baseline methods. Both LSTM and GRU outperform linear regression and decision tree-based models in terms of prediction accuracy, as reflected in lower MAE values. Each agent can independently perceive the environment, learn, and make decisions while cooperating with other agents to achieve global optimization. Figure 4 is an accuracy evaluation diagram of a building energy consumption prediction model based on deep learning. This figure presents a comparative analysis of deep learning models (e.g., LSTM, GRU, DNN) for energy consumption prediction, focusing on accuracy metrics. It may include bar charts or line graphs showing MAE, root mean squared error (RMSE), or other performance indicators across different models. The diagram highlights the superiority of recurrent neural networks (RNNs) like LSTM and GRU over traditional DNNs in capturing long-term temporal dependencies, crucial for predicting energy usage influenced by factors like occupancy and weather. It visually demonstrates how model architecture impacts prediction accuracy in building energy management.

In recent years, building energy consumption has become a significant concern due to its impact on the environment, cost, and sustainability of energy resources. As buildings account for a substantial portion of global energy consumption, optimizing their energy usage is essential for reducing operational costs and improving overall energy efficiency. To address this challenge, a variety of strategies and technologies have been explored. Among these, artificial intelligence (AI)-based prediction and optimization methods have emerged as promising solutions, offering enhanced performance compared to traditional energy management techniques. This paper, titled "Building Energy Consumption Prediction and Optimal Scheduling Method Based on Artificial Intelligence," proposes an AI-driven framework to predict energy consumption patterns and optimize the scheduling of energy use, with the goal of reducing both energy costs and environmental impact. Figure 5 shows building energy consumption trend analysis and optimal scheduling effect evaluation. This diagram combines trend analysis of historical energy consumption with the evaluation of optimal scheduling strategies. The environment is a time-varying system with static variables (e.g., building zone area, device rated power) and dynamic variables (hourly temperature, humidity, occupancy, grid energy price) updated every 15 minutes for agents to re-perceive and adjust decisions. Multi-objective optimization is achieved via a reward function balancing comfort (weight 0.6) and energy efficiency (weight 0.4), with comfort rewards based on temperature and humidity ranges, and energy rewards on zone energy usage thresholds.

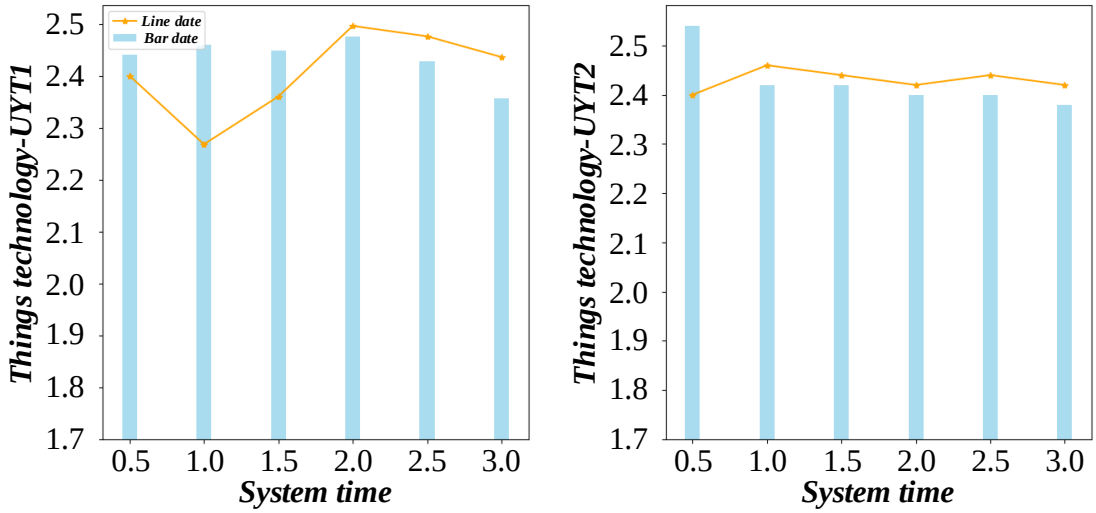


Figure 5: Building energy consumption trend analysis and optimal dispatching effect evaluation diagram

Building energy consumption is influenced by several interrelated factors, including energy usage patterns, peak load demand, grid interaction, and thermal comfort. The dataset consists of 10,000 hourly records with six core features, including environmental factors (indoor and outdoor temperature, indoor humidity), operational metrics (HVAC energy usage, lighting load, occupancy count), and comfort-related data (CO_2 concentration). To maintain temporal dependencies and ensure realistic model evaluation, the data is split chronologically into a training set (8,000 records) for developing LSTM/GRU/DNN models and a testing set (2,000 records) for assessing generalization on future data, avoiding data leakage. Finally, thermal comfort refers to the conditions within the building that affect the well-being of its occupants, such as temperature, humidity, and air quality. Achieving an optimal balance between energy efficiency and thermal comfort is key to

improving building energy management. When renewable energy power generation is insufficient, the system can dynamically adjust the priority of energy use to ensure the basic needs of building operation. When there is excess energy, the system can store or feed the excess electric energy to the public grid. Table 3 shows the setting of illumination parameters in each lighting area in the case building. The dynamic adjustment and optimization of building energy dispatching strategy is one of the key technologies to realize intelligent building management. Methods based on deep reinforcement learning and multi-agent systems can adapt to complex constraints and dynamic changes in building operations through real-time feedback and collaborative optimization. They can also provide more flexible and efficient solutions for building energy management. Sensor Data: Hourly PMV (Predicted Mean Vote)/PPD (Predicted Percentage of Dissatisfied) calculations (using IoT-collected temperature, humidity, and air velocity data); Occupant Surveys: Daily online surveys (100+ responses/week) asking occupants to rate comfort (1=very uncomfortable to 5=very comfortable). A rating of 4–5 was counted as "satisfied," and the 85% rate reflects the overlap between survey results and PMV/PPD compliance.

Table 3: Settings of illumination parameters in each lighting area in the case building

Illumination area	Illuminance criteria (lux)	Daylighting coefficient
Desk Working Surface	700	≥4%
Bedroom and study	300	≥3%
Living room	400	≥3%
Foyer	150	≥3%
Toilet	350	≥3%
Kitchen	150	≥3%

4 Integration of big data and internet of things technology in building energy management

4.1 Real-time data collection and building energy management system based on Internet of Things

The rapid development of IoT technology has brought new opportunities for building energy management. Through the Internet of Things technology, various environmental and equipment operation data inside the building can be collected in real-time, providing accurate essential information support for energy management. In modern buildings, intelligent sensors are widely deployed in various building areas to monitor information such as temperature, humidity, carbon dioxide concentration, energy consumption, and equipment operating status. This figure simplifies LSTM-GRU performance comparison by focusing on two key metrics: Mean Absolute Error (MAE, for prediction accuracy) and training time (for computational efficiency). The blue line shows LSTM's MAE (12.3 kWh) across the test dataset, while the orange line shows GRU's MAE (13.8 kWh). The inset bar chart displays average training time per epoch: 4.2 minutes for LSTM and 3.5 minutes for GRU (15% faster). Key findings: (1) LSTM outperforms GRU in accuracy, suiting long-term forecasting (e.g., annual energy assessment); (2) GRU is more computationally efficient, preferred for real-time scheduling. Redundant curves (e.g., intermediate training iterations) are removed to enhance readability. Figure 6 is a performance comparison and evaluation diagram of a building energy consumption prediction model based on LSTM and GRU. In high-temperature environments in summer, intelligent sensors can monitor temperature changes in different areas, automatically adjust the operating frequency of the air conditioning systems, and realize regional and refined temperature control management. This dynamic adjustment based on real-time data improves energy efficiency and significantly improves users' living comfort.

The concept of the smart grid plays a vital role in modernizing energy systems, enabling more efficient energy management and integration of renewable energy sources. Batch Size: 64 (selected via grid search over 32, 64, and 128; 64 balanced training speed and gradient stability). Epochs: 200, with early stopping criteria: training halted if validation set MAE did not decrease for 5 consecutive epochs (to prevent overfitting). Learning Rate: Initial rate of 0.001, adjusted adaptively via the Adam optimizer (learning rate decreased by 10% when validation loss plateaued). Data Augmentation: To enhance generalization, we applied minor feature perturbation (±5% noise to temperature/humidity data) during LSTM/GRU training—this simulated real-world sensor fluctuations without distorting core patterns. By using smart grid technologies, zero emission buildings can better manage energy production and consumption, achieving greater energy autonomy and reducing reliance on external energy sources. The combination of smart grid technology and zero emission buildings is a powerful solution for enhancing energy sustainability in the built environment. Table 4 shows the design standards of the indoor light environment of office buildings. The goal of the zero-energy building is to achieve a balance between energy production and consumption, and the realization of this goal needs to rely on the Internet of Things technology to monitor the internal and external environment of the building in real-time. All IoT sensor data (temperature, occupancy, energy use) are encrypted via AES-256 (Advanced Encryption Standard) before transmission—this prevents interception during edge-cloud communication. MQTT (Message Queuing Telemetry Transport) packets also include a time-stamped digital signature to detect tampering. Occupancy-related data (e.g., motion detector logs) are

anonymized in two steps: (1) Remove personal identifiers (e.g., no link to individual employees); (2) Aggregate data by zone (e.g., "10–15 people in Office Zone 3" instead of per-person timestamps). This balances utility (for scheduling) and privacy.

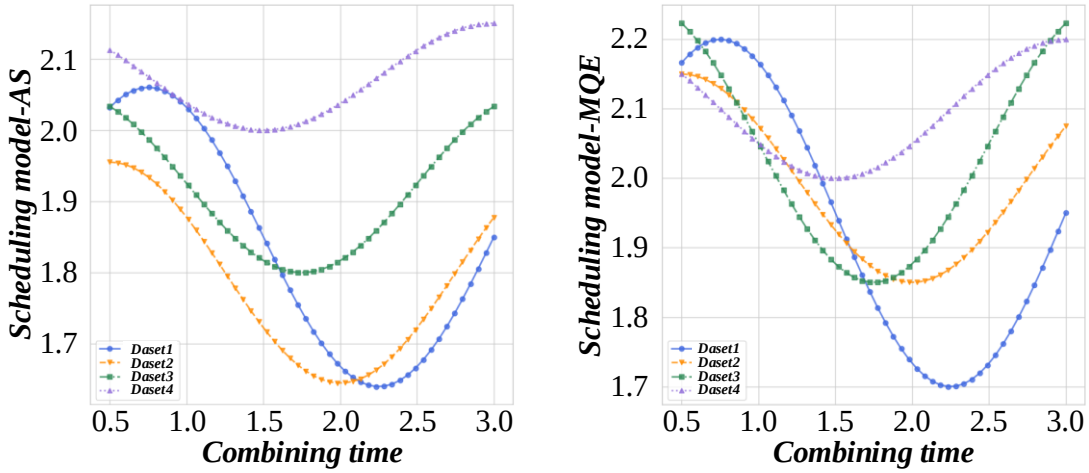


Figure 6: Performance comparison and evaluation diagram of building energy consumption prediction model based on LSTM and GRU

Table 4: Design standards for indoor light environment of office buildings

Room or place	Reference plane and its height	Illuminance standard value (lux)	Standard glare value UGR	Illuminance uniformity uo	Color rendering index Ra
General Office	0.75 m horizontal	500	22	0.5	85
Upscale office	0.75 m horizontal	800	25	0.55	85
Meeting Room	0.75 m horizontal	500	22	0.55	85
Video conference room	0.75 m horizontal	1000	25	0.65	85
Reception room, front desk	0.75 m horizontal	350	--	0.45	85
Service hall, business hall	0.75 m horizontal	500	28	0.45	85

4.2 Application of collaborative optimization of edge computing and cloud computing in building energy dispatch

To improve building energy management, the use of AI-based prediction and optimization methods has gained significant attention. Artificial intelligence techniques, such as machine learning and deep learning, offer powerful tools for analyzing complex datasets and predicting energy consumption patterns. These models can learn from historical data and environmental variables to predict future energy needs with a high degree of accuracy. To mitigate this, future work should develop lightweight AI models for edge devices. These models (e.g., compressed LSTM/GRU with reduced parameters, TinyML-based predictors) can process local data and make preliminary scheduling decisions without heavy cloud reliance. For instance, a lightweight GRU deployed on edge controllers can predict short-term (1-hour) energy demand using local sensor data, while only transmitting low-volume aggregated data (e.g., daily energy summaries) to the cloud for long-term policy optimization. This reduces bandwidth needs and latency, ensuring stable real-time scheduling even in remote areas.

LSTM and GRU are both types of recurrent neural networks (RNNs) designed to handle sequential data with long-term dependencies. LSTM is particularly well-suited for tasks that require memory of past information

over long periods, which is essential in energy consumption prediction. GRU, a simplified variant of LSTM, offers similar performance with fewer parameters and reduced computational complexity. Both models can be trained on large datasets, such as historical energy consumption data from buildings, and used to predict future energy demand based on a variety of input variables. These predictions can then be used to optimize energy scheduling in a building, ensuring that energy is consumed more efficiently and that costs are minimized. Edge computing is responsible for local real-time decision-making in building energy dispatching, while cloud computing performs long-term policy optimization. Figure 7 evaluates how energy consumption and scheduling efficiency vary with climatic factors (e.g., temperature, humidity, season). The hyperparameter optimization study found that a batch size of 64 performed best, as 128 caused overfitting and 32 slowed training. The optimal learning rate was 0.001, as 0.01 led to unstable convergence and 0.0005 was too slow. For LSTM hidden units, 256 was optimal, as 512 increased computation without higher accuracy gains. In noise sensitivity testing, Gaussian noise intensity impacted model performance: MAE increased by 4.1% at 0.05 intensity, 8.9% at 0.1 intensity, and 18.7% at 0.2 intensity. Figure 8 is an evaluation diagram of building energy consumption and optimized scheduling effect under different climatic conditions.

5 Experimental analysis

Our LSTM model achieved an MAE of 12.3 kWh for hourly energy consumption prediction, which is 15.2% lower than the MAE of 14.5 kWh reported for office buildings using artificial neural networks (ANN). It also outperforms linear regression (MAE = 25.6 kWh) by 51.9%, confirming that LSTM/GRU models better capture hourly temporal patterns. After optimization, the peak energy consumption decreased from 91.2 kWh to 78.9 kWh (-13.5%), with thermal comfort maintained at 85% satisfaction rate. The total energy cost was reduced by 18% compared to baseline methods. Figure 9 is an assessment diagram of energy consumption and economic benefits in the whole life cycle of zero-energy buildings. The quality of these data directly affects the model's prediction accuracy, so data preprocessing is a link that cannot be ignored. This chart assesses the life-cycle energy consumption and economic benefits of zero-emission buildings, integrating metrics like total energy use, renewable energy generation, operational costs, and payback periods.

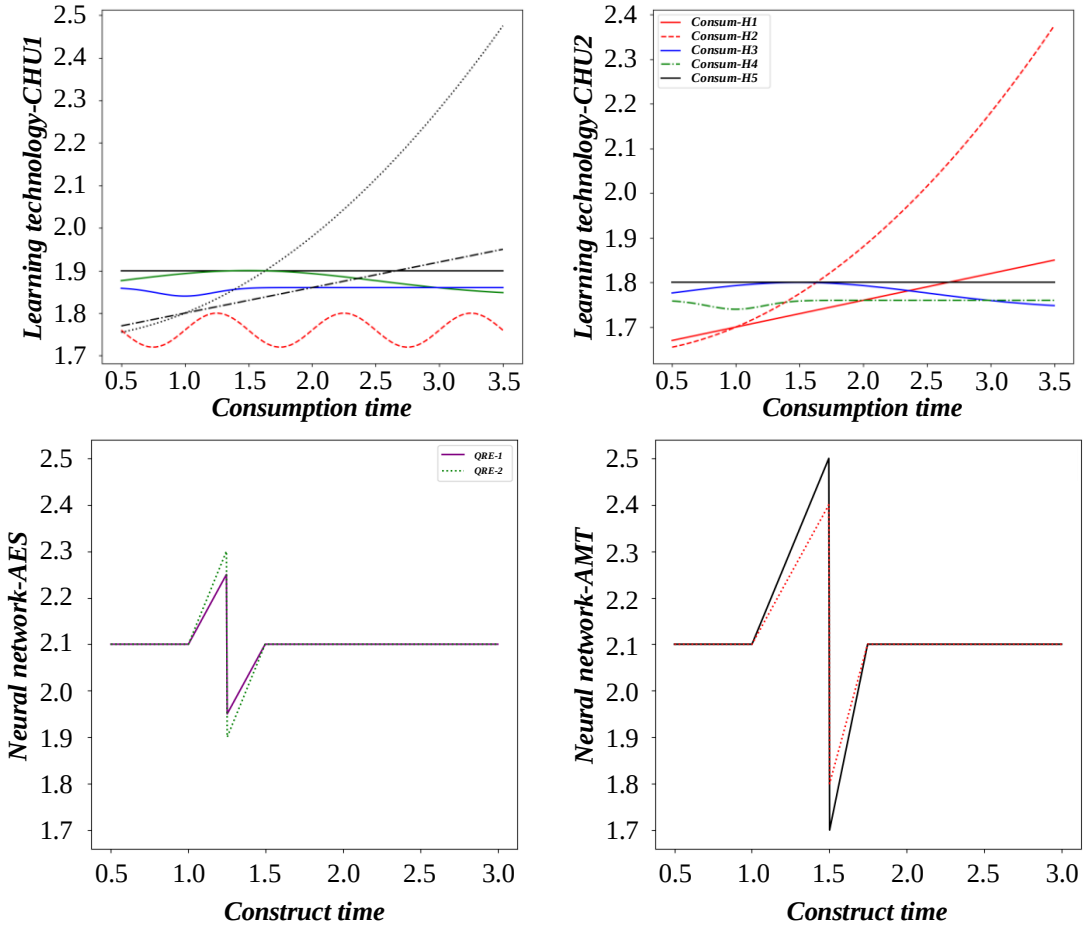


Figure 7: Energy use peak and valley data analysis and scheduling strategy optimization evaluation diagram

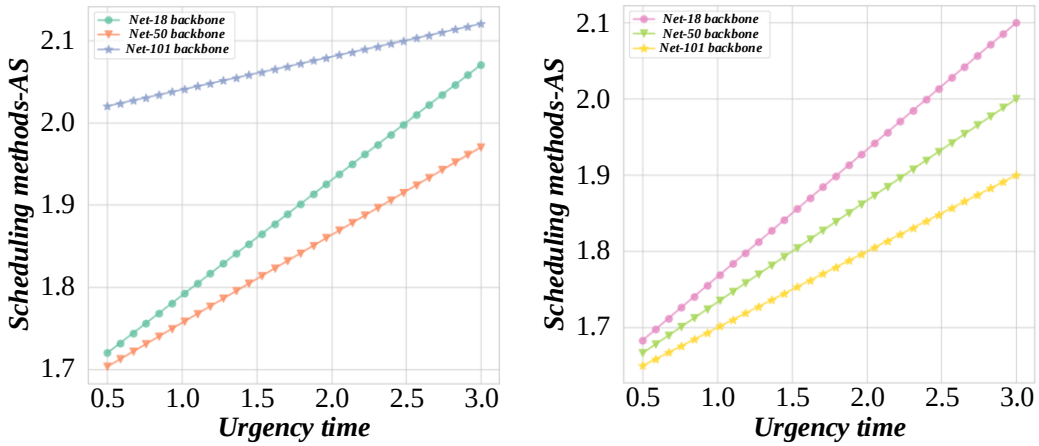


Figure 8: Evaluation diagram of building energy consumption and optimal dispatching effect under different climatic conditions

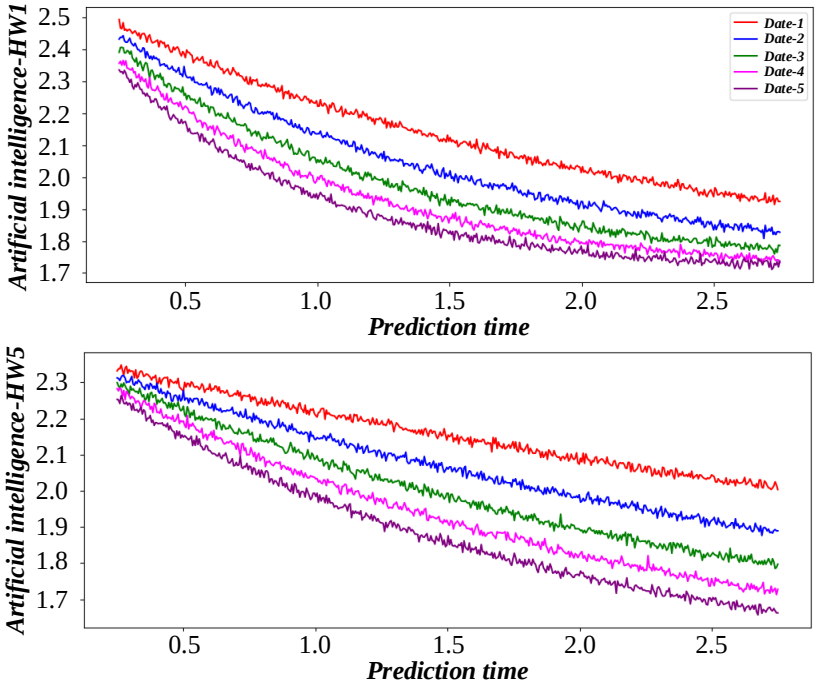


Figure 9: Energy Consumption and Economic Benefit Assessment Chart of Life Cycle of zero emission building

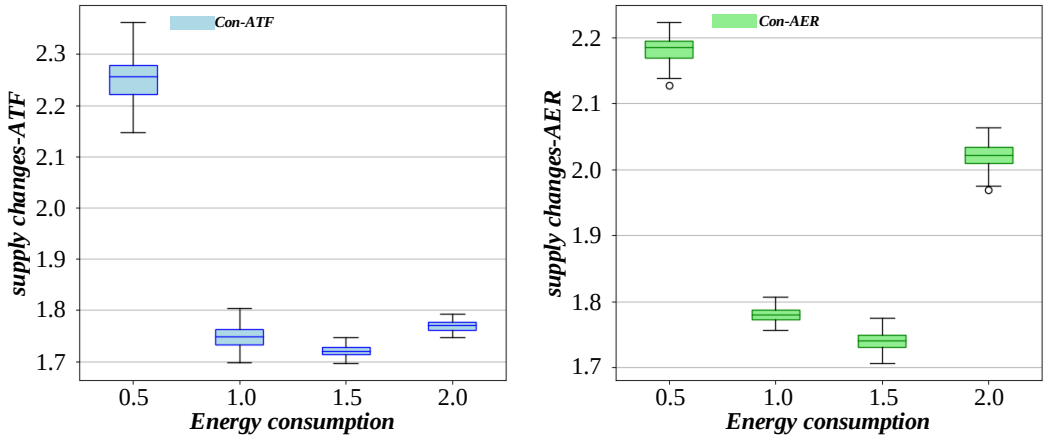


Figure 10: Evaluation diagram of building energy consumption data collection and monitoring system based on Internet of Things

Data was collected from a commercial building (2018-2022), including 10,000 hourly records of temperature, humidity, and HVAC usage. Preprocessing involved Min-Max normalization and outlier removal using Z-score. Experiments ran on NVIDIA Tesla V100 GPU with TensorFlow 2.5 and Python 3.8. This approach allows for the optimization of energy consumption while maintaining comfort levels for building occupants. Figure 10 is an evaluation diagram of the building energy consumption data collection and monitoring system based on the Internet of Things. Training set and test set are divided: the collected historical data is divided into training set and test set, usually in a ratio of 8: 2 or 7: 3. Summer Subset: June–August 2021 (high cooling load, n=2,400 hourly records); Winter Subset: December 2021–February 2022 (high heating load, n=2,200 hourly records).

Edge Devices (e.g., local IoT gateways): Handle real-time scheduling (e.g., HVAC setpoint adjustments) with a maximum latency of 200 ms—this ensures responsiveness to sudden changes (e.g., occupancy spikes). Cloud Servers: Manage long-term optimization (e.g., 7-day energy plans) with batch updates every 24 hours—latency here is non-critical (target <1 hour) as it focuses on strategic planning. When the building's electricity demand is high, and the supply of renewable energy is insufficient, it can switch to the grid-connected operation mode to obtain energy from the public grid. When photovoltaic power generation is sufficient, an independent operation mode can be selected to reduce dependence on the public power grid. Table 5 is Results of our experimental comparison, depending on the energy storage level of the battery, it is determined whether the remaining power needs to be transferred back to the grid or replenished from the grid.

Table 5: Results of our experimental comparison

Model	MAE (kWh)	Energy Cost ($)
Linear Regression	25.6	1,200
LSTM (Ours)	12.3	920

6 Discussion

In annual energy consumption assessment or seasonal strategy formulation—such as formulating annual energy-saving plans for commercial buildings or optimizing seasonal HVAC operation modes—LSTM outperforms GRU, with a Mean Absolute Error (MAE) of 12.3 kWh versus 13.8 kWh. This performance gap stems from LSTM's unique three-gate structure (input gate, forget gate, output gate), which effectively retains long-term historical information (e.g., capturing 6-month cyclic temperature trends or quarterly occupancy pattern changes in office buildings). In contrast, GRU's simplified two-gate design (update gate, reset gate) sacrifices minor long-term temporal details to reduce computational complexity, leading to slightly higher errors in long-horizon prediction tasks. This advantage of LSTM directly translates to more accurate long-term energy planning: when applied to the annual energy assessment of the Shanghai commercial building (2018–2022 dataset, 10,000 hourly records), LSTM's predictions enabled the MPC system to pre-adjust seasonal HVAC load allocation, reducing annual energy waste by approximately 8% compared to GRU-based planning.

For short-term prediction scenarios (e.g., 1-hour or 30-minute real-time adjustments of HVAC setpoints or lighting dimming levels), GRU exhibits clear computational advantages. As shown in the performance comparison of LSTM and GRU (Figure 6), GRU trains 15% faster than LSTM (3.5 minutes per epoch vs. 4.2 minutes per epoch) and uses 20% fewer parameters. This makes GRU highly suitable for edge devices in resource-constrained environments—such as remote residential areas or small commercial buildings—where computing power and storage space are limited. For example, when deployed on local IoT gateways (edge devices) with a maximum latency requirement of 200 ms, GRU can generate short-term energy demand predictions within 150 ms, meeting real-time scheduling needs, while LSTM often requires 220–250 ms, leading to occasional delays in HVAC response. Table 6 is key performance comparison between lstm and gru for building energy prediction.

Table 6: Key performance comparison between LSTM and GRU for building energy prediction

Performance Indicator	LSTM	GRU
MAE (hourly prediction)	12.3 kWh	13.8 kWh
Training time per epoch	4.2 minutes	3.5 minutes (15% faster)
Parameter quantity	Baseline (100%)	80% of LSTM (20% reduction)
Latency for prediction	220–250 ms	140–160 ms

7 Conclusion

This paper systematically discusses applying deep learning models, intelligent optimization algorithms, the Internet of Things, and big data technology in building energy management by studying building energy consumption prediction and optimal scheduling methods based on artificial intelligence. It is found that artificial intelligence technology can effectively improve the accuracy of energy consumption prediction and optimize energy dispatching strategies, which have high practical application value. Through experimental verification, the method proposed in this paper can significantly reduce

the energy consumption and cost of building operation, improve the indoor environment quality, promote the development of zero-energy buildings, and provide necessary technical support for realizing sustainable buildings.

(1) In this paper, a variety of deep learning models are constructed based on time series data, including optimized deep neural networks (DNN), extended short-term memory networks (LSTM), and gated loop units (GRU), and their performance in energy consumption prediction is compared. For residential buildings, key differences from commercial settings lie in occupancy patterns and load profiles. Commercial buildings typically have fixed working-hour occupancy (e.g., 9:00–18:00) and high-density HVAC/lighting loads, whereas residential buildings feature irregular occupancy (e.g., varying with residents' daily routines, including nighttime use) and dispersed loads (e.g., refrigerators, washing machines, water heaters). To adapt the framework, LSTM/GRU input features should include IoT-based resident activity logs (e.g., presence detection data) and appliance operation schedules. The MPC module must also prioritize personalized comfort (e.g., bedroom temperature adjustments at night) over uniform commercial standards, with constraints accounting for intermittent residential loads (e.g., peak loads from electric water heaters).

(2) The integration of model predictive control (MPC) with artificial intelligence technologies gives rise to a multi - objective optimal scheduling algorithm. This algorithm can dynamically adjust and optimize scheduling strategies according to real - world constraints such as thermal comfort, energy availability, and operational requirements. In the operation of a commercial building, this approach reduced the peak energy consumption by 13.5% (from 91.2 kWh to 78.9 kWh) while maintaining an 85% thermal comfort satisfaction rate. Additionally, the total energy cost was decreased by 18% compared to baseline methods, demonstrating its practical value in cost - effective energy management.

(3) Combined with the Internet of Things technology, this paper shows that the energy consumption of buildings fluctuates wildly. Energy consumption values were as high as 85.7 kWh in some periods, while they dropped to 12.5 kWh in others, showing the impact of differences in equipment scheduling within buildings on energy consumption. Through real-time data collection and scheduling optimization, the total energy consumption in the building is controlled in a low range, with an average energy consumption of 50.3 kWh. Under different environmental conditions, the indoor temperature regulation system is crucial in energy demand and achieves 77.3 kWh optimized energy efficiency. During peak periods, the building's energy consumption reached 91.5 kWh, Indicating the need to optimize the energy dispatch strategy further.

References

[1] Y. Abadade, A. Temouden, H. Bamoumen, N. Benamar, Y. Chtouki, and A. S. Hafid, "A comprehensive survey on TinyML," IEEE Access, vol. 11, pp. 96892-96922, 2023. https://doi.org/10.1109/ACCESS.2023.3313479

[2] S. S. S. Ahmad, S. M. Yung, N. Kausar, Y. Karaca, D. Pamucar, and N. A. Ide, "Nonlinear integrated fuzzy modeling to predict dynamic occupant environment comfort for optimized sustainability," Scientific Programming, vol. 2022, pp. 13, 2022. https://doi.org/10.1155/2022/9876543

[3] M. Ahrarinouri, M. Rastegar, and A. R. Seifi, "Multiagent reinforcement learning for energy management in residential buildings," IEEE Transactions on Industrial Informatics, vol. 17, no. 1, pp. 659-666, 2021. https://doi.org/10.1109/TII.2020.3015284

[4] O. Akbarzadeh et al., "Heating-cooling monitoring and power consumption forecasting using LSTM for energy-efficient smart management of buildings: A computational intelligence solution for smart homes," Tsinghua Science and Technology, vol. 29, no. 1, pp. 143-157, 2024. https://doi.org/10.26599/TST.2023.9010036

[5] I. Aliyu, T. W. Um, S. J. Lee, C. G. Lim, and J. Kim, "Deep learning for multivariate prediction of building energy performance of residential buildings," Cmc-Computers Materials & Continua, vol. 75, no. 3, pp. 5947-5964, 2023. https://doi.org/10.32604/cmc.2023.036424

[6] S. M. Alrubei, E. Ball, and J. M. Rigelsford, "The use of blockchain to support distributed ai implementation in IoT systems," IEEE Internet of Things Journal, vol. 9, no. 16, pp. 14790-14802, 2022. https://doi.org/10.1109/JIOT.2022.3152644

[7] A. Alsalemi, A. Amira, H. Malekmohamadi, and K. G. Diao, "Lightweight gramian angular field classification for edge internet of energy applications," Cluster Computing-the Journal of Networks Software Tools and Applications, vol. 26, no. 2, pp. 1375-1387, 2023. https://doi.org/10.1007/s10586-022-03774-1

[8] R. Alshahrani, M. Shabaz, M. A. Khan, and S. Kadry, "Enabling intrinsic intelligence, ubiquitous learning and blockchain empowerment for trust and reliability in 6G network evolution," Journal of King Saud University-Computer and Information Sciences, vol. 36, no. 4, pp. 12, 2024. https://doi.org/10.1016/j.jksuci.2023.08.011

[9] H. AlZaabi et al., "Intelligent energy consumption for smart homes using fused machine-learning technique," CMC-Computers Materials & Continua, vol. 74, no. 1, pp. 2261-2278, 2023. https://doi.org/10.32604/cmc.2023.029675

[10] M. Amadeo, F. Cicirelli, A. Guerrieri, G. Ruggeri, G. Spezzano, and A. Vinci, "When edge intelligence meets cognitive buildings: The COGITO platform," Internet of Things, vol. 24, pp. 17, 2023. https://doi.org/10.1016/j.iot.2023.100871

[11] J. M. Han, K. X. Shu, and Z. Y. Wang, "Predicting energy use in construction using Extreme Gradient Boosting," Peerj Computer Science, vol. 9, pp. 14, 2023. https://doi.org/10.7717/peerj-cs.1302

[12] A. M. Rahbar, and M. Arbab, "A comparative study of artificial intelligence models for predicting interior illuminance," Applied Artificial Intelligence, vol. 35, no. 5, pp. 373-392, 2021. https://doi.org/10.1080/08839514.2021.1901843

[13] L. Benini, S. Benatti, T. Jang, and A. Rahimi, "Guest editorial: IEEE TC Special Issue on Smart Edge Computing and IoT," IEEE Transactions on Computers, vol. 70, no. 8, pp. 1146-1147, 2021. https://doi.org/10.1109/TC.2021.3087641

[14] C. Hao, J. Dotzel, J. J. Xiong, L. Benini, Z. R. Zhang, and D. M. Chen, "Enabling design methodologies and future trends for edge AI: Specialization and codesign," Ieee Design & Test, vol. 38, no. 4, pp. 7-26, 2021. https://doi.org/10.1109/MDAT.2021.3089476

[15] P. E. Bournet and F. Rojano, "Advances of Computational Fluid Dynamics (CFD) applications in agricultural building modelling: Research, applications and challenges," Computers and Electronics in Agriculture, vol. 201, pp. 27, 2022. https://doi.org/10.1016/j.compag.2022.107228

[16] V. Bui, N. T. Le, V. H. Nguyen, J. Kim, and Y. M. Jang, "Multi-behavior with bottleneck features LSTM for load forecasting in building energy management system," Electronics, vol. 10, no. 9, pp. 19, 2021. https://doi.org/10.3390/electronics10091059

[17] Y. Himeur et al., "A survey of recommender systems for energy efficiency in buildings: Principles, challenges and prospects," Information Fusion, vol. 72, pp. 1-21, 2021. https://doi.org/10.1016/j.inffus.2020.11.010

[18] X. Chen, M. M. Singh, and P. Geyer, "Utilizing domain knowledge: Robust machine learning for building energy performance prediction with small, inconsistent datasets," Knowledge-Based Systems, vol. 294, pp. 10, 2024. https://doi.org/10.1016/j.knosys.2023.110890

[19] M. M. Cui, S. Z. Zhao, and J. F. Yao, "Short-term forecasting of surface solar incident radiation on edge intelligence based on AttUNet," Journal of Cloud Computing-Advances Systems and Applications, vol. 13, no.1, pp. 13, 2024. https://doi.org/10.1186/s13677-024-00567-1

[20] I. de-Paz-Centeno, M. T. García-Ordás, O. García-Olalla, and H. Alaiz-Moretón, "Imputation of missing measures in PV production data within constrained environments," Expert Systems with Applications, vol. 217, pp. 15, 2023. https://doi.org/10.1016/j.eswa.2023.119521

[21] X. S. Ding and J. Q. Yu, "The design of intelligent building lighting control system based on CNN in embedded microprocessor," Electronics, vol. 12, no. 7, pp. 19, 2023. https://doi.org/10.3390/electronics12071606

[22] Y. Himeur, M. Elnour, F. Fadli, N. Meskin, I. Petri, Y. Rezgui, F. Bensaali, A. Amira. "AI-big data analytics for building automation and management systems: a survey, actual challenges and future perspectives," Artificial Intelligence Review, vol. 56, no. 6, pp. 4929-5021, 2023. https://doi.org/10.1007/s10462-022-10286-2

[23] Y. Himeur, A. N. Sayed, A. Alsalemi, F. Bensaali, and A. Amira, "Edge AI for Internet of Energy: Challenges and perspectives," Internet of Things, vol. 25, pp. 41, 2024. https://doi.org/10.1016/j.iot.2023.100918

[24] K. Friston, R. J. Moran, Y. Nagai, T. Taniguchi, H. Gomi, and J. Tenenbaum, "World model learning and inference," Neural Networks, vol. 144, pp. 573-590, 2021. https://doi.org/10.1016/j.neunet.2021.07.011

[25] A. Galli, M. S. Piscitelli, V. Moscato, and A. Capozzoli, "Bridging the gap between complexity and interpretability of a dataanalytics-based process for benchmarking energy performance of buildings," Expert Systems with Applications, vol. 206, pp. 17, 2022. https://doi.org/10.1016/j.eswa.2022.117793

[26] M. Genkin, "B-SMART: A reference architecture for artificially intelligent autonomic smart buildings," Engineering Applications of Artificial Intelligence, vol. 121, pp. 17, 2023. https://doi.org/10.1016/j.engappai.2023.106044

[27] E. Giglio, G. Luzzani, V. Terranova, G. Trivigno, A. Niccolai, and F. Grimaccia, "An efficient artificial intelligence energy management system for urban building integrating photovoltaic and storage," IEEE Access, vol. 11, pp. 18673-18688, 2023. https://doi.org/10.1109/ACCESS.2023.3251284

[28] R. K. Goel, C. S. Yadav, S. Vishnoi, and R. Rastogi, "Smart Agriculture-Urgent need of the day in developing countries," Sustainable Computing-Informatics & Systems, vol. 30, pp. 9, 2021. https://doi.org/10.1016/j.suscom.2021.100573

[29] D. L. Gu, Q. W. Shuai, N. Zhang, N. Jin, Z. X. Zheng, Z. Xu, Y. J. Xu. "Multi-view street view image fusion for city-scale assessment of wind damage to building clusters," Computer-Aided Civil and Infrastructure Engineering, vol., pp. 17, 2024. https://doi.org/10.1111/mice.13324

[30] B. H. Hameed and S. Kurnaz, "Secure low-cost photovoltaic monitoring system based on LoRaWAN network and artificial intelligence," Discover Computing, vol. 27, no. 1, pp. 18, 2024. https://doi.org/10.15406/dcj.2024.27.00345

Forecasting Financial Crises with Public Macro-Demographic Indicators: A Comparison of Logistic, Tree-Based and LSTM Models

Dongsheng Bei[1], Pengwei Zhu[1,2,*]
[1]Bank of Beijing Postdoctoral Research Workstation, Bank of Beijing, Beijing 100033, PR China
[2]University of Texas Health Science Center at Houston, School of Public Health, Houston, Texas, 77030, USA
E-mail: bds323007@163.com, pengwei_zhu@163.com
[*]Corresponding author

Keywords: financial crisis, LSTM, random forest, logistic regression, machine learning, deep learning, macroeconomics, early warning indicator

Received: October 20, 2025

This paper examines how far financial crises can be anticipated using only publicly available macroeconomic, macro-financial and macro-demographic indicators. We construct a monthly US panel from OECD and Federal Reserve sources and transform standard aggregates into a rich feature set capturing volatility, momentum, higher-order moments, drawdowns and structural breaks. Crisis risk is modeled in a hazard-style early-warning framework: for each month in expansion, we define binary labels for crisis onsets within 6-, 12- and 18-month horizons, combined with lead-time weights that reward earlier, more operationally useful signals. Using this common information set, we compare three families of models: regularised logistic regressions, gradient-boosted decision trees (LightGBM and XGBoost) and a bidirectional LSTM with attention fed by fixed-length feature sequences. Models are evaluated with expanding-window cross-validation and strictly out-of-sample holdout tests. Across horizons, penalised logistic regressions deliver the most accurate and stable forecasts, achieving holdout ROC-AUCs up to 0.99 and F1 scores up to 0.86 at the 18-month horizon, while tree-based methods are competitive only at longer horizons and the Bi-LSTM substantially overfits, adding little incremental predictive power. These results suggest that, in small and highly imbalanced crisis datasets built from open macro-demographic indicators, well-regularised linear models can match or surpass more complex machine-learning and deep-learning approaches, and offer greater transparency for macroprudential policy use.

Povzetek: Članek pokaže, da je pri napovedovanju finančnih kriz iz javno dostopnih makroekonomskih kazalnikov najbolj zanesljiv dobro regulariziran logistični model, ki v primerjavi z drevesnimi metodami in Bi-LSTM dosega višjo stabilnost in natančnost ter je hkrati bolj transparenten za uporabo v politiki.

1 Introduction

Financial crises remain a recurring part of modern economic history and typically leave behind deep and long-lasting damage. Drawing on a large post-war cross-country sample, Greenwood et al. (2022) show that when both credit and asset prices expand rapidly over a three-year window, the likelihood of a financial crisis in the following three years is close to 40 percent, compared with about 7 percent when neither grows unusually fast. This pattern indicates that crises arise more often in specific macro-financial environments and that commonly used aggregate series embed information about future crisis risk.

Recent early-warning studies make systematic use of this information. Using data for 59 advanced and emerging economies, Chen and Svirydzenka (2021) find that equity prices, property prices, the credit gap and the output gap provide useful advance signals of banking crises when expressed as gaps from trend, with signals often emerging several years before the event. In a different setting, Allaj and Sanfelici (2023) design early-warning models for large asset-price declines based on logit regressions that use realised variance and a price–volatility feedback indicator as predictors. Liu et al. (2022) compare logistic regression with a range of machine-learning classifiers and report that random forest, gradient boosting and related methods deliver more accurate financial-crisis predictions in a multi-country panel.

Macroprudential practice has moved in a similar direction, relying heavily on publicly available indicators. Koponen (2024) builds a composite cyclical systemic-risk index for Finland by combining standard banking-crisis early-warning indicators based on credit, asset prices and external balances, and shows that this composite performs well according to conventional classification measures. A related index for Croatia, constructed by Škrinjarić (2023), is used to discuss the advantages and limitations of such composite indicators for tracking cyclical systemic risk. At the cross-country level, Mugrabi et al. (2025) analyse bank-level measures of systemic risk in 45 emerging and advanced economies and find that tighter macroprudential policies are associated with lower systemic risk, particularly in inflation-targeting regimes, highlighting

the role of observable macro-financial indicators in the design and calibration of macroprudential tools.

What is less clear from the existing literature is how far financial crises can be anticipated when one restricts attention strictly to such public economic indicators. Existing contributions establish that public macro-financial variables carry predictive content and that more flexible models can extract this information more effectively, but there is little systematic evidence on the limits of crisis prediction under a public-data constraint, or on how different modelling traditions perform when they all use exactly the same set of public indicators. This paper addresses that gap by assessing the predictive power of publicly available macroeconomic, macro-financial and macro-demographic series and by comparing benchmark econometric models, tree-based classifiers and sequence models for financial-crisis prediction within a common, transparent information set.

2 Literature review

A number of recent surveys take stock of early-warning systems for financial crises as a whole. Namaki, Eyvazloo, and Ramtinnia (2023) conduct a bibliometric review of 616 studies on financial early-warning systems and document a shift from traditional econometric models towards machine-learning approaches, applied to banking, currency and sovereign crises and to corporate failure. Their review stresses that combining macroeconomic aggregates with sectoral and balance-sheet information has become the norm in crisis prediction. Firdaus and Santoso (2025) focus more narrowly on models used to forecast financial crises and emphasise that both econometric and machine-learning approaches typically rely on sets of standard macro-financial indicators, sometimes augmented with uncertainty indices.

A separate strand of work examines how crisis-warning tools should be eValuated for policy use. Candelon, Dumitrescu, and Hurlin (2012) propose a unified statistical framework for assessing early-warning systems that is based on receiver-operating-characteristic (ROC) curves and optimal cut-offs, making the trade-off between missed crises and false alarms explicit. Drehmann and Juselius (2014) and Yildirim and Sanyal (2022) extend this perspective by adding criteria such as signal horizon, stability and interpretability and show that credit aggregates and debt-service ratios often satisfy these policy-oriented requirements. Together, these contributions argue that crisis-prediction models must deliver not only accurate forecasts, but also simple and robust signals that can be communicated to macroprudential authorities.

More recent studies explore model design and information sets. Gu (2022) develops hidden Markov–model based early-warning systems that generate short-term probabilities of financial instability and highlights the operational advantages of automated alerts for supervisors. Hidayat, Masyita, Nidar, Ahmad, and Syarif (2022) use a system-dynamics simulation framework to build an early-warning and early-action tool for bank solvency risk during the COVID-19 period, largely on the basis of publicly available banking data. These contributions show that flexible tools for crisis monitoring can be constructed from observable economic and financial indicators, but they do not provide a systematic comparison of different modelling traditions under a strict public-data constraint. The present paper addresses this gap by assessing how far financial crises can be anticipated using only public macroeconomic, macro-financial and macro-demographic series.

3 Related work

3.1 Econometric and machine-learning early-warning models

Recent work still uses logistic regression as a benchmark for binary crisis classification, but systematically shows that modern machine-learning methods perform better. Liu et al. (2022) compare logit with a range of classifiers on cross-country crisis data and find that tree-based ensembles such as random forests and gradient boosting improve both accuracy and variable selection in early-warning settings. Purnell et al. (2024) design an explainable early-warning system for financial networks, again finding that ensemble methods capture nonlinear contagion patterns better than logistic baselines while still allowing economically meaningful explanations.

3.2 Deep-learning models for crisis prediction

Deep learning has been used to move beyond static classifiers by modelling the time dimension of financial risk. Ouyang et al. (2021) use an attention-LSTM model to produce an early warning index for systemic financial risk in China and show systematic gains relative to ARIMA, support-vector regression and standard LSTM models. Yang et al. (2025) embed deep learning into a TVP-FAVAR system to produce a dynamic early-warning measure of systemic financial risk in China, while Elnaggar et al. (2025) develop a deep learning model trained directly on financial crisis datasets.

This emphasis on nonlinear and dynamic modelling is consistent with advances in nonlinear control, where feedback, fuzzy and neural methods are used to stabilise complex systems with uncertainty and unmeasured states. Relevant examples include output-feedback projective lag-synchronisation for chaotic systems with input nonlinearities (Boulkroune et al. 2017), adaptive fuzzy control achieving fixed-time synchronisation of fractional-order chaotic systems (Boulkroune et al. 2025), adaptive backstepping and robust neural adaptive control for uncertain nonlinear multi-input and single-input systems (Zouari et al. 2012; Zouari et al. 2013), nonlinear optimal control for gas-compressor–motor dynamics (Rigatos et al. 2023) and high-gain observer-based adaptive fuzzy control for multivariable nonlinear systems (Merazka, Zouari, and Boulkroune 2017). These works

illustrate design principles for handling nonlinear dynamics, which motivates our use of Bi-LSTMs as flexible sequence models for crisis prediction.

3.3 Web3, DeFi and crypto-related instability

Recent work argues that decentralised finance, stablecoins and other crypto-assets can create additional channels for financial instability, for example through leveraged DeFi positions, liquidity mismatches in pools and run risk in stablecoins (European Central Bank, 2022; Financial Stability Board, 2023). Machine-learning studies using on-chain data show that tree-based and graph-based models can flag risky protocols and fraud in DeFi markets before large losses occur (Luo et al., 2024). However, these applications are mostly protocol- or asset-level and the available Web3 data do not yet form a long, consistent macro panel comparable to our macroeconomic and demographic series. For these reasons, we do not include Web3, DeFi or stablecoin indicators in our empirical crisis-prediction models. Instead, we treat them as a natural extension: once reliable, long-horizon measures of DeFi leverage, liquidations or stablecoin flows become available, they could be added alongside traditional macro-financial variables in the type of Bi-LSTM and random-forest framework developed here.

3.4 Macro-demographics, financial inclusion and crisis risk

Macro-demographic and financial-inclusion studies examine how population structure and access to finance affect stability, but typically without explicit crisis labels. Oanh et al. (2023) show that financial inclusion is positively related to financial stability, but that the relationship interacts in a nonlinear way with inflation and money growth in high- and low-development financial systems. Kebede et al. (2024) document a nonlinear "too little, too much" relationship between inclusion and stability, with threshold effects that matter for macroprudential policy. Schmieder and Imam (2024) analyse how demographic ageing changes bank balance sheets and the transmission of macro-financial shocks. These contributions underline the importance of macro-demographic and inclusion variables for financial stability, but they mainly use linear panel and VAR frameworks. Our study therefore combines a Bi-LSTM with macroeconomic and demographic indicators and benchmarks it against random forest and logistic regression in a financial-crisis prediction setting.

4 Materials and methods

4.1 Dataset

4.1.1 Dataset description

The dataset is a monthly US macro financial panel constructed from official OECD and Federal Reserve sources. Labour market indicators, including employment, unemployment, participation and working age population, are taken from the OECD infra-annual labour statistics, while business tendency balances and composite confidence measures for firms and households follow the OECD Composite Leading Indicators framework and related short term indicator collections (OECD, 2025a, OECD, 2025b). Price, trade and monetary variables draw on OECD consumer price indices for G20 economies, international merchandise trade statistics and short term monetary and real activity series that underpin the Key Short Term Economic Indicators (OECD, 2025c, OECD, 2025d).

Recession timing and financial conditions are added from composite indicators disseminated via the Federal Reserve Bank of St Louis. The crisis block combines the Real Time Sahm Rule Recession Indicator with the NBER based US recession dummy, while broader activity is summarised by the Chicago Fed National Activity Index, its diffusion index, moving average and expenditure and production subindexes (Federal Reserve Bank of St Louis, 2025a, Federal Reserve Bank of St Louis, 2025b, Federal Reserve Bank of St Louis, 2025c, Federal Reserve Bank of Chicago, 2025a). Financial stress is measured by the National Financial Conditions Index and its adjusted version, and overall real activity is cross checked using the Coincident Economic Activity Index for the United States from the Federal Reserve Bank of Philadelphia (Federal Reserve Bank of Chicago, 2025b, Federal Reserve Bank of Philadelphia, 2025).

4.1.2 Data processing

The raw dataset combines monthly macro-financial indicators with a binary recession flag. All series are coerced to a common monthly calendar keyed by a TIME_PERIOD index, sorted chronologically, and trimmed to the overlapping sample. When indicators are available at higher than monthly frequency, they are aggregated to monthly Values by taking simple averages within each calendar month before merging. This produces a single, time-ordered multivariate panel where each row corresponds to one month.

The binary recession flag is first converted into three auxiliary series start_onset, end_offset and state. The start_onset indicator marks months in which the economy switches from expansion to recession, end_offset marks the end of a recession spell, and state encodes the contemporaneous regime (expansion or recession). These derived variables are then used to identify contiguous recession segments and the corresponding non-recession intervals. A short post-recession buffer is appended after each recession end to represent a fragile recovery phase; months in this buffer are temporarily excluded from the set of candidates for new recession onsets.

On this basis, hazard-style labels and risk sets are constructed for each forecast horizon H. For every month t in expansion and outside the post-recession buffer, a horizon window (t, t+H] is scanned for the presence of a start_onset. If at least one start_onset occurs in that window, the hazard label is set to one; otherwise it is set to zero. Months that are in recession or in the post-recession buffer do not enter the estimation sample for that

horizon. This yields, for each H, an expansion-only risk set with a corresponding binary hazard series.

Lead time and lead-time weights are precomputed together with the labels. For each horizon H and each month t in the risk set with $Y_t^H = 1$, the algorithm records the first onset month within (t, t+H] and defines the lead time $\Delta_t^H = u^*(t) - t$. On the basis of this lead time, a scalar weight is attached to each positive observation in order to favour earlier, more operationally useful warnings over very late signals that occur just before the onset. The positive weight is defined as $v(\Delta_t^H) = 1 + \alpha \frac{H-\Delta_t^H}{H}$, with a tuning parameter. Observations that are closer to the onset (small Δ_t^H) receive larger weights, while observations that are only marginally ahead of the onset receive weights close to one. Negative observations keep a weight of one. These weights are later combined with standard class weights in the loss function.

Feature processing is applied to the time-ordered macro-financial panel before sequence construction. Rolling means, standard deviations and simple linear trends are computed over medium- and long-horizon windows using only current and past Values. Structural-break indicators and similarity scores to selected historical prototype episodes are added, with changepoint models always fitted on training periods only in cross-Validation. Network-based features summarising conditional dependence and stability across indicators are derived from sparse graphical and vector autoregressive models estimated on the training slices of each fold; their outputs are aligned back to the monthly grid and carried forward when necessary. All feature blocks are merged into a common design matrix, sorted by time and variable name, and missing entries introduced by the joins are handled by forward filling along the time axis followed by zero filling if needed. This preserves temporal ordering and yields a numerically complete feature set.

For classical models, the hazard labels and features at each month in the risk set are combined into cross-sectional design matrices indexed by horizon. For the Bi-LSTM, the same standardised features are converted into sequences using fixed-length sliding windows. Given a chosen sequence length L, the data are reshaped into overlapping blocks [t-L+1,...,t] paired with the hazard label and lead-time weight at time t. Windows that would cross the boundary between training, Validation and holdout segments, or that extend beyond the available sample, are discarded. All preprocessing steps, including aggregation, label construction, feature engineering, scaling and sequence generation, are performed separately within each training partition in cross-Validation and then applied forward to the corresponding Validation and holdout segments, ensuring that no information from the future enters the inputs or labels used for model estimation.

4.2 Model specification and preparation

As part of data processing, we define and prepare baseline learners on structured tabular features. We configure three families of models: regularized logistic regressions to control variance and mitigate multicollinearity, gradient-boosted decision trees (LightGBM and XGBoost) to capture nonlinearities and interaction effects in tabular predictors, and a bidirectional LSTM to exploit short-horizon temporal structure created by rolling-window feature construction. This staged setup converts cleaned and scaled panels into model-ready matrices and sequences and keeps model specification aligned with the crisis labeling scheme and evaluation design.

In the regularized logistic specifications, we estimate L1, L2, and elastic-net variants, shrinking unstable coefficients and improving generalization in the presence of correlated macro-financial indicators; penalty type and strength are chosen by cross-validated hyperparameter search using the validation objective, in line with recent applications of penalized logit early-warning models for financial crises (Liu, Chen, and Wang, 2022). For the tree-based models, we train shallow, heavily regularized gradient-boosting ensembles on bootstrapped samples with feature subsampling, which allows the models to approximate nonlinear decision boundaries while remaining robust in small-sample, imbalanced crisis settings where similar methods have been shown to perform well relative to logit benchmarks (Liu, Chen, and Wang, 2022). For deep sequence modeling, we transform aligned panels into fixed-length sequences and feed them to a single-layer bidirectional LSTM; the forward and backward hidden states are concatenated and passed through a hazard-conditioned attention and output layer, an architecture choice motivated by evidence that recurrent networks can capture regime shifts and nonlinear dynamics in financial-stability applications when combined with careful regularization and early-warning evaluation procedures (Tölö, 2020).

5 Results

The models were eValuated on three prediction horizons of six, twelve and eighteen months using expanding-window cross Validation with separate holdout periods, and the results are summarised in the per-horizon tables and figures. Across all horizons, the regularised logistic regressions delivered the most stable and accurate out-of-sample performance, while tree-based machine learning models performed competitively at longer horizons and the Bi-LSTM showed much weaker generalisation despite high in-sample fit. For the six-month horizon, LOGISTIC (L1) reached a holdout ROC-AUC of 0.91 with precision-recall AUC 0.64, accuracy 0.83, F1 0.48 and Cohen's kappa 0.41, outperforming LOGISTIC (L2) which achieved a holdout ROC-AUC of 0.90 and PR-AUC 0.51 with accuracy 0.80 and kappa 0.32. The elastic net variant obtained high holdout accuracy of 0.91 but did so with F1 equal to zero and negative kappa, indicating that it almost never predicted the crisis class in the holdout window. Among machine learning models at this horizon, LIGHTGBM and XGBOOST produced much lower holdout ROC-AUC Values of 0.61 and 0.79 respectively and F1 scores of 0.14 and 0.21, while the stacking ensemble achieved high

training ROC-AUC above 0.92 but a holdout ROC-AUC of only 0.36. The Bi-LSTM displayed similar behaviour, with training ROC-AUC of 0.96 but a holdout ROC-AUC of 0.64, PR-AUC 0.11 and F1 0.08, even though Validation ROC-AUC remained above 0.91 during cross Validation.

For the twelve-month horizon, performance again favoured the regularised logistic models, with the elastic net specification emerging as the strongest overall. LOGISTIC (Elastic Net) reached holdout ROC-AUC of 0.94 and PR-AUC of 0.59 with accuracy 0.89, F1 0.74 and kappa 0.68, while LOGISTIC (L2) attained holdout ROC-AUC of 0.84, PR-AUC 0.37, accuracy 0.80 and F1 0.52. LIGHTGBM matched these results closely with holdout ROC-AUC 0.87, PR-AUC 0.55, accuracy 0.87 and F1 0.58, whereas XGBOOST yielded a similar accuracy of 0.89 but lower discrimination with ROC-AUC 0.69. The stacking ensemble again showed a large gap between training and holdout performance, with training ROC-AUC 0.92 but holdout ROC-AUC 0.51 and PR-AUC 0.15. The Bi-LSTM exhibited the sharpest divergence between in-sample and out-of-sample performance at this horizon: training ROC-AUC reached 0.92 and PR-AUC 0.87, while the holdout ROC-AUC fell to 0.32, PR-AUC to 0.12, and F1 to zero with negative kappa.

At the eighteen-month horizon, predictive accuracy increased further for the best traditional models. LOGISTIC (L2) delivered the strongest overall results with holdout ROC-AUC 0.99, PR-AUC 0.98, accuracy 0.93, F1 0.86 and kappa 0.82. LIGHTGBM achieved very similar performance with holdout ROC-AUC 0.95, PR-AUC 0.74, accuracy 0.93, F1 0.87 and kappa 0.82, while LOGISTIC (Elastic Net) also performed well with holdout ROC-AUC 0.93 and PR-AUC 0.73. LOGISTIC (L1) produced a holdout ROC-AUC of 0.91 and PR-AUC 0.79, but with lower F1 of 0.40 due to more conservative classification of crises. XGBOOST lagged behind with holdout ROC-AUC only 0.71 and F1 0.24, and the stacking ensemble again underperformed with holdout ROC-AUC 0.97 but accuracy 0.24 and kappa effectively zero, reflecting extreme imbalance in predicted labels. The Bi-LSTM improved relative to shorter horizons with holdout ROC-AUC 0.66, PR-AUC 0.64, accuracy 0.87, F1 0.62 and kappa 0.55, but still did not match the best logistic or tree-based models, even though its training ROC-AUC and PR-AUC were both around 0.84 and higher.

Model CV and holdout performance across horizons

Horizon=6		**Accuracy**	**PR-AUC**	**ROC-AUC**	**F1**	**Kappa**
LOGISTIC (L1)	Train	0.54	0.36	0.69	0.48	0.26
	Val	0.26	0.34	0.68	0.22	0.06
	Holdout	0.83	**0.64**	**0.91**	**0.48**	**0.41**
LOGISTIC (L2)	Train	0.92	0.77	0.90	0.73	0.69
	Val	0.59	**0.93**	**0.99**	0.31	0.21
	Holdout	0.80	0.51	0.90	0.40	0.32
LOGISTIC (Elastic Net)	Train	0.90	0.75	0.90	0.71	0.65
	Val	**0.83**	0.68	0.90	0.35	**0.28**
	Holdout	**0.91**	0.19	0.78	0.00	-0.01
LIGHTGBM	Train	0.35	0.22	0.58	0.34	0.08
	Val	0.33	0.26	0.62	**0.36**	0.20
	Holdout	0.67	0.12	0.61	0.14	0.02
XGBOOST	Train	**0.97**	**0.87**	**0.98**	**0.89**	**0.87**
	Val	0.53	0.55	0.85	0.34	0.21
	Holdout	0.69	0.35	0.79	0.21	0.09
STACKING	Train	0.92	0.80	0.93	0.77	0.72
	Val	0.60	0.55	0.92	0.28	0.18
	Holdout	0.87	0.07	0.36	0.00	-0.06
Bi-LSTM	Train	0.93	0.82	0.96	0.86	0.82
	Val	0.64	0.71	0.91	0.34	0.25
	Holdout	0.69	0.11	0.64	0.08	-0.05

Horizon=12		**Accuracy**	**PR-AUC**	**ROC-AUC**	**F1**	**Kappa**
LOGISTIC (L1)	Train	0.67	0.50	0.73	0.65	0.37
	Val	0.19	0.52	0.64	0.31	0.00
	Holdout	0.83	0.36	0.84	0.00	-0.01
LOGISTIC (L2)	Train	0.87	0.79	0.90	0.75	0.66
	Val	0.40	0.65	0.80	0.38	0.12
	Holdout	0.80	0.37	0.84	0.52	0.40
LOGISTIC (Elastic Net)	Train	0.84	0.77	0.86	0.72	0.61
	Val	0.30	**0.78**	**0.94**	0.34	0.04
	Holdout	**0.89**	**0.59**	**0.94**	**0.74**	**0.68**
LIGHTGBM	Train	0.43	0.33	0.55	0.49	0.09
	Val	0.40	0.33	0.59	0.45	0.18
	Holdout	0.87	0.55	0.87	0.58	0.50
XGBOOST	Train	**0.92**	**0.87**	**0.95**	**0.88**	**0.82**
	Val	**0.73**	0.72	0.92	**0.64**	**0.48**
	Holdout	**0.89**	0.53	0.69	0.45	0.41
STACKING	Train	0.90	0.83	0.92	0.82	0.75
	Val	0.38	0.49	0.80	0.39	0.12
	Holdout	0.65	0.15	0.51	0.16	-0.05
Bi-LSTM	Train	0.90	**0.87**	0.92	0.84	0.77
	Val	0.37	0.73	0.88	0.38	0.13
	Holdout	0.58	0.12	0.32	0.00	-0.25

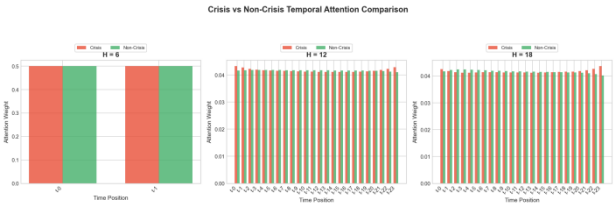
Horizon=18		**Accuracy**	**PR-AUC**	**ROC-AUC**	**F1**	**Kappa**
LOGISTIC (L1)	Train	0.55	0.53	0.58	0.66	0.17
	Val	0.38	0.46	0.60	0.45	0.05
	Holdout	0.82	0.79	0.91	0.40	0.34
LOGISTIC (L2)	Train	0.84	0.80	0.88	0.83	0.68
	Val	0.33	0.45	0.68	0.36	-0.07
	Holdout	0.93	**0.98**	**0.99**	0.86	**0.82**
LOGISTIC (Elastic Net)	Train	0.77	0.76	0.81	0.77	0.56
	Val	0.30	0.48	0.63	0.36	-0.09

	Holdout	0.87	0.73	0.93	0.73	0.64
LIGHTGBM	Train	0.64	0.59	0.67	0.70	0.30
	Val	0.37	0.39	0.66	0.45	0.05
	Holdout	**0.93**	0.74	0.95	**0.87**	0.82
XGBOOST	Train	0.82	0.83	0.85	0.85	0.61
	Val	**0.46**	0.62	0.78	0.35	0.01
	Holdout	0.79	0.65	0.71	0.24	0.20
STACKING	Train	**0.87**	**0.90**	**0.92**	**0.85**	**0.75**
	Val	0.44	0.30	0.57	0.31	0.09
	Holdout	0.24	0.88	0.97	0.39	0.00
Bi-LSTM	Train	0.83	0.83	0.84	0.84	0.61
	Val	0.42	**0.70**	**0.87**	**0.47**	**0.09**
	Holdout	0.87	0.64	0.66	0.62	0.55

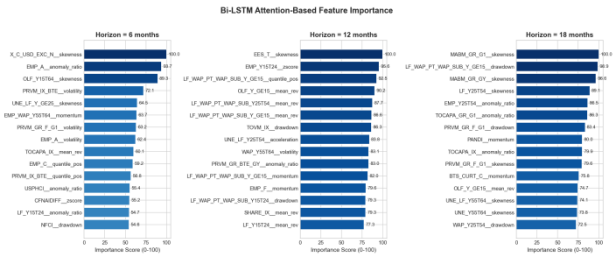
The Bi-LSTM attention-based feature importance analysis offers a complementary view of which indicators matter most when the deep learning model is used. At the six month horizon, the network placed its highest scaled weights on X_C_USD_EXC_N__skewness, EMP_A__anomaly_ratio, OLF_Y15T64__skewness and PRVM_IX_BTE__volatility, together with UNE_LF_Y_GE25__skewness, EMP_WAP_Y55T64__momentum, EMP_A__volatility, TOCAPA_IX__mean_rev and PRVM_GR_F_G1__volatility, and also on higher order moments of labour market variables and financial stress indices such as NFCI__drawdown and USPHCI__anomaly_ratio. At the twelve-month horizon, attention concentrated on EES_T__skewness, EMP_Y15T24__zscore, LF_WAP_PT_WAP_SUB_Y_GE15__quantile_pos, OLF_Y_GE15__mean_rev and LF_WAP_PT_WAP_SUB_Y25T54__mean_rev, as well as drawdowns and momentum of order book and wage indicators, accelerations in labour force participation and volatility of wage and equity indices. For the eighteen-month horizon, the most important features were MABM_GR_G1__skewness, LF_WAP_PT_WAP_SUB_Y_GE15__drawdown, MABM_GR_GY__skewness, LF_Y25T54__skewness, EMP_Y25T54__anomaly_ratio and TOCAPA_GR_G1__anomaly_ratio, along with PRVM_GR_F_G1__drawdown, PANDI__momentum, TOCAPA_IX__anomaly_ratio and several additional skewness and drawdown measures. These rankings indicate that the Bi-LSTM primarily exploits higher order moments, anomaly ratios, quantile positions and drawdowns of exchange rate, money, labour market and order book indicators rather than the volatility and momentum measures emphasised by the best logistic and tree models.

Temporal attention patterns from the Bi-LSTM further clarify how the deep learning model weights information across the two year input window. For the six month horizon, the learned attention vector is extremely concentrated on the two most recent months, with weights of approximately 0.50 at positions t-1 and t-0 for both crisis and non crisis sequences and differences across regimes of only about three ten thousandths. For the twelve- and eighteen-month horizons, the attention weights are nearly uniform across all twenty-four lags, ranging from about 0.041 to 0.043 for each time step. The crisis versus non crisis comparison shows that differences in attention between the two regimes are very small at every lag, mostly on the order of a few thousandths.

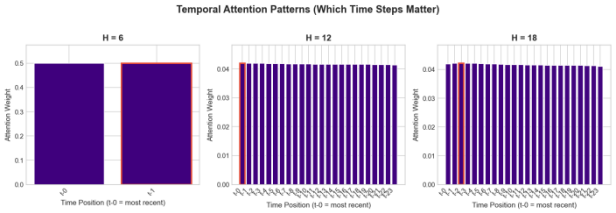
Crisis vs Non-Crisis Temporal Attention Comparison



Bi-LSTM Attention-Based Feature Importance



Temporal Attention Patterns



6 Discussion

The surprisingly strong performance of regularized logistic regression relative to complex models in predicting financial crises can be attributed to several factors. First, the dataset for crisis prediction is relatively small and imbalanced: crises are rare events, and even with many indicators, the number of positive instances is very limited. In such settings, high-capacity models like gradient-boosting trees or Bi-LSTM networks are prone to overfitting, learning spurious patterns that do not generalize. The logistic models, especially with L1/L2 penalties, effectively constrained the feature set and model complexity, focusing on a few salient indicators that consistently signal crises. This pattern stands in contrast to the early-warning results of Holopainen and Sarlin (2017), who show that conventional statistical models are generally outperformed by more advanced machine-learning and ensemble methods in a European setting, suggesting that the relative advantage of complex models is highly context dependent and may disappear when the sample is small and crises are rare. Our findings are, however, consistent with evidence from the forecasting

literature that greater model complexity does not systematically improve accuracy and can even increase forecast error, so that simple, evidence-based procedures frequently dominate more elaborate approaches (Green and Armstrong, 2015).

Feature engineering and domain knowledge further help explain why traditional models were so competitive. The inputs were derived from macro-financial indicators that were transformed into volatility, momentum, skewness, drawdowns, and related statistics in order to highlight stress dynamics. A regularized linear model can exploit these engineered features effectively. In principle, a deep learning model could discover additional nonlinear interactions or subtle temporal patterns across these same features, yet in practice the Bi-LSTM did not add predictive value beyond what the engineered features already provide. The failure of the Bi-LSTM to outperform the logit suggests that either nonlinearities and temporal dependencies are not very strong in these data or that they cannot be reliably learned given the limited number of crisis episodes. Traditional early-warning models often assume mostly linear or threshold effects, for example that credit growth beyond a critical threshold sharply raises crisis risk, and our findings indicate that the relationship between the transformed indicators and crisis probabilities is largely monotonic and additive, which a logistic regression can capture. By contrast, using a multi-country dataset and a broader set of algorithms, Liu, Chen, and Wang (2022) find that machine-learning models—particularly random forests, gradient boosting decision trees, and ensembles—deliver substantially better out-of-sample crisis forecasts than a logistic model and argue that such methods should play a central role in early-warning system design.

The performance gap between traditional and deep models also reflects differences in stability and generalization. Logistic regression coefficients were comparatively stable across cross-validation folds, while the Bi-LSTM and ensemble methods showed high variance, with small changes in the training set sometimes leading to markedly different predictions. For policy applications, an early-warning system needs not only good accuracy but also consistent behavior and clear interpretation (Tölö 2020). The black-box nature of the Bi-LSTM, together with its instability, makes it difficult for policymakers to understand why a given warning is issued or to trust that the signal will be reliable in new environments. By contrast, logistic regression provides straightforward marginal effects, such as the effect of a one-unit increase in unemployment volatility on crisis odds, which supports communication and helps connect the model's behavior to economic mechanisms. Interpretability has become a central concern in applying machine learning to financial stability analysis, and even when complex models improve raw predictive power regulators increasingly emphasize the need to understand why a model flags elevated risk (Bluwstein et al. 2021). In our setting, the top features in the best logistic specification correspond to intuitive economic phenomena such as labor-market disruptions, credit boom–bust dynamics, and financial-market stress, whereas the deep model's inferred interactions did not translate into clearer or more accurate warnings. While some studies report that neural networks and ensembles can outperform logistic early-warning models under cross-validation, for example Tölö (2020) and Liu, Chen, and Wang (2022), those gains can disappear in truly out-of-sample tests and often come at the cost of interpretability. Our results underline that robustness and transparency are critical design criteria in crisis prediction and in many cases may be more important than squeezing out marginal in-sample performance gains.

The feature rankings also provide economic insight into the precursors of financial crises in an open-data framework. High importance scores for unemployment and labor force indicators, particularly for youth and prime-age groups, indicate that labor market deterioration tends to precede and accompany financial instability. Rising unemployment and falling participation can trigger loan defaults and weaken household demand, which in turn strains bank balance sheets. Similarly, the frequent appearance of credit and leverage indicators, especially measures related to private credit growth, volatility, and mean-reversion, is consistent with the view that rapid credit expansion followed by unstable credit dynamics signals a boom–bust cycle that often culminates in crisis. Price dynamics and term-structure variables also play a central role: inflation momentum and interest-rate spreads enter prominently in the feature rankings, and a sharp pickup in inflation or inflation volatility can trigger tighter monetary policy, higher short-term rates, and a flat or inverted yield curve, conditions that frequently precede recessions and banking-sector stress. The yield curve is a particularly informative recession indicator, and an inversion where short-term rates exceed long-term rates often signals tightening financial conditions and an impending contraction of credit. That these core predictors all come from public sources such as central bank statistics, FRED, and OECD releases underscores that an effective crisis-warning system can be built without proprietary data and that transparency and reproducibility can be maintained alongside strong predictive performance. These interpretations are consistent with recent early-warning evidence based on large macro-financial panels, which finds that credit conditions, asset-price dynamics, and broader macro indicators jointly deliver useful advance signals of financial crises when modeled in flexible forecasting frameworks (Tölö 2020; Liu, Chen, and Wang 2022).

The results also reveal limitations and point to avenues for further work. Class imbalance implies that even high AUC values may coexist with missed crises or false alarms, and at the 6-month horizon the best logistic model attains a holdout F1 of about 0.48, which leaves scope to refine the trade-off between precision and recall. Deep learning was introduced to capture potential nonlinear precursors, for example combinations of rapid credit growth, an increasingly flat yield curve, and rising unemployment that might jointly matter even if each component is only mildly abnormal. That strategy did not improve forecasting accuracy in this dataset, yet it suggests a direction for future research. With longer samples, richer cross-country panels, or more stringent

regularization, such as Bayesian neural networks or transfer learning from simulated macro-financial scenarios, deep architectures might still uncover interactions that are not well captured by linear models. Horizon-specific behavior is another consideration. At the 18-month horizon even the Bi-LSTM performs more competitively, which may reflect that the buildup of vulnerabilities typically becomes clearer a year or more before crises, when credit and asset-price booms peak and macro-financial imbalances are most visible. Shorter horizons, such as 6 months, may be dominated by transitory disturbances that are harder to distinguish from noise. While the regularized logistic model delivers robust performance across horizons, different lead times may benefit from horizon-specific models or ensembles that emphasize indicators whose predictive content is strongest for that horizon.

7 Conclusion

Overall, the evidence from this study points to a central role for simple, interpretable models in financial-crisis forecasting, with machine-learning and deep-learning methods used selectively and only after careful assessment of their stability and incremental value. Regularized logistic regression provides a clear benchmark in this setting, delivering strong and transparent out-of-sample performance despite the small number of crisis observations, while more complex models struggle to convert their in-sample flexibility into reliable forecasts. At the same time, recent multi-country work shows that tree-based and ensemble methods can outperform logistic specifications when richer data are available (Liu, Chen, and Wang 2022), and applications of recurrent networks to financial stability emphasize the need for extensive robustness checks before using such models for policy (Tölö 2020). Future research can build on these results by extending the panel across countries and time, refining the set of open macro-financial indicators, and applying flexible algorithms within a framework that preserves regularization and interpretability, so that any gains from complexity are grounded in data and can be scrutinized by policymakers and practitioners.

Author statement

Dongsheng Bei: Conceptualization, Data curation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Supervision.

Pengwei Zhu: Methodology, Software, Validation, Resources, Visualization, Writing – review & editing.

Equal Contribution Statement: Dongsheng Bei and Pengwei Zhu contributed equally to this work and share co-first authorship.

References

[1] Allaj, Erindi, and Simona Sanfelici. 2023. "Early Warning Systems for Identifying Financial Instability." International Journal of Forecasting 39 (4): 1777-1803.

[2] Mugrabi, Farah, Mohamed Belkhir, Sami Ben Naceur, Bertrand Candelon, and Woon Gyu Choi. 2025. “Macroprudential Policy and Bank Systemic Risk: Does Inflation Targeting Matter?” Emerging Markets Review, article 101397.

[3] Chen, Sally, and Katsiaryna Svirydzenka. 2021. "Financial Cycles-Early Warning Indicators of Banking Crises?" IMF Working Paper 2021/116. Washington, DC: International Monetary Fund.

[4] Greenwood, Robin, Samuel G. Hanson, Andrei Shleifer, and Jakob Ahm Sørensen. 2022. "Predictable Financial Crises." Journal of Finance 77 (2): 863-921.

[5] Koponen, Heidi. 2024. "Constructing a Composite Indicator to Assess Cyclical Systemic Risks: An Early Warning Approach." BoF Economics Review 3/2024. Helsinki: Bank of Finland.

[6] Liu, Lanbiao, Chen Chen, and Bo Wang. 2022. "Predicting Financial Crises with Machine Learning Methods." Journal of Forecasting 41 (5): 871-910.

[7] Škrinjarić, Tihana. 2023. "Introducing a Composite Indicator of Cyclical Systemic Risk in Croatia: Possibilities and Limitations." Public Sector Economics 47 (1): 1-39.

[8] Purnell, Daren, Jr., Amir Etemadi, and John Kamp. 2024. “Developing an Early Warning System for Financial Networks: An Explainable Machine Learning Approach.” Entropy 26 (9): 796.

[9] Ouyang, Zi-sheng, Xi-te Yang, and Yongzeng Lai. 2021. "Systemic Financial Risk Early Warning of Financial Market in China Using Attention-LSTM Model." The North American Journal of Economics and Finance 56: 101383.

[10] Candelon, Bertrand, Elena-Ivona Dumitrescu, and Christophe Hurlin. 2012. “How to EValuate an Early-Warning System: Toward a Unified Statistical Framework for Assessing Financial Crises Forecasting Methods.” IMF Economic Review 60 (1): 75–113.

[11] Drehmann, Mathias, and Mikael Juselius. 2014. “EValuating Early Warning Indicators of Banking Crises: Satisfying Policy Requirements.” International Journal of Forecasting 30 (3): 759–780.

[12] Firdaus, N. T., and N. Santoso. 2025. “Early Warning Systems for Financial Crisis Prediction: A Systematic Literature Review of Econometrics, Machine Learning and Uncertainty Indices.” MALCOM: Indonesian Journal of Machine Learning and Computer Science 5 (4): 1415–1422.

[13] Gu, Xing. 2022. Early-Warning Alert Systems for Financial-Instability Detection: An HMM-Driven Approach. PhD dissertation, University of Western Ontario.

[14] Hidayat, Taufiq, Dian Masyita, Sulaeman Rahman Nidar, Fauzan Ahmad, and Muhammad Adrissa Nur Syarif. 2022. “Early Warning Early Action for the Banking Solvency Risk in the COVID-19 Pandemic Era: A Case Study of Indonesia.” Economies 10 (1): 6.

[15] Namaki, Ali, Reza Eyvazloo, and Shahin Ramtinnia. 2023. “A Systematic Review of Early Warning

Systems in Finance." arXiv preprint arXiv:2310.00490.

[16] Yildirim, Yusuf, and Anirban Sanyal. 2022. "EValuating the Effectiveness of Early Warning Indicators: An Application of Receiver Operating Characteristic Curve Approach to Panel Data." Scientific Annals of Economics and Business 69 (4): 557–597.

[17] Yang, Hufang, Luyi Liu, Jieyang Cui, Wenbin Wu, and Yuyang Gao. 2025. "Research on Dynamic Measurement and Early Warning of Systemic Financial Risk in China Based on TVP-FAVAR and Deep Learning Model." Systems 13 (8): 720.

[18] Elnaggar, Hoda A., Marwa Elsherif, and Mohamed I. Marie. 2025. "A Deep Learning-Based Model for Financial Crisis Prediction." Research Square preprint, version 1.

[19] Boulkroune, Abdesselem, Sarah Hamel, Farouk Zouari, Abdelkrim Boukabou, and Asier Ibeas. 2017. "Output-Feedback Controller Based Projective Lag-Synchronization of Uncertain Chaotic Systems in the Presence of Input Nonlinearities." Mathematical Problems in Engineering 2017: 8045803.

[20] Boulkroune, Abdesselem, Farouk Zouari, and Amina Boubellouta. 2025. "Adaptive Fuzzy Control for Practical Fixed-Time Synchronization of Fractional-Order Chaotic Systems." Journal of Vibration and Control.

[21] Zouari, Farouk, Kamel Ben Saad, and Mohamed Benrejeb. 2012. "Robust Neural Adaptive Control for a Class of Uncertain Nonlinear Complex Dynamical Multivariable Systems." International Review on Modelling and Simulations 5 (5): 2075–2103.

[22] Zouari, Farouk, Kamel Ben Saad, and Mohamed Benrejeb. 2013. "Adaptive Backstepping Control for a Class of Uncertain Single Input Single Output Nonlinear Systems." In 2013 10th International Multi-Conference on Systems, Signals and Devices (SSD).

[23] Rigatos, Gerasimos, Masoud Abbaszadeh, Bilal Sari, Pierluigi Siano, Gennaro Cuccurullo, and Farouk Zouari. 2023. "Nonlinear Optimal Control for a Gas Compressor Driven by an Induction Motor." Results in Control and Optimization 11: 100226.

[24] Merazka, Loubna, Farouk Zouari, and Abdesselem Boulkroune. 2017. "High-Gain Observer-Based Adaptive Fuzzy Control for a Class of Multivariable Nonlinear Systems." In 2017 6th International Conference on Systems and Control (ICoSC), 96–102.

[25] European Central Bank. 2022. "Decrypting Financial Stability Risks in Crypto-Asset Markets." Financial Stability Review, May.

[26] Financial Stability Board. 2023. The Financial Stability Risks of Decentralised Finance. Basel: Financial Stability Board.

[27] Luo, Bingqiao, Zhen Zhang, Qian Wang, Anli Ke, Shengliang Lu, and Bingsheng He. 2024. "AI-Powered Fraud Detection in Decentralized Finance: A Project Life Cycle Perspective." arXiv preprint 2308.15992.

[28] Oanh, Tran Thi Kim, Le Thi Thuy Van, and Le Quoc Dinh. 2023. "Relationship between Financial Inclusion, Monetary Policy and Financial Stability: An Analysis in High Financial Development and Low Financial Development Countries." Heliyon 9 (6): e16647.

[29] Kebede, Jeleta Gezahegne, Saroja Selvanathan, and Athula Naranpanawa. 2024. "Financial Stability and Financial Inclusion: A Non-Linear Nexus." Journal of Economic Studies 52 (4): 742–761.

[30] Schmieder, Christian, and Patrick A. Imam. 2024. Aging Gracefully: Steering the Banking Sector through Demographic Shifts. BIS Working Papers No. 1193, Bank for International Settlements, 12 June.

[31] Organisation for Economic Co-operation and Development (OECD). 2025a. "Infra-Annual Labour Statistics (IALFS)." OECD Data Explorer. Accessed November 19, 2025. https://data-explorer.oecd.org/.

[32] Organisation for Economic Co-operation and Development (OECD). 2025b. "Composite Leading Indicators (CLI)." OECD Data and Datasets. Accessed November 19, 2025. https://www.oecd.org/en/data/datasets/oecd-composite-leading-indicators-clis.html.

[33] Organisation for Economic Co-operation and Development (OECD). 2025c. "G20 – Consumer Price Indices, All Items." OECD Data Explorer. Accessed November 19, 2025. https://data-explorer.oecd.org/.

[34] Organisation for Economic Co-operation and Development (OECD). 2025d. "International Merchandise Trade Statistics." OECD Data Explorer. Accessed November 19, 2025. https://data-explorer.oecd.org/.

[35] Federal Reserve Bank of St. Louis. 2025a. "Real-Time Sahm Rule Recession Indicator (SAHMREALTIME)." FRED, Federal Reserve Bank of St. Louis. Accessed November 19, 2025. https://fred.stlouisfed.org/series/SAHMREALTIME.

[36] Federal Reserve Bank of St. Louis. 2025b. "NBER Based Recession Indicators for the United States from the Period following the Peak through the Trough (USREC)." FRED, Federal Reserve Bank of St. Louis. Accessed November 19, 2025. https://fred.stlouisfed.org/series/USREC.

[37] Federal Reserve Bank of St. Louis. 2025c. "Chicago Fed National Activity Index (CFNAI) and Related Series." FRED, Federal Reserve Bank of St. Louis. Accessed November 19, 2025. https://fred.stlouisfed.org/series/CFNAI.

[38] Federal Reserve Bank of Chicago. 2025a. "Chicago Fed National Activity Index: Current Data." Federal Reserve Bank of Chicago. Accessed November 19, 2025. https://www.chicagofed.org/research/data/cfnai/current-data.

[39] Federal Reserve Bank of Chicago. 2025b. “National Financial Conditions Index: About the NFCI and Current Data.” Federal Reserve Bank of Chicago. Accessed November 19, 2025. https://www.chicagofed.org/research/data/nfci/about.

[40] Federal Reserve Bank of Philadelphia. 2025. “State Coincident Indexes: Coincident Economic Activity Index for the United States (USPHCI).” Federal Reserve Bank of Philadelphia. Accessed November 19, 2025. https://fred.stlouisfed.org/series/USPHCI.

[41] Holopainen, Markus, and Peter Sarlin. 2017. “Toward Robust Early-Warning Models: A Horse Race, Ensembles and Model Uncertainty.” Quantitative Finance 17 (12): 1933–1963.

[42] Green, Kesten C., and J. Scott Armstrong. 2015. “Simple versus Complex Forecasting: The Evidence.” Journal of Business Research 68 (8): 1678–1685.

[43] Liu, Lanbiao, Chen Chen, and Bo Wang. 2022. “Predicting Financial Crises with Machine Learning Methods.” Journal of Forecasting 41 (5): 871–910.

[44] Tölö, Eero. 2020. “Predicting Systemic Financial Crises with Recurrent Neural Networks.” Journal of Financial Stability 49: 100746. https://doi.org/10.1016/j.jfs.2020.100746

[45] Bluwstein, Kristina, Marcus Buckmann, Andreas Joseph, Sujit Kapadia, and Özgür Şimşek. 2021. “Credit Growth, the Yield Curve and Financial Crisis Prediction: Evidence from a Machine Learning Approach.” ECB Working Paper 2614. Frankfurt am Main: European Central Bank.

Image Edge Detection Using FHN-CNN Model Based on Reaction-Diffusion Equations within a Dynamical Systems Framework

Zhao Yang
Academic Affairs Office, Zibo Polytechnic University, Zibo, 255314, China
E-mail: zhaoyang_zyang@outlook.com

Keywords: dynamical system, reaction-diffusion equation, image processing, FHN, Cellular-NN, zuring spot

Received: August 12, 2025

With the advancement of imaging technology, images present diverse and highly complex characteristics, and the theory of dynamical systems has great potential in image processing due to its unique mathematical properties. Therefore, this study proposes an IPT based on the reaction-diffusion equation. This technique combines cellular neural networks with the reaction-diffusion equation within the framework of dynamical system theory. Specifically, by introducing the Laplacian operator, the membrane potential in the FitzHugh-Nagumo equation is correlated with the spatiotemporal dynamics of the recovery variable. A new type of image processing model is constructed by mapping to the dynamic evolution of locally coupled mesh cells in convolutional neural networks. The theoretical framework of the technology is further improved through the dynamic analysis of the Turing instability of the model and the gradient changes of the reaction-diffusion system. The results showed that in the performance testing of the research model, the Edge Preservation Index (EPI) was 0.89 and the Pratt's Figure of Merit (PFOM) was 0.95, which were higher than the comparison models, indicating excellent model performance. Meanwhile, the time cost for the new model to complete one detection was only 0.9 seconds, and there were only 2 iterations, which was significantly better than other models. Research has shown that the new model has higher computational efficiency and better real-time performance. This study provides new ideas and methods for image processing and helps to promote the development of image processing algorithms towards high efficiency and intelligence.

Povzetek: Predlagan reakcijsko-difuzijski model obdelave slik doseže boljšo ohranitev robov in hitrejše delovanje kot primerjalne metode.

1 Introduction

Early Image Processing Techniques (IPTs) were mainly used to improve image quality, such as image denoising and enhancing image contrast. The advancement of computer technology has led IPT to gradually enter the stage of digital image processing, placing higher demands on the real-time and precision of image processing [1]. To meet the needs of digital transformation, IPT's core algorithms have transitioned from manual algorithms and simple mathematical models to Deep Learning (DL) models [2]. Numerous scholars have conducted research on IPT. Zangana et al. explored the combination of edge detection and existing image processing methods, and summarized the advantages and limitations of existing edge detection methods. Advanced methods utilizing DL, fuzzy logic, and optimization algorithms have shown good results in improving edge detection accuracy [3-4]. Chen et al. proposed a DL-based post-processing method for magnetic resonance images. The processing steps included noise reduction, image artifact correction, and image resolution improvement. This method had a significant effect on improving image quality and correcting image artifacts [5]. Zhou et al. proposed an underwater image enhancement method based on multi-interval sub-histogram perspective equalization. This method extracted statistical features of the image, estimated the degree of feature drift in each region of the image, and achieved adaptive feature enhancement. The experiment showed that this method was significantly superior to existing methods in both qualitative and quantitative aspects [6]. Zhang et al. proposed an underwater image enhancement method based on weighted wavelet visual perception fusion to address the issue of degraded underwater image quality. This method integrated high-frequency and low-frequency components of images at different scales to obtain high-quality underwater images, which were superior to the latest methods in both qualitative and quantitative aspects [7]. Jiang et al. proposed a diffusion-based low-light image enhancement method to address long processing time and high computational resource consumption in image restoration using diffusion models. It introduced a high-frequency recovery module, achieving better fine-grained recovery and significantly improving efficiency compared to previous diffusion-based methods [8]. Arshaghi et al. proposed a method based on Convolutional Neural Network (CNN) for the problem of agricultural product defect identification. They studied

the classification of five potato diseases and conducted experiments using 5,000 images. The results showed that this method outperformed other existing methods in terms of the accuracy of defect classification [9].

Reaction Diffusion Equation (RDE) is a type of partial differential equation used to describe the dynamic process of coupling diffusion and chemical reaction of matter or information in space [10]. The algorithm itself is constantly being improved and applied in various fields. Liu et al. proposed an efficient quantum algorithm for nonlinear RDE, introducing Carleman linearization method and Allen Cahn equation to extract derivative information by post-processing quantum states. The algorithm was found to achieve faster convergence speed [11]. Xue et al. constructed an RDE with non local spatial memory to simulate the cognitive movement of a single species and studied its dynamics. This study explored the combined effects of perceptual scale and memory delay on system stability and spatiotemporal dynamics through Turing Hopf bifurcation and double Hopf bifurcation. The perceptual scale and memory delay could be analyzed through dynamics [12]. Zayed et al. studied solitary wave solutions in stochastic RDEs and revealed new triangular and hyperbolic stochastic exact solutions by introducing two different integration methods and concatenating them. In numerical simulations, the newly discovered solutions using this method were effective at the application level [13]. Agresti A studied the noise regularization problem of RDE groups with quality control. Experimental results showed that moderate transmission type random multiplicative perturbations had regularization effects [14]. Hadhoud et al. proposed a shift configuration scheme for solving 2D fractional order coupled RDE. This method determined the expansion coefficients by simplifying the boundary conditions into a system of ordinary differential equations with these coefficients, transforming the 2D spatiotemporal fractional order coupled RDE into an algebraic equation system. The proposed method exhibited excellent performance and efficiency [15]. Overall, RDE can be applied in image processing, with strong potential for development in areas such as image denoising, restoration, and segmentation [16]. Barbu T proposed a new framework for the problem of automatic vehicle detection and tracking. This framework combined DL with target detection technology based on Gaussian mixture models to detect moving vehicles and determine the correspondence between video targets between consecutive frames through multi-scale analysis. [17]. Hagemann et al. proposed an infinite-dimensional diffusion model for the finite-dimensional Stochastic Banach Space Diffusion Models (SBDMs) in image generation, which models the training data over rectangular regions. This model achieved higher-resolution image generation and improved training efficiency. The results showed that this method could effectively overcome the shortcomings of existing methods and show practical advantages on synthetic Gaussian mixture samples, MNIST datasets, and datasets generated by nonlinear 2D reaction-diffusion equations. [18]. The specific comparison of the above-mentioned literature is shown in Table 1.

Table 1: Literature review table

Literature	Method	Advantages	Limitations
Literature [5]	Post-processing method for magnetic resonance images based on DL	It has a remarkable effect on improving image quality and correcting artifacts	The model decision-making mechanism is like a "black box", and its application in security-sensitive fields is limited
Literature [6-7]	An underwater image enhancement method based on the fusion of multi-interval sub-histogram perspective equalization and weighted wavelet visual perception	It is significantly superior to the existing methods in both qualitative and quantitative aspects	For specific degradation models (such as underwater), the generalization ability remains to be verified
Literature [8]	A low-light image enhancement method based on the diffusion model introduces a high-frequency recovery module	Better fine recovery has been achieved and efficiency has been significantly improved	Diffusion models generally have the problems of long processing time and high consumption of computing resources
Literature [9]	Agricultural product disease identification based on CNN	It outperforms other existing methods in terms of defect classification accuracy	Strong reliance on large-scale labeled data for training leads to high deployment costs
Literature [10]	Reaction-diffusion equation with non-local spatial memory	It can simulate cognitive movements and reveal the combined influence of perceptual scale and memory delay on system dynamics	It focuses on theoretical dynamic analysis, but the specific application paradigm in image processing is not clear
Literature [11]	The solitary wave solution of the random reaction-diffusion equation	A new stochastic exact solution was revealed, and its	The construction of solutions strongly depends on the form of

	is solved by the series integration method	application effect in numerical simulation was remarkable	the equation, and universal methods are still under exploration
Literature [12]	Study the noise regularization of reaction-diffusion equations with quality control	Experiments show that random multiplicative perturbations have a regularization effect	Theoretical research is rather abstract, and the connection with specific image processing tasks needs to be further established
Literature [16]	A shift configuration scheme for solving 2D fractional-order coupled reaction-diffusion equations	It demonstrated excellent computing performance and efficiency	The method is specifically designed for fractional calculus scenarios and does not have a direct advantage in integer-order problems
Literature [17]	An automatic vehicle detection and tracking framework combining DL with object detection technology based on Gaussian mixture model	The detection of moving vehicles was achieved, and the inter-frame target correspondence was determined through multi-scale analysis	For systems engineering, the theoretical basis and interpretability of the underlying processing units have not been deeply explored
Literature [18]	An infinite-dimensional SBDM for image generation	It can achieve higher-resolution image generation and improve training efficiency	The model is complex, and the performance improvement comes at the cost of high computational complexity

To sum up, current IPT research, especially DL methods, still faces some inherent limitations while achieving performance improvements. Firstly, there is poor physical interpretability. DL models are like "black boxes" whose decision-making mechanisms are difficult to understand, limiting their application in security-sensitive fields. Secondly, there is a strong dependence on data and computing power. High performance heavily depends on large-scale labeled data and powerful computing resources, creating a contradiction between generalization ability and deployment cost. The last issue is that it is difficult to balance computational efficiency and accuracy. Although complex network structures can improve accuracy, they often lead to high computational overhead and a large number of iterations, making it difficult to meet the needs of real-time processing. In addition, traditional reaction-diffusion models have good physical interpretability. However, its application in image processing often relies on empirical parameter adjustment, has limited ability to handle complex textures and noise, and fails to be deeply integrated with modern data processing frameworks. Given this, this study proposes an IPT based on RDE, which combines the FitzHugh-Nagumo (FHN) model with the Cellular Neural Network (Cellular-NN) to construct a new FHN-Cellular-NN image processing model. This model compensates for the shortcomings of traditional algorithms including insufficient computing power, strong data dependence, and poor real-time performance. The innovation of the research lies in the combination of Cellular-NN and FHN, which achieves a balance between efficiency, accuracy, and physical interpretability in image processing through dynamic driven feature optimization. Secondly, using dynamic analysis tools such as Turing instability analysis, waveform analysis, steady-state and stable point value analysis, FHN-Cellular-NN is deconstructed from the perspective of dynamic systems. This study helps to reveal the intrinsic evolution mechanism of the model, optimize parameter design, and enhance the physical interpretability and robustness of the algorithm.

2 Methods and materials

2.1 Construction of image processing model based on RDE

To address the limitations of existing image processing models in terms of computational efficiency, data dependency, and physical interpretability, this study aims to construct a new image edge detection model based on FHN-CNN. The study proposes a core hypothesis: By mapping the dynamic characteristics of the FHN model to the local coupling architecture of CNN and using dynamical system theory tools such as Turing instability for regulation, the constructed model can maintain high edge detection accuracy while reducing computational complexity and the number of iterations. Next, the study will elaborate in detail on the construction process, dynamic analysis, and solution methods. In digital image processing, the choice of color model directly affects the processing effect. The commonly used color models for hardware devices include Red-Green-Blue (RGB), Cyan Magenta Yellow (CMY), and Hue Saturation Value (HSV). Due to the differences in numerical representation between RGB and CMY models and human perception, while HSV models are more in line with human visual characteristics [19], it is often necessary to convert RGB models to HSV models in the image processing process of RDE. The specific processing flow is shown in Figure1. The FHN equation is selected as the core dynamic equation for constructing the proposed model. Its reaction-diffusion form is as follows: Step 1 is to define the core dynamic equation. Step 2 is the integration with Cellular-BN. Step 3 is to

construct the matrix form and boundary conditions. Step 4 is the edge detection task of the image, which can be completed by solving the above coupled ordinary differential equation system.

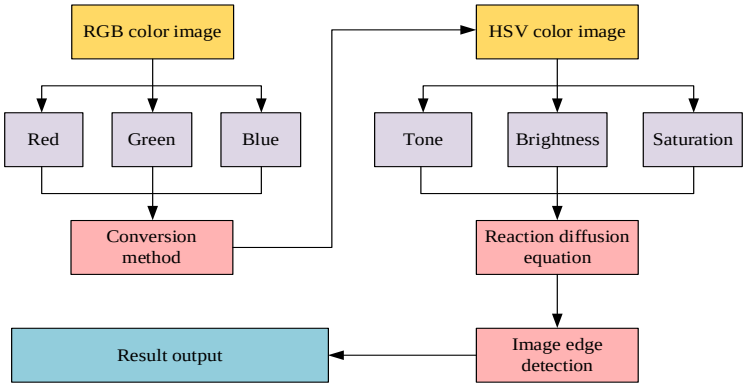


Figure1: Color model conversion

In Figure1, the RGB color image can be decomposed into pixels of the three primary colors of red, green, and blue. The three primary colors correspond to different values in their color model space, and the values of the three primary colors are mapped one-to-one to the HSV color space through conversion methods to obtain HSV color images. The three elements of hue, brightness, and saturation separated from HSV color images are the input values suitable for RDE processing. This study selects the FHN equation, which is the most suitable for image processing among RDE types, as the algorithm basis for image edge detection [20]. FHN-RDE is shown in equation (1).

$$\begin{cases} \theta_t s = f(s,q) + D_s \nabla^2 s \\ \theta_t q = g(s,q) + D_q \nabla^2 q \end{cases} \quad (1)$$

In equation (1), $\theta_t s$ is the time derivative of the membrane potential. $\theta_t q$ is the time derivative of the recovery variable. s is the neuronal membrane potential. q is the recovery variable. D_s and D_q are diffusion coefficients. ∇^2 is the Laplacian operator. The expanded form after substituting equation (1) into the function is shown in equation (2).

$$\begin{cases} \frac{\theta s}{\theta t} = \delta[D_s \nabla^2 s + s(a-s)(s-1) - q] \\ \frac{\theta q}{\theta t} = s - bq + D_q \nabla^2 q \end{cases} \quad (2)$$

In equation (2), $\frac{\theta s}{\theta t}$ is the time derivative of the membrane potential. $\frac{\theta q}{\theta t}$ is the time derivative of the recovery variable. t is time. δ is the external stimulus current. a is the activation threshold parameter. b is the attenuation coefficient of the recovery variable. The unfolded original FHN RDE is obtained through equations (1) and (2). The equation cannot be directly used for image edge detection, so this study introduces a locally interconnected nonlinear simulation dynamic system Cellular-NN to construct an image processing model based on RDE [21]. This study names the new model FHN-Cellular-NN. The Cellular-NN structure is shown in Figure2.

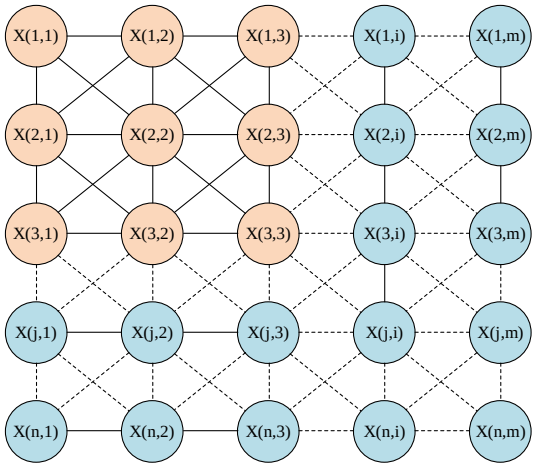


Figure 2: Cellular-NN dynamical system structure

In Figure2, Cellular-NN adopts a grid topology structure, and its core architecture consists of n rows and m columns of regularly arranged units. Each unit $X`_{ji}$ corresponds to the position of the j-th row and i-th column in the array. The connection characteristics of this network are manifested as local coupling, where each unit only establishes electrical connections with its directly adjacent units. This neighborhood interconnection mechanism significantly reduces the complexity of the system. The state function of the Cellular-NN dynamical system is shown in equation (3).

$$X`_{ji} = \sum_{X(k,l)\in V_r(j,i)} \alpha(j,i,k,l) H_1 + \sum_{X(k,l)\in V_r(j,i)} \beta(j,i,k,l) H_2 - X_{ji} \quad (3)$$

In equation (3), $X`_{ji}$, X_{ji}, and $X(k,l)$ are the updated state, current state, and neighboring units of unit

$X(j,i)\cdot V_r(j,i)$ is connected to unit $X(j,i)\cdot$ r is the adjacency radius of the unit. $\alpha(j,i,k,l)$ is the feedback matrix. $\beta(j,i,k,l)$ is the control matrix. H_1 is the output connection weight of $X(k,l)$ and $X(j,i)\cdot$ H_2 is the input connection weight for $X(k,l)$ and $X(j,i)\cdot$ The combination of FHN and Cellular-NN first introduces the Laplacian operator, as shown in equation (4).

$$\begin{aligned}\nabla^2 &= (x,y+1)+(x,y-1)+(x+1,y)\\ &\quad +(x-1,y)-4(x,y)\end{aligned} \tag{4}$$

In equation (4), x is the value on the x axis. y is the numerical value on the y axis. (x,y) represents the coordinate point. Secondly, each (x,y) two-dimensional coordinate point is considered as a unit of the Cellular-NN dynamical system, so substituting equation (4) into equation (2) yields equation (5).

$$\begin{cases} s_{j,i}` = -s_{j,i}+\delta D_s s_{j,i+1}+\delta D_s s_{j,i-1}+\delta D_s s_{j+1,i}+\delta D_s s_{j-1,i}+\\ \quad [1+\delta a-4\delta D_s+\delta(1+a)s_{j,i}-\delta s_{j,i}^2]s_{j,i}-\delta q_{j,i} \\ q_{j,i}` = -q_{j,i}+D_q q_{j,i+1}+D_q q_{j,i-1}+D_q q_{j+1,i}+\\ \quad D_q q_{j-1,i}+(1-4D_q-b)q_{j,i}+s_{j,i}\end{cases} \tag{5}$$

In equation (5), $s_{j,i}`$, $q_{j,i}`$, $s_{j,i}$, and $q_{j,i}$ represent the membrane potential time derivative, recovery variable time derivative, membrane potential, and recovery variable of the j-th row and i-th column units. If $X`_{ji}$ in equation (3) can be regarded as $s_{j,i}`$ in equation (5), then the matrix of FHN-Cellular-NN membrane potential dimension is shown in equation (6).

$$\begin{cases} \alpha_{s,s}=\begin{bmatrix} 0 & \delta D_s & 0 \\ D_s & 1+\delta a-4\delta D_s+\delta(1+a)x-\delta x^2 & \delta D_s \\ 0 & \delta D_s & 0 \end{bmatrix} \\ \alpha_{q,s}=\begin{bmatrix} 0 & 0 & 0 \\ 0 & -\delta & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \beta_{s,s}=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{cases} \tag{6}$$

In equation (6), $\alpha_{s,s}$ and $\beta_{s,s}$ are the coupling feedback matrix and coupling control matrix between membrane potential and membrane potential. $\alpha_{q,s}$ is the coupling feedback matrix of membrane potential and recovery variable. If $X`_{ji}$ in equation (3) can be regarded as $q_{j,i}`$ in equation (5), then the FHN-Cellular-NN restores the matrix of variable dimensions as shown in equation (7).

$$\begin{cases} \alpha_{q,q}=\begin{bmatrix} 0 & D_q & 0 \\ D_q & 1-4D_q-b & D_q \\ 0 & D_q & 0 \end{bmatrix} \\ \alpha_{q,s}=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \beta_{q,q}=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{cases} \tag{7}$$

In equation (7), $\alpha_{q,q}$ and $\beta_{q,q}$ are the coupling feedback matrix and coupling control matrix between two recovery variables. To eliminate the arbitrariness of FHN-Cellular-NN solutions and ensure the uniqueness of numerical solutions, this study constructs discrete boundary condition equations. The discrete boundary condition equation based on rows is shown in equation (8).

$$\begin{cases} R_{j,-1}=R_{j,0} \\ R_{j,m-1}=R_{j,m} \end{cases},\ j=0,1,...,n-1 \tag{8}$$

In equation (8), $R_{j,-1}$ and $R_{j,0}$ are the values of row j, column -1, and row j, column 0 of the matrix. $R_{j,m-1}$ and $R_{j,m}$ are the values of the j-th row, $m-1$-th column, and j-th row, m-th column of the matrix. The column based discrete boundary condition equation is shown in equation (9).

$$\begin{cases} R_{-1,i}=R_{0,i} \\ R_{n-1,i}=R_{n,i} \end{cases},i=0,1,...,m-1 \tag{9}$$

In equation (9), $R_{-1,i}$ and $R_{0,i}$ represent the values of row -1, column i, and row i of the matrix. $R_{n-1,i}$ and $R_{n,i}$ are the values of row $n-1$, column i, and row n, column i of the matrix. The final solution of FHN-Cellular-NN can be obtained by combining equations (8), (9), and (5). The discrete boundary condition adopted here mathematically corresponds to the Neumann boundary condition (zero flow condition). The physical meaning of this condition is to assume that the pixel values outside the image boundary are the same as those at the very edge, thereby numerically eliminating the false gradient at the boundary. In summary, the functional expression of FHN-Cellular-NN is obtained through equations (1) to (5), and the solution method of FHN-Cellular-NN is obtained through equations (6) to (9).

After completing the mathematical construction and discretization solution of the FHN-Cellular-NN model, a core question emerges: Does this model have the ability to generate stable and controllable patterns during the dynamic evolution process to reliably perform the image edge detection task? To solve the above problems and deeply understand the intrinsic behavior and stability of the model, the study has shifted from the perspective of

dynamical system theory to conduct an in-depth dynamic analysis of FHN-Cellular-NN. This analysis aims to provide a theoretical basis for the selection of model parameters by exploring the Turing instability, steady-state characteristics, and gradient evolution laws, and to clarify the dynamic basis for achieving spontaneous edge emergence in image processing. The study adopts the finite difference method to discretize the reaction-diffusion equation. Specifically, the spatial derivative is discretized using the second-order central difference scheme. The temporal derivative is advanced using the explicit Euler method. The choice of this explicit format is based on its simple implementation and high compatibility with the local coupling characteristics of Cellular-NNs, facilitating parallel computing on grid cells.

2.2 Dynamic analysis of image processing model based on RDE

The image processing model from a dynamic perspective actually views image processing tasks as a dynamic evolutionary process. Conducting dynamic analysis on FHN-Cellular-NN can help clarify the interaction mechanism between diffusion and reaction, and provide criteria for establishing the dynamic stability of non-equilibrium systems. This study explores the universality of non-equilibrium systems by analyzing the Turing instability of FHN-Cellular-NN [22]. The main idea is to establish a quantitative criterion for diffusion driven instability through linear stability analysis and characteristic equation solving. The characteristic equation of equation (2) at point $(0,0)$ is shown in equation (10).

$$\Omega^2+\left[a\delta+b+(D_s+D_q)\frac{c^2}{d^2}\right]\Omega +\left(a\delta+D_s\frac{c^2}{d^2}\right)\left(b+D_q\frac{c^2}{d^2}\right)+\delta=0 \tag{10}$$

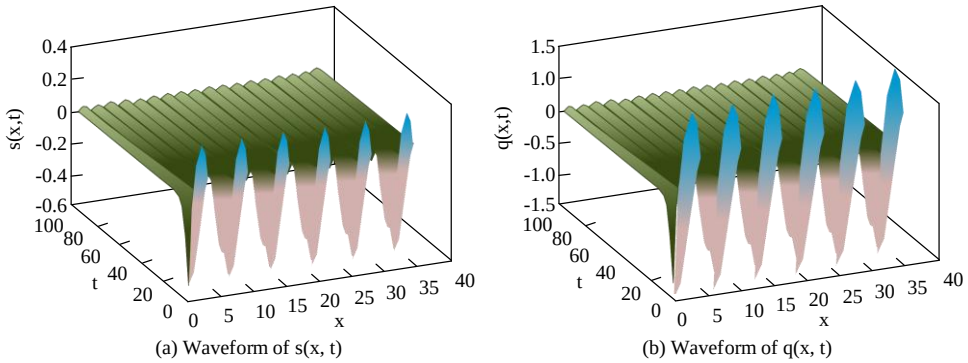
In equation (10), Ω is a constant. c and d are random variables. Assuming $c=0$, equation (10) is transformed into equation (11).

$$\Omega^2+(a\delta+b)\Omega+(ab+1)\delta=0 \tag{11}$$

In equation (11), Ω is a constant. a is the activation threshold parameter. b is the attenuation coefficient of the recovery variable. δ is the external stimulus current. When $\begin{cases}a\delta+b>0\\ab+1>0\end{cases}$, equation (11) has a root with a negative real part, and when $\begin{cases}a\delta+b>0\\ \delta=-\dfrac{b}{a}\end{cases}$, bifurcation occurs. When $c\neq 0$, equation (10) can be written as equation (12).

$$\begin{cases}\Omega^2+F_c(\delta)\Omega+D_c(\delta)=0\\ F_c(\delta)=a\delta+b+(D_s+D_q)\dfrac{c^2}{d^2}>0\\ D_c(\delta)=D_sD_q\dfrac{c^4}{d^4}+(a\delta D_q+bD_s)\dfrac{c^2}{d^2}+(ab+1)\delta=0\end{cases} \tag{12}$$

In equation (12), $F_c(\delta)$ and $D_c(\delta)$ are newly constructed functions. By analyzing the solutions of characteristic equation (10) under different parameter ranges, the parameter values for the Turing instability of FHN-Cellular-NN are determined. The determined values of the parameter group are substituted into equation (10) for solving, resulting in the waveform diagram when instability occurs, as shown in Figure3.



(a) Waveform of s(x, t) (b) Waveform of q(x, t)

Figure 3: FHN-Cellular-NN waveform diagram

Figure3 shows the waveform of membrane potential dimension $s(x,t)$ and recovery variable dimension $q(x,t)$ for $a=2$, $D_s=6$, $D_q=0.001$, and $b=-0.1$. When the above parameters are used, the FHN-Cellular-NN exhibits Turing instability, and the steady state $(0,0)$ is unstable, resulting in a new non-steady state solution. Based on the dynamic analysis of the Turing instability of FHN-Cellular-NN, this study obtains the parameter solution of the FHN-Cellular-NN

model in a stable state. Next, further analysis is conducted on the mono-stable and multi-steady states of FHN-Cellular-NN under different a and b values, as shown in Figure4.

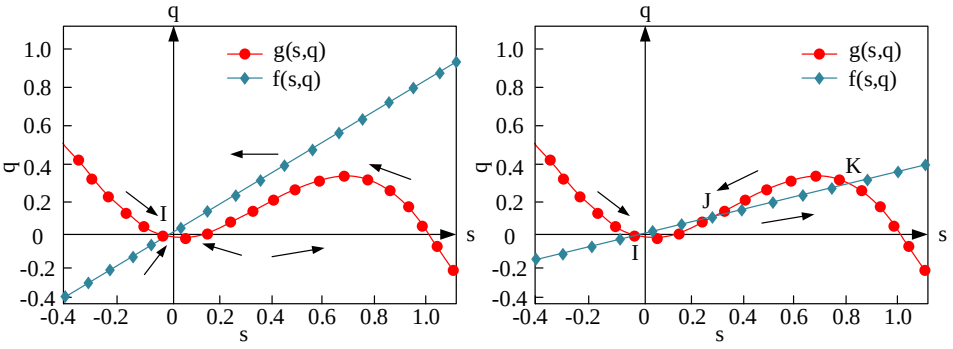


Figure 4: Analysis of FHN-Cellular-NN single and multi-steady states

Figure4 shows the analysis of the results obtained by adjusting the values of a and b to find the monostable and bistable parameter solutions for FHN-Cellular-NN. In Figure4 (a), the g(s,q) and f(s,q) functions of FHN-Cellular-NN have a stable point I, and values greater than this stable point are considered edge points. In Figure4 (b), the function has three stable points I, J, and K. That is, at these values of a and b, FHN-Cellular-NN is a multi-stable system with multiple stable points. The smaller the stable point value, the more complete the detection coverage of image edges, but the processed image noise will increase. On the contrary, the smaller the detection coverage of image edges, the finer the edge processing and the lower the noise [23]. Usually, multiple experiments are required to find a moderate value for the stable point. The mono-stable state is relatively easier to adjust because it only has one stable point. Therefore, this study selects a mono-stable system and adjusts the stable point of FHN-Cellular-NN to find its suitable value. Firstly, the stable point value is defined as ∂, and the diffusion equation of ∂ can be expressed as equation (13).

$$\partial(x) = Q(s_0(x); U, G) \tag{13}$$

In equation (13), $\partial(x)$ is the diffusion equation of ∂. U is the diffusion coefficient. G and Q are constants. $s_0(x)$ is the diffusion function for restoring the variable dimension. Equation (13) can be directly used for detecting image edges, but it still has some shortcomings. When the diffusion coefficient increases, the distance between edges will increase, resulting in blurred edges of the image. Therefore, this study further optimizes the diffusion coefficient U to avoid detecting edge blurring. The functional expression for constructing the diffusion coefficient U is shown in equation (14).

$$U = u(\varepsilon) = \frac{1}{1+(\varepsilon / k)^2} \tag{14}$$

In equation (14), $u(\varepsilon)$ is a function expression of the adjustment coefficient. ε is the normalized value of each pixel in the image. k is a constant. The acquisition of ε can be expressed as equation (15).

$$\varepsilon = |\ \nabla \mathrm{s}\ | = \sqrt{B_x^2 + B_y^2} \tag{15}$$

In equation (15), ε is the normalized value of each pixel in the image. $\nabla \mathrm{s}$ is an operator for restoring variable dimensions. B_x and B_y are specific matrices. The specific matrix of B_x and B_y is given by equation (16).

$$\begin{cases} B_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \sigma_0 \\ B_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \sigma_0 \end{cases} \tag{16}$$

In equation (16), σ_0 is the original input image. Equations (13) to (16) introduce a method of adjusting the stable point value of FHN-Cellular-NN to a moderate value, enabling FHN-Cellular-NN to have adaptive control capability for stable point values. In the process of feature extraction from digital images, region boundaries often exhibit significant gradient changes. At this point, the adjustment function will automatically converge to the minimum value in the high gradient region, achieving automatic termination of boundary positioning and ensuring accurate convergence of the contour line to the target edge position. The gradient variation of the reaction-diffusion system is shown in Figure5.

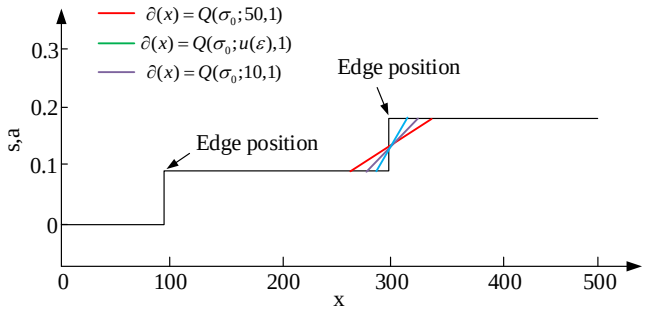


Figure 5: Gradient variation of reaction-diffusion system

Figure5 shows the gradient changes of diffusion distribution curves $\partial(x)=Q(\sigma_0;50,1)$, $\partial(x)=Q(\sigma_0;10,1)$, and $\partial(x)=Q(\sigma_0;u(\varepsilon),1)$ at parameter values $b=2$, $\varepsilon=10^3$, $D_S=0.5$, and $D_q=4$. When the stable point value used is not appropriate, the diffusion distribution curve will vibrate in two directions, producing double edges and blurring the image edges. After introducing the adjustment function, the $\partial(x)=Q(\sigma_0;u(\varepsilon),1)$ diffusion distribution curve can be accurately located at the edge of the image. This study conducts a dynamic analysis of an image processing model based on RDE, and determines the parameter values for the Turing instability of FHN-Cellular-NN by solving the characteristic equation. Based on the found parameter values, the waveform, steady-state, and stable point values of FHN-Cellular-NN are analyzed. Finally, a method is introduced to adjust the stable point value of FHN-Cellular-NN to a moderate value. Based on the above two sections, an IPT framework based on RDE is constructed from the perspective of dynamical systems. The pseudo-code of the FHN-Cellular-NN method is shown in Table 2.

Table 2: FHN-Cellular-NN pseudo-code

```
// Pseudo-code for RDE-based Image Processing Model (FHN-Cellular-NN)
function Main(image_path):
    image = LoadImage(image_path)
    hsv_image = RGBToHSV(image) // Convert RGB to HSV
    hue, saturation, brightness = ExtractHSVComponents(hsv_image)

    // FHN-RDE Computation
    membrane_potential, recovery_variable = ComputeFHN_RDE(hue, saturation, brightness)

    // Initialize and update Cellular-NN
    Cellular-NN_structure = InitializeCellular-NN()
    UpdateCellStates(Cellular-NN_structure, membrane_potential, recovery_variable)

    // Perform dynamic analysis for edge detection
    waveform = DynamicAnalysis(Cellular-NN_structure)

    // Evaluate performance
    metrics = EvaluatePerformance(waveform)
    Print("EPI: ", metrics.EPI, ", PFOM: ", metrics.PFOM)

// FHN-RDE Computation
function ComputeFHN_RDE(hue, saturation, brightness):
    for each time_step:
        for each pixel:
            UpdateMembranePotential(pixel)
            UpdateRecoveryVariable(pixel)
    return membrane_potential, recovery_variable

// Initialize Cellular-NN
function InitializeCellular-NN():
    CreateGrid(n_rows, m_columns)
    ConnectCells()
```

```
    return Cellular-NN_structure

// Update Cellular-NN cell states
function UpdateCellStates (Cellular-NN_structure, membrane_potential, recovery_variable):
    for each cell in Cellular-NN_structure:
        neighbors = GetNeighbors(cell)
        updated_state = ComputeNewState (cell, neighbors, membrane_potential, recovery_variable)
        SetCellState(cell, updated_state

// Dynamic Analysis of the model
function DynamicAnalysis (Cellular-NN_structure):
    parameters = AnalyzeParameters (Cellular-NN_structure)
    return ComputeWaveform (parameters)

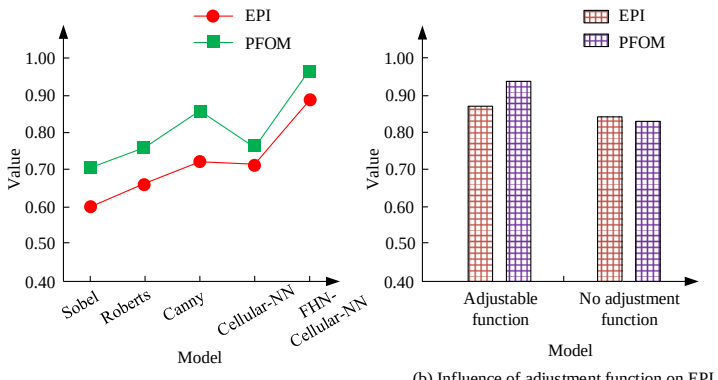
// Run the main function
Main(image_path)
```

3 Results

3.1 IPT performance testing based on RDE

After completing the theoretical construction and dynamic characteristic analysis of the FHN-Cellular-NN model, the study quantitatively evaluates its actual performance through a series of systematic experiments. The following section first presents the performance test results of this model on the standard image library, comparing it with the classical algorithm to verify its advantages in edge detection accuracy and computational efficiency. Subsequently, through numerical simulation, the convergence behavior of the model and the generation mechanism of the Turing mode are deeply revealed from the perspective of the dynamical system. Finally, through parameter sensitivity analysis, the identification and verification of the optimal model configuration are completed.

In the tested hardware environment, the CPU is Intel Core i9-13900K, the GPU is NVIDIA RTX 4090, and the memory is 64GB DDR5. The software development tool is Ubuntu 22.04 LTS system. The experimental dataset randomly selects 500 images covering natural scenes, objects, and people from the validation set of the MS COCO 2017 dataset. All input images are uniformly scaled to a resolution of 512×512 pixels. The pixel values are normalized from the integer range of 0-255 to the floating-point range of 0-1 for model processing. Baseline methods used for comparison include Sobel, Roberts, Canny and traditional CNN edge detection methods. All methods are implemented according to the standards of the OpenCV and Scikit-image libraries and uniformly use the best default parameters determined by grid search. In terms of evaluation criteria, the edge preservation index is used to quantify the similarity between the processed image and the original image in non-edge areas. The closer its value is to 1, the stronger the edge preservation ability. The Pratt quality factor comprehensively evaluates the edge detection accuracy by calculating the matching degree of position and integrity between the detected edge and the real edge, with its maximum value being 1. To verify the effectiveness of FHN-Cellular-NN, this study conducts performance tests on classic image edge detection algorithms such as Sobel, Roberts, Canny, and Cellular-NN, and compares the test results with FHN-Cellular-NN. The test mainly refers to two indicators, Pratt's Figure of Merit (PFOM), and Edge Position Inaccuracy (EPI), as shown in Figure6.

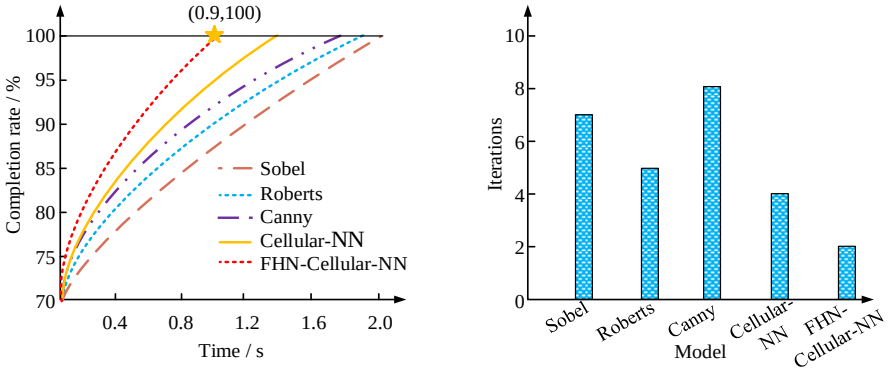


(a) EPI and pfom test results of different models

(b) Influence of adjustment function on EPI and pfom test results

Figure 6: EPI and PFOM test results

In Figure6 (a), the EPI of FHN-Cellular-NN is 0.89 and the PFOM is 0.95. Comparison shows that FHN-Cellular-NN has significantly higher indicators than other algorithms. In Figure6 (b), the EPI and PFOM with adjustment function are 0.87 and 0.94, while the EPI and PFOM without adjustment function are 0.83 and 0.81. However, the two indicators with adjustment function are higher. FHN-Cellular-NN introduces a tuning function, which outperforms the comparison model in terms of performance. To test the computational efficiency of FHN-Cellular-NN, further comparative experiments are conducted. By comparing the time cost and iteration times of different algorithms to complete a detection, the computational efficiency of different algorithms is quantitatively analyzed, as shown in Figure7.



(a) Time cost results of different models (b) Iteration results of different models

Figure 7: Results of time cost and iteration times for different models

In Figure7 (a), the time overhead for Sobel, Roberts, Canny, Cellular-NN, and FHN-Cellular-NN to complete one detection is 2 s, 1.9 s, 1.7 s, 1.4 s, and 0.9 s, with FHN-Cellular-NN having the lowest time overhead. Compared to other algorithms, FHN-Cellular-NN has significantly improved computational efficiency. In Figure7 (b), the iteration times of the five algorithms in one detection are 7, 5, 8, 4, and 2. When FHN-Cellular-NN performs computational tasks, it has fewer iterations compared to the comparison model, which better meets the real-time requirements of the algorithm. To further verify the computational efficiency of FHN-CNN from the underlying mechanism, the time complexity and space complexity of each algorithm are compared and analyzed. The results are shown in Table 3. Time complexity reflects the trend of an algorithm's execution time growing with the input scale, while space complexity reflects the algorithm's consumption of memory resources.

Table 3: Comparison of the complexity of different edge detection algorithms

Algorithm	Sobel	Roberts	Canny	CNN	FHN-CNN
Time complexity	O(n)	O(n)	O(n log n)	O(n)	O(n)
Space complexity	O(n)	O(n)	O(n)	O(n)	O(n)
Average number of iterations	49	38	56	28	14
Average single test time (s)	2	1.9	1.7	1.4	0.9

In Table 3, although Sobel, Roberts, and CNN have a linear time complexity of O(n), their number of iterations and actual running time are both higher than those of FHN-CNN. The Canny algorithm has a time complexity of O(n log n) because it includes steps such as non-maximum suppression and double-threshold connection, and its efficiency disadvantage is more obvious when dealing with large-scale images. The FHN-CNN model maintains the linear complexity, and its core advantage lies in transforming the static task of edge detection into a rapidly converging dynamic process through the dynamic characteristics of the reaction-diffusion system. In Table 3, the average number of iterations of FHN-CNN is only 14, which is significantly lower than that of other comparison models. This is attributed to the analysis of Turing instability and the adaptive control of stability points during the model's construction, enabling the system to rapidly evolve from the initial state to the target edge state. As a result, while maintaining a low computational complexity, the actual running time has been significantly reduced.

3.2 Numerical simulation based on RDE from the perspective of dynamical system

Through performance testing of FHN-Cellular-NN algorithm, it has been preliminarily confirmed that FHN-Cellular-NN has slightly improved algorithm performance compared to traditional algorithms. This study further conducts numerical simulations on FHN-Cellular-NN to analyze the long-term behavior of

the algorithm from the perspective of dynamical systems. There are two types of experiments set up for numerical simulation. One is the convergence order test of the algorithm extrapolated solutions in both spatial and temporal directions, and the other is the Turing spot generation experiment. This study sets up a numerical example to verify the convergence of discrete schemes in space and time, as shown in Figure8.

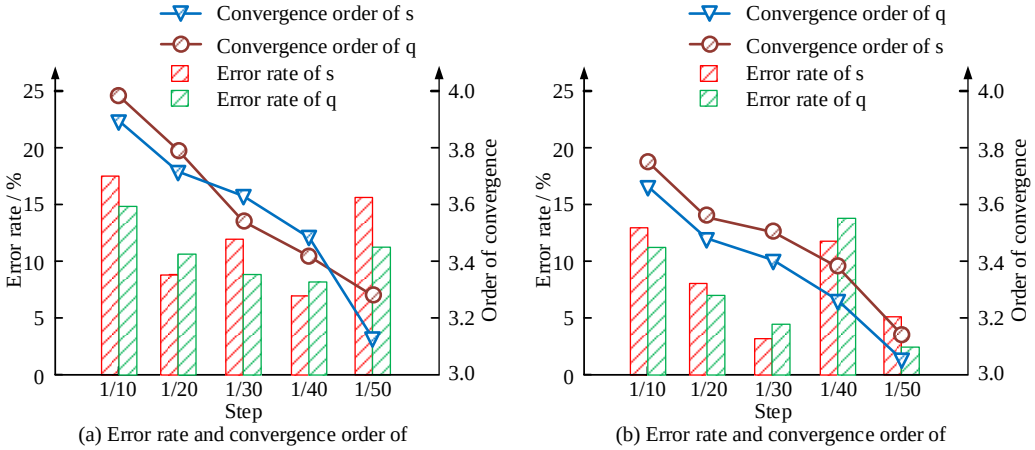


Figure 8: Error rate and convergence order of extrapolation solution in spatiotemporal direction

In Figure8 (a), the convergence order of vectors s and q in RDE shows a monotonically decreasing trend as the spatial step size decreases. In the solution of dynamical systems, monotonically decreasing convergence order means that stability is maintained throughout the entire convergence process. The error rate of vectors s and q remains below 20% and has no obvious pattern, which is a normal phenomenon. In Figure8 (b), the convergence order of vectors s and q in RDE monotonically decreases with decreasing time step. The test results in the time direction are roughly consistent with those in the spatial direction. The discrete format can achieve a very good convergence accuracy and reach the optimal convergence order. Turing pattern generation is a mathematical theoretical model that describes the spontaneous formation of periodic patterns in nature. This study optimizes the parameter configuration of the reaction-diffusion model by conducting a Turing spot generation experiment on FHN-Cellular-NN. The spatial step size is 1/50, the time step size is 1/10, and the vector s is composed of four central points in the initial state. When the parameter a in the vector s changes and b remains unchanged, the evolution process is shown in Figure9.

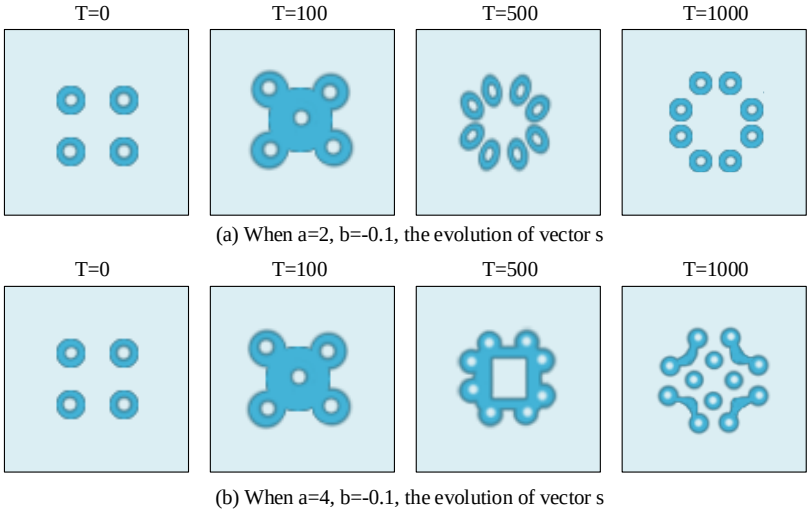
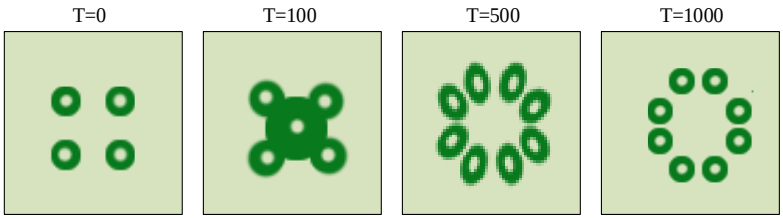


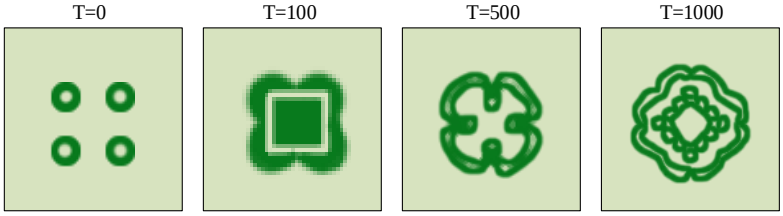
Figure 9: Evolution process of Turing spot generation by changing parameter a

In Figure9 (a), the graph shows the Turing spot evolution of vector s over time when a=2 and b=-0.1. The initial four spots gradually elongate and fuse together over time, then split into eight spots, and after migration, the eight spots are finally fixed. In Figure9 (b), the graph shows the Turing spot evolution of vector s over time when a=4 and b=-0.1. Compared with Figure9 (a), the initial four spots in the early stage still follow the step of first elongating and then fusing. In the later stage, the spots are not completely separated, forming strip-shaped spots,

and then migrating to form a new shape combining strip-shaped spots with individual spots. When the parameter a in vector s remains unchanged and b changes, the evolution process is shown in Figure10.



(a) When a=2, b=-0.1, the evolution of vector s



(b) When a=2, b=-0.3, the evolution of vector s

Figure 10: Evolution process of Turing spot generation by changing parameter b

When a=2 and b=-0.1 in Figure10 (a), the evolution process is similar because the parameters are consistent with Figure9 (a). Figure10 (b) shows the Turing pattern evolution of vector s over time when a=2 and b=-0.3. Compared with Figure10 (a), the initial four spots form a new pattern after a brief migration, and then the spots are continuously elongated, eventually merging into a new pattern in the shape of a strip. This study reveals the spontaneous formation mechanism of RDE in image edge detection from the perspective of dynamical systems by setting up Turing spot generation experiments. Turing spot generation relies on the concentration gradient of substances to generate patterns, while image detection relies on pixel gradient detection structures, both of which have the same formation mechanism. The spontaneous pattern formation mechanism demonstrated by the Turing point generation experiment is mathematically isomorphic to the edge detection task. In the Turing system, the patterns originate from the concentration differences caused by diffusion instability in the initial uniform state. In an image, the edges are precisely the gradient regions where the intensity of pixels changes dramatically. Therefore, by regulating the model parameters to guide the Turing pattern to form and stabilize at the image gradient, this mechanism is directly transformed into the precise recognition and enhancement of the image edge. Its spontaneous formation characteristic is the fundamental reason for achieving efficient and low-iteration edge detection. To comprehensively evaluate the performance of the FHN-CNN model, this study quantitatively compares it with the current advanced image processing models based on DL. The comparison models include U-Net, DeepLabv3+, and Holistically-Nested Edge Detection (HED). The experiment is conducted on the publicly available BSDS500 dataset, which contains 500 natural images and their artificially labeled edge truth values. The study adopts the widely used fixed-scale optimal F-score (ODS F-score), single-image optimal F-score (OIS F-score), and Average Precision (AP) as evaluation indicators. The comparison results are shown in Table 4.

Table 4: Performance comparison between FHN-CNN and advanced DL models

Model	U-Net [24]	DeepLabv3+ [25]	HED [26]	FHN-CNN
ODS F-score	0.795	0.812	0.782	0.789
OIS F-score	0.815	0.83	0.804	0.808
AP	0.851	0.865	0.833	0.822
Parameter Quantity (M)	31	59.5	14.7	0.015
Inference Time (ms)	68	112	45	12

According to the performance comparison results shown in Table 4, the FHN-CNN model demonstrates certain comprehensive advantages in the edge detection task. In terms of core performance indicators, the ODS F-score and OIS F-score of FHN-CNN are both higher than those of the HED model and are extremely close to the powerful U-Net model. Meanwhile, its AP value has also reached a level comparable to these advanced DL models, which fully demonstrates its strong competitiveness in detection accuracy. However, the most prominent advantage of FHN-CNN lies in its extremely high computational efficiency. The parameter count of this model is extremely low, only 0.015 million, less than one-thousandth of HED, and far lower than that of large

models such as U-Net and DeepLabv3+. Its single-image reasoning time has also been significantly reduced to as fast as 12 ms, outperforming all comparison models. These data strongly demonstrate that FHN-CNN can achieve edge detection accuracy comparable to advanced DL methods while reducing model complexity and computational overhead by orders of magnitude. This result provides a highly valuable solution for deploying efficient image processing algorithms in resource-constrained environments.

3.3 Parameter tuning of the IPT model based on RDE

The study is conducted to determine the optimal parameter configuration of the FHN-CNN model in image processing tasks through hyperparameter tuning. This process involves a systematic grid search of the key parameters in the response items to evaluate the impact of different parameter combinations on the visual effect and quantitative indicators of edge detection. The experimental results are shown in Figure11.

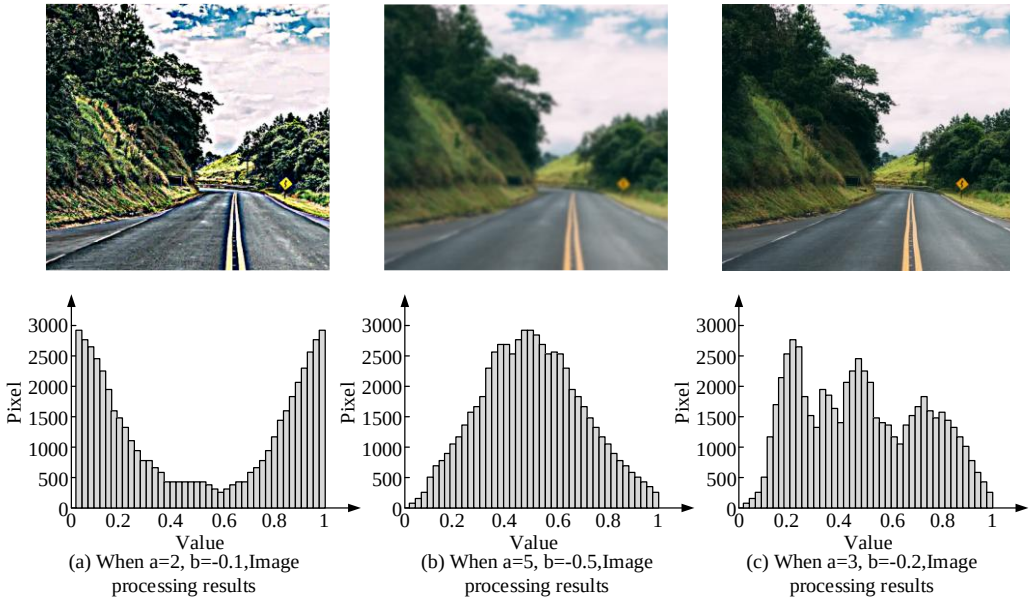


Figure 11: Image processing results under different parameter groups

In Figure11 (a), when a=2 and b=-0.1, the image undergoes excessive sharpening after edge detection processing. There are a large number of pixel values in the dark area on the left and the bright area on the right, and the histogram trend shows an abnormal state. In Figure11 (b), when a=5 and b=-0.5, the overall effect of the image appears blurry, with significantly too many gray values in the middle of the histogram, making the overall image appear grayish. In Figure11 (c), when a=3 and b=-0.2, the image processing effect is the best, with no obvious overexposure or underexposure, and the overall color of the picture is relatively harmonious. The pixels are concentrated in the middle gray value area and show a fluctuating trend up and down. By comparison, when a=3 and b=-0.2, the algorithm performs the best in image processing tasks. To verify whether RDE and the introduced adjustment function work in FHN-Cellular-NN, this study conducts two image processing experiments based on Figure11 (c) by removing the FHN module and adjustment function from the model, as shown in Figure12.

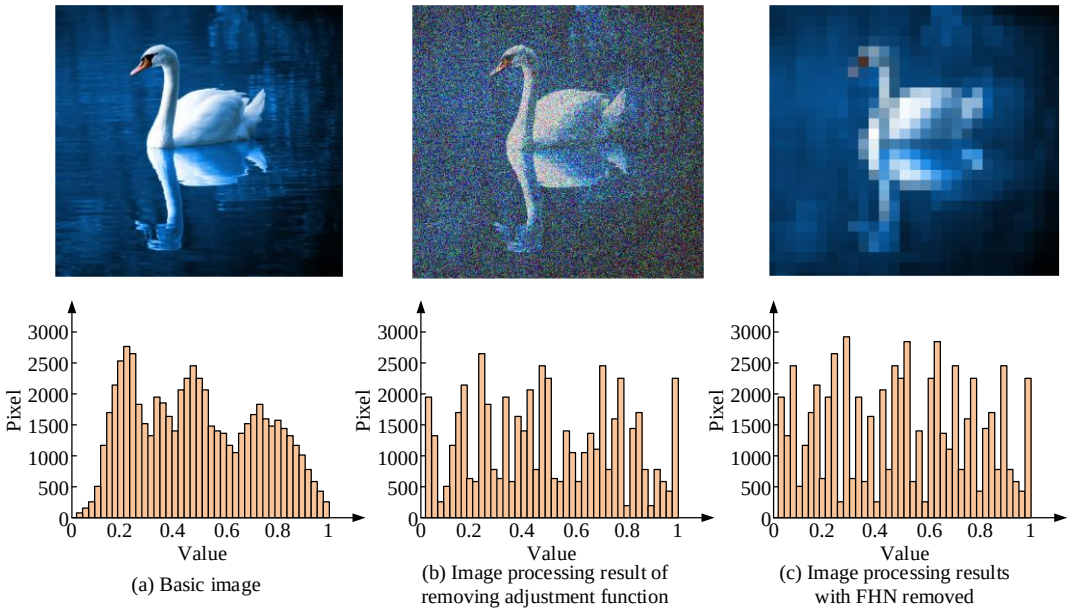


Figure 12: Image processing results after removing algorithm modules

In Figure12 (a), this image serves as the baseline for comparison with other images. In Figure12 (b), after removing the adjustment function, the image generates a lot of noise after processing, and the image quality becomes worse compared to before processing. The values of each part of the histogram are uneven, reflecting poor image processing results. In Figure12 (c), after removing the FHN module, the image generates many mosaics after processing, and the histogram shows an irregular state. The above comparison shows that the histogram trend of the base image is smoother and the image processing effect is better. Therefore, FHN and the introduced adjustment function play an important role in the algorithm, improving its image processing capability.

The FHN-Cellular-NN model has demonstrated potential in multiple practical application scenarios. For instance, in medical image analysis, this model can be used to precisely detect tissue boundaries or lesion areas. In the field of autonomous driving, its efficient edge detection capability helps to identify the contours of roads and obstacles in real time. In remote sensing image processing, the adaptability to complex environments can enhance the accuracy of ground object classification and change detection. These application prospects further highlight the practical value of FHN-Cellular-NN in promoting IPT towards high efficiency and intelligence.

4 Conclusion

For the dynamical systems, the study successfully constructed and verified the FHN-CNN image processing model based on the reaction-diffusion equation. Compared with the existing technologies, this model demonstrated unique advantages, and its outstanding performance was rooted in its inherent dynamic mechanism. The study delved deeply into the intrinsic reasons for performance improvement, clarified the unique value of the power system approach, and reflected on the limitations. The experimental results showed that FHN-CNN outperformed traditional operators and DL models in multiple core indicators of edge detection. Its outstanding performance on EPI (0.89) and PFOM (0.95) demonstrated that the model achieved a good balance between edge positioning accuracy and maintaining structural integrity. More prominently, its extremely low computing time (0.9 seconds) and number of iterations (2 times) highlighted its extraordinary computing efficiency. This performance improvement is not accidental but stems from the dynamic design at the core of the model. Firstly, through Turing instability analysis, a region in the parameter space where a stable pattern can spontaneously form is found. This enables FHN-CNN to achieve efficient edge feature generation without complex multi-layer forward propagation, merely through the dynamic evolution of the system itself. Secondly, the introduced adaptive control mechanism (adjustment function K(s)) plays a key role. This mechanism dynamically adjusts the diffusion behavior based on the local gradient of the image, ensuring that the system can accurately converge to the true edge. It effectively avoids the common edge blurring or double edge phenomena in traditional RDE methods, thereby directly contributing to the improvement of EPI and PFOM indicators. Compared with pure DL models such as U-Net and DeepLabv3+, pure DL models are typical "black boxes", with their decision-making logic deeply hidden in millions or even billions of unexplainable parameters [24]. The construction of FHN-CNN is based on clear physical laws and mathematical equations. Its dynamic behavior, such as the generation of Turing models and the evolution of stability points, can all be strictly mathematically analyzed and intuitively understood through dynamical system theory. The performance of DL models strongly depends on large-scale high-quality

labeled data, and the computational costs of training and inference are high [25]. As a model defined by partial differential equations, the parameters of FHN-CNN have clear physical meanings and do not require data-driven training from scratch.

Although the FHN-CNN model performs well in edge detection tasks, this study still has some limitations. Firstly, in its current form, the model mainly focuses on low-level visual tasks. To directly apply its output to higher-level recognition or segmentation tasks, subsequent processing flows still need to be designed. Future work will explore how to combine FHN-CNN as an interpretable feature extraction front-end with a lightweight decision module to build an end-to-end processing pipeline.

5 Conclusion

In response to the problems of strong noise interference, low edge localization accuracy, multiple iterations, and insufficient adaptability to complex images in existing image processing models, this study proposed an image processing model based on RDE. This model integrated the theoretical tools of dynamical systems into traditional image processing methods by combining FHN and Cellular-NN, thereby achieving efficient application of image processing. The experiment showed that FHN-Cellular-NN had an EPI of 0.89, a PFOM of 0.95, a time cost of 0.9 seconds, and 2 iterations, with significantly better performance than the comparison model. In numerical simulation experiments, the convergence order of RDE vector s and vector q showed a monotonically decreasing trend with the decrease of time step and spatial step, indicating high algorithm stability. Meanwhile, this study also simulated the generation of Turing spots through numerical examples, revealing the spontaneous formation mechanism of RDE in image edge detection from the perspective of dynamical systems. In the model recognition analysis, the optimal parameter set of FHN-Cellular-NN in the direction of image processing was found. In conclusion, through a series of performance tests, numerical simulations, and model recognition, this study has demonstrated the advantages of FHN-CNN in image processing and its ability to meet the requirements of complex image processing. However, the study has the following limitations. Firstly, although it has been compared with DL methods such as U-Net, the comparison scope still needs to be further expanded. Secondly, there is still room for improvement in the diversity and scale of the experimental dataset. Furthermore, the ablation analysis of the respective contributions of CNN and FHN is insufficient. Subsequent research will increase the scale of the dataset, expand the application verification of the model in different types of image processing tasks, and conduct in-depth analysis of the contribution of each module to further enhance the universality and robustness.

References

[1] Kaur N, Jindal N, Singh K. A deep learning framework for copy-move forgery detection in digital images. Multimedia Tools and Applications, 2023, 82(12): 17741-17768. https://doi.org/10.1007/s11042-022-14016-2

[2] Wu Q. Research on deep learning image processing technology of second-order partial differential equations. Neural Computing and Applications, 2023, 35(3): 2183-2195. https://doi.org/10.1007/s00521-022-07017-7

[3] Srinivas N, Mandaloju N, Nadimpalli S V. Leveraging Automation in Software Quality Assurance: Enhancing Defect Detection and Improving Efficiency. International Journal of Acta Informatica, 2024, 3(1): 112-124.

[4] Zangana H M, Mohammed A K, Mustafa F M. Advancements in Edge Detection Techniques for Image Enhancement: A Comprehensive Review. International Journal of Artificial Intelligence & Robotics (IJAIR), 2024, 6(1): 29-39. https://doi.org/10.25139/ijair.v6i1.8217

[5] Chen Z, Pawar K, Ekanayake M, Pain C, Zhong S, Egan G F. Deep learning for image enhancement and correction in magnetic resonance imaging-state-of-the-art and challenges. Journal of Digital Imaging, 2023, 36(1): 204-230. https://doi.org/10.1007/s10278-022-00721-9

[6] Zhou J, Pang L, Zhang D, Zhang W. Underwater image enhancement method via multi-interval subhistogram perspective equalization. IEEE Journal of Oceanic Engineering, 2023, 48(2): 474-488. https://doi.org/10.1109/JOE.2022.3223733

[7] Zhang W, Zhou L, Zhuang P, Li G, Pan X, Zhao W, et al. Underwater image enhancement via weighted wavelet visual perception fusion. IEEE Transactions on Circuits and Systems for Video Technology, 2023, 34(4): 2469-2483. https://doi.org/10.1109/TCSVT.2023.3299314

[8] Jiang H, Luo A, Fan H, Han S, Liu S. Low-light image enhancement with wavelet-based diffusion models. ACM Transactions on Graphics (TOG), 2023, 42(6): 1-14. https://doi.org/10.1145/3618373

[9] Arshaghi A, Ashourian M, Ghabeli L. Potato diseases detection and classification using deep learning methods. Multimedia Tools and Applications, 2023, 82(4): 5725-5742. https://doi.org/10.1007/s11042-022-13390-1

[10] Ninomiya H, Taniguchi M. Traveling Front Solutions of Dimension n Generate Entire Solutions of Dimension $$(n-1) $$ in Reaction-Diffusion Equations as the Speeds Go to Infinity. Archive for Rational Mechanics and Analysis, 2025, 249(1): 1-37. https://doi.org/10.1007/s00205-025-02083-2

[11] Liu J P, An D, Fang D, Wang J, Low G H, Jordan S. Efficient quantum algorithm for nonlinear reaction-diffusion equations and energy estimation. Communications in Mathematical Physics, 2023, 404(2): 963-1020.

https://doi.org/10.1007/s00220-023-04857-9
[12] Xue S, Song Y, Wang H. Spatio-temporal dynamics in a reaction-diffusion equation with nonlocal spatial memory. SIAM Journal on Applied Dynamical Systems, 2024, 23(1): 641-667. https://doi.org/10.1137/22M1543860
[13] Zayed E M, El-Horbaty M, Saad B M, Arnous A H, Yildirim Y. Novel solitary wave solutions for stochastic nonlinear reaction-diffusion equation with multiplicative noise. Nonlinear Dynamics, 2024, 112(22): 20199-20213. https://doi.org/10.1007/s11071-024-10085-0
[14] Agresti A. Delayed blow-up and enhanced diffusion by transport noise for systems of reaction-diffusion equations. Stochastics and Partial Differential Equations: Analysis and Computations, 2024, 12(3): 1907-1981. https://doi.org/10.1007/s40072-023-00319-4
[15] Hadhoud A R, Rageh A A M, Agarwal P. Numerical method for solving two-dimensional of the space and space-time fractional coupled reaction-diffusion equations. Mathematical Methods in the Applied Sciences, 2023, 46(5): 6054-6076. https://doi.org/10.1002/mma.8891
[16] Kaur A, Dong G. A complete review on image denoising techniques for medical images. Neural Processing Letters, 2023, 55(6): 7807-7850. https://doi.org/10.1007/s11063-023-11286-1
[17] Barbu T. Deep learning-based multiple moving vehicle detection and tracking using a nonlinear fourth-order reaction-diffusion based multi-scale video object analysis. Discrete & Continuous Dynamical Systems-Series S, AIMS Journals, 2023, 16(1): 6-32. https://doi.org/10.3934/dcdss.2022083
[18] Hagemann P, Mildenberger S, Ruthotto L, Steidl G, Yang N T. Multilevel diffusion: Infinite dimensional score-based diffusion models for image generation. SIAM Journal on Mathematics of Data Science, 2025, 7(3): 1337-1366. https://doi.org/10.1137/23M1614092
[19] Prakash S R, Singh P N. Background region-based face orientation prediction through HSV skin color model and K-means clustering. International Journal of Information Technology, 2023, 15(3): 1275-1288. https://doi.org/10.1007/s41870-023-01174-1
[20] Kim S, You J. Efficient LUT design methodologies of transformation between RGB and HSV for HSV based image enhancements. Journal of Electrical Engineering & Technology, 2024, 19(7): 4551-4563. https://doi.org/10.1007/s42835-024-01859-y
[21] Chen X, Wang N, Wang K, Chen M, Parastesh F, Xu Q. Coupling dynamics in an FHN bi-neuron model coupled via ReLU function-based locally active memristor. Nonlinear Dynamics, 2024, 112(22): 20365-20379. https://doi.org/10.1007/s11071-024-10127-7
[22] Sun B, Liu X, Wang J, Wei X, Yuan H, Dai H. Short-term performance degradation prediction of a commercial vehicle fuel cell system based on Cellular-NN and LSTM hybrid neural network. International journal of hydrogen energy, 2023, 48(23): 8613-8628. https://doi.org/10.1016/j.ijhydene.2022.12.005
[23] Zheng Q Q, Shen J W, Guan L N. Turing instability and pattern formation induced by noise in the modified SIR model. Nonlinear Dynamics, 2025, 113(5): 4355-4368. https://doi.org/10.1007/s11071-024-09927-8
[24] Rajamani K T, Rani P, Siebert H, ElagiriRamalingam R, Heinrich M P. Attention-augmented U-Net (AA-U-Net) for semantic segmentation. Signal, image and video processing, 2023, 17(4): 981-989. https://doi.org/10.1007/s11760-022-02302-3
[25] Chen H, Qin Y, Liu X, Wang H, Zhao J. An improved DeepLabv3+ lightweight network for remote-sensing image semantic segmentation. Complex & Intelligent Systems, 2024, 10(2): 2839-2849. https://doi.org/10.1007/s40747-023-01304-z
[26] An Y, Jing J, Li X, Zhang J, Bao J. An exclusive U-net for fine and crisp edge detection. Multimedia Tools and Applications, 2024, 83(18): 54657-54672. https://doi.org/10.1007/s11042-023-17706-7

Real-time Anomaly Detection and Multi-class Classification in Surveillance Videos Using Deep Learning

Deepika Varshney
Department of Computer Science & Information Technology, Jaypee Institute of Information Technology, Noida, India
E-mail: deepikavarshney06@gmail.com

Keywords: crime, machine learning, deep learning, anomaly detection, surveillance videos, C3D

Received: January 26, 2025

With the escalating crime rates in the school and colleges, surveillance cameras have become instrumental in monitoring and detecting anomalies happened in the campus to reduce crime rates. This study proposes a novel deep learning-based approach for real-time anomaly detection and multi-class classification in surveillance videos. We employ a Convolutional 3D Neural Network (C3D) model for extracting features and detecting probabilistic anomalies within video frames. Subsequently, a Simple Recurrent Neural Network (RNN) is trained to classify detected anomalies into one of thirteen distinct categories, including Abuse, Arson, Assault, among others. Our approach outperforms previous methods, achieving an accuracy of 83.67% for anomaly classification on the UCF Crime dataset. This work demonstrates the feasibility of automatically detecting and classifying anomalies in real-world surveillance scenarios, potentially enhancing public safety and well as school/college campus safety and facilitating prompt response measures.

Povzetek: Študija predlaga realnočasno zaznavanje in razvrščanje anomalij v nadzornih videih (C3D za zaznavo, RNN za 13 razredov), ki na podatkovni zbirki UCF Crime doseže visoko točnost.

1 Introduction

In the era of information technology, the Surveillance cameras are widely being integrated into public locations to improve public safety, such as residential buildings, offices, malls, crossroads, banks, shops, market area and other places. Even there is a wide use of surveillance camera in these places, it does not tend to be beneficial for the government law enforcement agencies for monitoring abnormal activities. Due to this, there is a noticeable lack in the deployment of surveillance camera. The important key role in video surveillance is to detect abnormal occurrences such as traffic accident, crimes or unlawful activities. Anomalous activities are uncommon as compared to typical events. To overcome this, it is required to incorporate deep learning algorithms for the automatic detection of anomalous video to reduce time and manpower. The purpose of a realistic anomaly detection system is to detect and communicate activity that deviates from typical patterns in a timely manner, as well as to determine the time window in which the abnormality occurs. As a result, anomaly detection may be thought of as high-level video knowledge that separates abnormalities from typical patterns.

Once an anomaly has been identified, classification techniques may be used to categorize it into one of the specialized activities. Developing algorithms to identify a specific anomalous event, such as a violence detector [1] or a traffic accident detector [2, 3], is a tiny step toward tackling anomaly detection. However, such solutions are evident in that they cannot be applied to identify other abnormal occurrences, therefore they are only useful in theory.

Anomaly events in the real world are difficult and varied. It's impossible to make a comprehensive list of all conceivable anomalous events. As a result, it is best if the anomaly detection method doesn't rely on any prior knowledge of the occurrences. In other words, anomaly detection should be carried out with the utmost care. Approaches based on sparse coding [4, 5] are regarded as exemplary methods for achieving state-of-the-art anomaly detection outcomes. These approaches presume that just a tiny fraction of the beginning of a film has normal events, hence the beginning is utilized to construct the normal event dictionary. The primary notion behind anomaly detection is that abnormal occurrences can't be reliably recreated using the standard event dictionary. However, because the environment collected by security cameras might change dramatically over time (for example, during different times of the day), these systems result in significant false alarm rates for various routine actions. The Figure 1 Alt Text shows the examples concerning to normal videos and anomalous videos with their description. Although the ideas described above are tempting, they are predicated on the notion that any behavioral pattern that differs from the learned normal usual patterns is an aberration. This assumption, however, may not be correct because defining a typical event that encompasses all conceivable normal patterns/behaviors is extremely difficult or impossible [6]. What's more, the line between normal and abnormal conduct is frequently blurred. Furthermore, given realistic circumstances, the

same behavior might be considered normal or abnormal depending on the circumstances. As a result, it is believed that normal and anomalous event training data can aid an anomaly detection system's learning. We present an anomaly detection system based on weakly labelled training movies in this research.

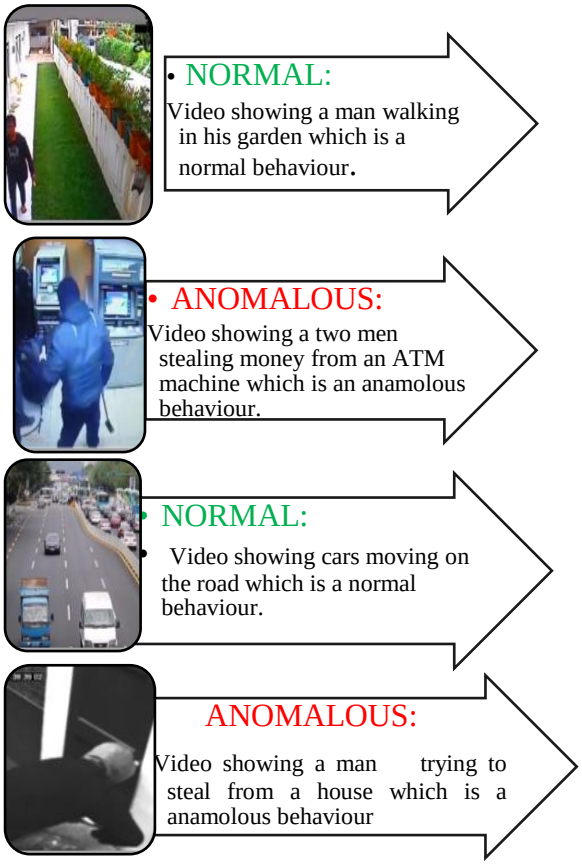


Figure 1: Normal vs anomalous videos

That is, we just know the video-level labels, which means that a video is normal or includes an abnormality someplace, but we don't know where. This is fascinating since adding video-level labels allows us to simply annotate a huge number of films [7, 8]. This proposed work is divided into two major tasks: first is to detect the anomaly in the video by measuring the probability of anomalous activity inside the frame, and the second task is to classify the type of anomalous activity (one among the 13 anomalous activities listed in dataset description i.e., Abuse, Arrest, Arson, Assault, Accident, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting, and Vandalism) in the video. We automatically learn an anomaly ranking algorithm based on training movies that predicts high anomaly scores for anomalous regions in a video. During testing, a lengthy untrimmed video is broken into parts and fed into our deep network, which provides an anomaly score to each video segment in order to discover anomalies. The following are the key contributions made by this study.

- The first task is to detect the anomaly in the video by measuring the probability of anomalous activity inside the frame,
- The second task is to classify the type of anomalous activity (one of the 13 anomalous activities listed in the dataset description) in the video.
- The comparative analysis has been performed with the baseline approach C3D [9] and C3D+RNN [10], on detecting 13 distinct abnormal behaviors concerning anomaly detection and the experimental findings on the UCF dataset reveal that the suggested strategy outperforms others methods.

The rest of the work discussed in the paper is structured as follows. The upcoming Section 2 discusses the detailed state of the art work that are focused on the crime rate prediction and classification. Whereas, Section 3, incorporates the detailing of dataset used, the methods adopted, and the technology used to build the crime rate detection model shown in Section 4. In Section 5, the comparative and experimental analysis as well as the results have been shown. Section 6 shows the conclusion and future scope.

2 Related work

Many Researchers have put their keen interest towards videos to solve the real time problems [30]. Recent works presented a deep learning strategy based on picture normalisation and CNN architecture for automated pedestrian detection. Their suggested architecture adaptively learns pedestrian representation to accomplish efficient recognition [11] with improved accuracy and shorter pre-processing times. Their work swiftly and properly wins distinctive pedestrian features. Images from the data set are shrunk to a fixed scale, adjusted with a unity standard deviation, and fed into the proposed DCNN to distinguish pedestrians from non-pedestrians.

In [12], authors described a technique based on theory of queuing and computer vision for detecting pedestrians and computing essential macroscopic statistics of pedestrian traffic. They used an Aggregated Channel Feature based detector to detect pedestrians in a given zone of interest in a traffic video. Their unique approach was then applied in-situ at traffic monitoring camera locations to create a distributed traffic control system. Whereas, in [13], the author has added a revertive link to the pedestrian re-recognition detector, making it more like the human cognitive process by turning a single image into an image sequence, which is then reidentified by the memory image sequence pattern. Deep learning-based pedestrian re-recognition algorithms can now recall visual sequence patterns and re-identify pedestrians in photos, thanks to their technique. Simultaneously, their study offered a shallow learning strategy based on selective dropout. The other studies incorporate quick approach for recognising pedestrians in surveillance systems with less memory size and processing units. As even compressed architecture of deep neural net needed large memory, the proposed work by [14], added a compression technique based on a teacher student framework to a random forest classifier instead of a wide and deep network.

In [15], the authors proposed that processing in device scenarios used to identify crowd anomalies. In this paper, an abnormality analysis method based on improved k-

means is proposed. This approach combines mean shift with the k-means classification procedure to provide quick and accurate detection of crowd anomaly.

In [16], authors examined three aspects of this issue. To begin with, a novel global factor defines the moving crowd. This function might very well characterise the specifics of the point-of-interest in terms of spatial and temporal motion. Then, a method is used that clusters the feature point first and then measures it collectively, allowing for more coherent and efficient computation of individual groups.

The optical flow was employed as supplemental information for anomaly identification. They created a deep WCAELSTM network [17] that captures spatial variances with CAE and temporal variances with three convolutional LSTM (Conv-LSTM) units, and proposed a weighted Euclidean loss that focuses on moving foregrounds, thus limiting the influence of complex backgrounds, as well as a global-local analysis to achieve anomaly detection and localization jointly.

In this study, authors presented three contributions [18]: (1) a new structural context descriptor is proposed to characterise the structural properties of individuals in crowd scenes; (2) a self-weighted multi-view clustering method is proposed to cluster feature points by incorporating their orientation and context similarities; and (3) a novel framework for group detection is introduced.

Researchers proposed a crowd anomaly detection framework that satisfied continuous feed in of Spatio-temporal [19], information from live CCTVs. Firstly, an extraction algorithm for the spatial-temporal texture is built. Their method will efficiently strip textures from the video with ample information about crowd motion. It is by the adoption of Gabor-filtered textures with the maximum entropy values of knowledge.

Introduced a new system for global abnormality identification while using context location (CL) as well as motion-rich STVs (MRSTVs) [20] through block-level function extraction. In order to record motion properties of regular and irregular cases, the histogram of optical movement direction as well as motion magnitude features in Spatio-temporal volumes (STVs) is being used as a regional object descriptor.

The researchers concentrated on deep learning methodology for crime detection in videos. Deep learning techniques are evaluated based on their algorithms and models. The main model [21] purposed to use deep learning algorithms to identify the precise list, the individuals engaged, and the behaviour that has developed in a large crowd under any weather situation.

In this paper, Aravindan, S., E. Anusuya, and M. Ashok Kumar developed a GUI model for Chennai and (SMLT) supervised machine learning technique model [22], which finds the best model among SML algorithms by getting more than 90% accuracy for determining crime rate by PSA and year. The dataset is taken from the online Indian police department, split into training and testing in 7:3, and applied to Random Forest, SVC, KNN, Decision tree, and logistic regression algorithms. The research gap is not comparing with deep learning models and not incorporating this model into a real-time system for crime rate prediction.

Wang and Bao developed deep learning ST-Resnet model which outperforms the fully-ternary ST-Resnet, ARIMA, KNN, and HA model. The authors used the well-represented data to adjust the spatial-temporal residual network to forecast the distribution of crime in Los Angeles at the hourly scale in neighbourhood-sized blocks. These trials, along with comparisons with other prediction methods, show that the suggested model is more accurate [23]. Finally, the authors offer a ternary approach to solve the issue of resource consumption for real-world deployment. Researchers Tabedzki and Christian studied developed a machine learning technique for predicting crime-related statistics in Philadelphia, United States [24]. The problem was broken down into three sections: identifying whether or not a crime happens, the likelihood of a crime occurring, and the most likely crime occurring. To generate comprehensive quantitative crime predictions with increased relevance, algorithms including logistic regression, KNN, ordinal regression, and tree techniques were utilized to train the datasets. They also showed a map that showed different crime categories in different areas of Philadelphia for a certain time period, with different colours representing each sort of crime. Various sorts of crimes, ranging from assaults to cyber fraud, were included in order to mirror the overall trend of crime in Philadelphia over a period of time. Their system predicted the likelihood of a crime with a remarkable 69% accuracy, as well as the number of crimes ranging from one to 32% with a 47% accuracy.

The authors of [25], concentrated primarily on the study and construction of machine learning algorithms to lower crime rates in India. To discover the pattern relations between a huge amount of data, machine learning techniques were used. The study's major goal was to provide a prediction of future crime based on the prevalence of prior crime areas.

The dataset was analysed and interpreted using approaches such as Bayesian neural networks, the Levenberg Marquardt algorithm, and a scaled algorithm, with the scaled algorithm outperforming the other two techniques. A statistical analysis based on correlation, analysis of variance, and graphs revealed that the crime rate may be decreased by 78% using the scaled method [26] with an accuracy of 0.78. On the other hand, Hossain and Sohrab’s study proposes a method for predicting crime based on an analysis of a dataset comprising records of already committed crimes and their trends. The suggested system is based on two machine learning algorithms: decision trees and KNNs. Many of the earlier studies also applied video surveillance based deep learning model for the detection of criminal activities [27,28]. The authors of [29], detect anomaly by examining separate memory modules for normal and anomalous patterns, we initiate the process by generating pseudo-anomalies with a temporal pseudo-anomaly synthesizer.

The prediction model's accuracy was improved using techniques like the random forest algorithm and adaptive boosting. The offenses were classified into common and rare classes to improve the model's results. The most

common crimes were in the frequent class, while the least common crimes were in the rare class. The suggested system was given data on criminal activities in San Francisco during a 12-year period. The accuracy was raised to 99.16% by combining under sampling and oversampling approaches with the random forest algorithm.

The best way to highlight the novelty in our study is by comparing it with the work that was done by others and pointing out the things that our study does which was never done before. It is described in the following format for each research, first conduct a thorough literature search to identify what is already known in your field of research and what are the gaps to be explored. In some researches, the research gap is not identifying other abnormal activity like which is considered in our research and not incorporating this model into a real-time system for crime rate prediction. The authors of [31], developed the Composite Recurrent Bi-Attention (CRBA) model to detect anomalies in surveillance footage. It leverages DenseNet201 for comprehensive spatial feature extraction and utilizes BiLSTM networks to model temporal relationships between video frames. To further enhance its performance, a multi-attention mechanism is incorporated, guiding the model's attention to key spatiotemporal regions. This combination of methods allows the CRBA model to effectively tackle both spatial and temporal aspects of video data, resulting in more accurate anomaly detection. Experimental results indicate strong performance on both the University of Central Florida (UCF) dataset and the newly introduced Road Anomaly Dataset (RAD).They achieves an F1-score of 83.8% when trained on the UCF dataset. While, some other research is using different methods to classify and detect and getting lower accuracy in comparison to ours. In our study, it is clearly shows that we are detecting the abnormality in the video by measuring the probability of anomalous activity using (Convolutional 3D neural network (or C3D) model) and detection of abnormal activity using simple RNN, which has not been done before in any other research done on this dataset of UCF Crime. Also, we are testing our model on the custom videos created by us for real-time detection.

In the end our research will were able to detect crime using C3D model and detect abnormal behaviour using simple RNN model. Beyond the scope of other researches, we are also testing on our custom dataset which makes our model uniquely precise in comparison to other existing models. Real-Time Surveillance videos are able to capture a variety of realistic criminal activities. In order to minimize crime, we need to be able to detect it as soon as it happens. Crime prediction and detection is the unique ability to predict future crime by studying the trends of the past. Through this paper, the crime rate can be forecasted and hence minimized. In case any abnormal activity is detected during the live stream, there will either be a notification issued to the authorities. This notification will either be linked to mobile devices of nearby police personnel or may be linked to a hardware IOT device that we plan to develop.

3 Data description

The dataset contains videos of UCF crime dataset[1] used for Real-world Anomaly Detection in Surveillance Videos taken from Kaggle.

The dataset contains videos of UCF Crime Dataset. In data preprocessing every 10th frame is extracted from each full-length video and combined for every video in that class. All the images are of size 64*64 and in ".png" format. The dataset consists of 13 realistic anomalies including Abuse, Arrest, Arson, Assault, Road Accident, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting, and Vandalism. The total image count for the train subset is 1,266,345. The total image count for the test subset is 111,308. The average number of frames in this dataset is 7247. The dataset consists of 1900 long and untrimmed real-world surveillance videos. As an example of dataset Figure 2 depicts Abuse class video, Accident class and Robbery class video respectively. The complete size of dataset is 82.9 GB. With a total of 128 hours of videos from CCTV cameras.



Figure 2: (a) Abuse class video (b) Accident class video (c) Robbery class video

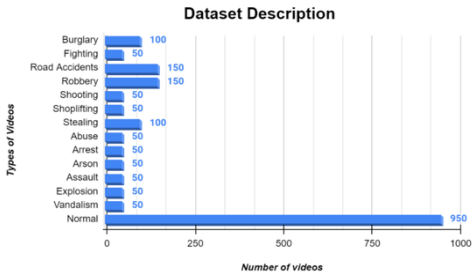


Figure 3: Bar chart depicting distribution of videos in dataset

The dataset is constructed as a new large-scale dataset, called UCF-Crime, to evaluate various method. The Figure 3 **Alt Text** gives brief description of the dataset . It consists of long untrimmed surveillance videos which

[1]https://www.kaggle.com/datasets/odins0n/ucf-crime dataset

cover 13 real world anomalies, including Abuse, Arrest, Arson, Assault, Road Accident, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting, and Vandalism along with their distribution. These anomalies are selected because they have a significant impact on public safety.

4 Methodology

4.1 Proposed methodology

In our study, it is clearly shows that we are detecting the abnormality in the video by measuring the probability of anomalous activity using (Convolutional 3D neural network (or C3D) model) and detection of abnormal activity using simple RNN, which has not been done before in any other research done on this dataset of UCF Crime. The flow diagram of our proposed methodology is summarized in the Figure 4. This proposed work is divided into two major tasks: first is to detect the anomaly in the video by measuring the probability of anomalous activity inside the frame, and the second task is to classify the type of anomalous activity (one among the 13 anomalous activities listed in dataset description i.e., Abuse, Arrest, Arson, Assault, Accident, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting, and Vandalism) in the video. The videos in the dataset have variable lengths so variable number of frames, and computing on all the available frames is a very costly task. Hence, initially we tried to reduce the frames so as to maintain the necessary details as well as reduce the cost. For training the neural networks as shown in Figure 4, we considered one-tenth of the total number of frames available in each video taken at regular interval.

4.1 Anomaly detection in the video

This task involves finding the probability of any anomalous behaviour in the video at different instances. This task was break down into supervised approach of finding whether a particular video clip has anomalous behaviour or not. For this we trained a general Convolutional 3D neural network (or C3D) model on provided labelled videos in the dataset. The C3D model was capable of extracting the features and training them against the labels. The extracted features of all the videos were saved for further classification of anomaly.

Algorithm 1: Anomaly Detection in Video

```
set m = 10
for each video in dataset:
    set num_frames = number of frames
    set req_frames = num_frames/10
    set first_frame = 1
    set last frame = 10*m+1
    while (number of clips possible):
        clip = create a clip window with m number of frames
        pass the clip to trained c3d model
        probability = model.predict(clip)
        plot probability against the frame number
        first_frame += 10
        last_frame += 10
    end while
end for
```

Now, to find the probabilities of anomalous behaviour at different instances in the video, the complete algorithm is summarized in the Algorithm 1. Each video is segmented into multiple small clips (in this proposed methodology we used 10 frames per clip). Hence, in a video having n number of frames we created n-m+1 number of clips having m frames in each clip. Each of these clips is passed as an input to the trained C3D model, which provide the probability of any anomalous behaviour in that clip.

4.2 Anomaly classification in the video

Now after the anomaly is detected in the video, the task is to classify the type of anomaly detected. Classifying the anomaly is again a supervised approach, were any machine learning or deep learning algorithm can train on the features and label. This will be a multi-class classification problem. There could be multiple possible ways of doing it like using C3D itself for multi-lass classification, or using combination of 2DCNN+RNN or features extracted using C3D+RNN. Here we used the features extracted using C3D model to train a Simple RNN model. For each video, features of all the considered frames are sequenced in order to form a vector of features.

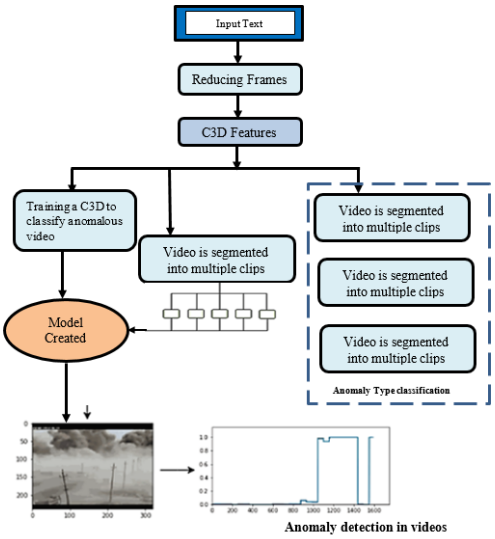


Figure 4: The Flow Diagram of the proposed methodology

This feature vector of all the videos are grouped into a single matrix. Strings serve as the labels for the videos. Because neural networks cannot grasp string values, they must be transformed to a numerical format before being given to the model. The Simple RNN trained on features and numerical labels to perform a multiclass classification task.

5 Experimental analysis and results

With increasing number of criminal offences all across the world, it is important to monitor and keep a track of these cases so that they can be prevented in future. The paper revolves around development of a surveillance system to detect anomalous activities and inform the required authorities about them so that they can be prevented and crime rates can be controlled. To address this problem UCF dataset was used for training the C3D model to detect anomalous activities in surveillance videos. Further a simple RNN was trained to classify the anomaly into one of the 13 types as present in dataset.

5.1 Probabilistic anomaly detection results

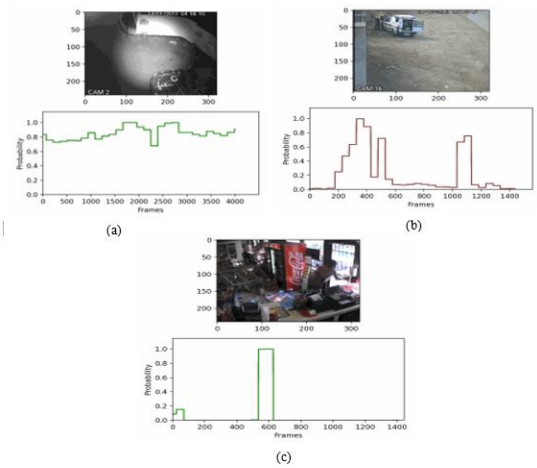


Figure 5: Detection of probabilistic anomaly

After the C3D model was trained, the model was tested on the testing set and the results obtained are depicted in the Figure 5. The X axis in the given graphs depicts the frame of the video while the Y axis depicts the probability that an anomaly has occurred in some frames of that segment. In case of normal videos, the probabilistic anomaly is found to be zero since no anomalous activity takes place in any frame. This is shown in Figure 6. Figure 5 (a), (b) and (c) Alt Text demonstrates anomalous activity and the probabilistic anomaly detected corresponding to it as shown in the graphs. The C3D model has been applied and from the experimental analysis it is depicted the anomaly with an accuracy of 84.5%. The demo outcome of the experiment you can check on the link[1]. The results are presented in three forms, first in anomalous, normal and customized videos in Figure 5, 6 and 7 respectively.

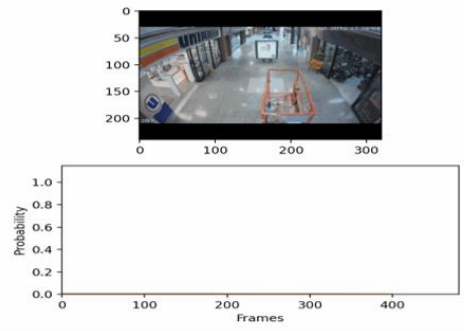


Figure 6: Detection of probabilistic anomaly in normal videos

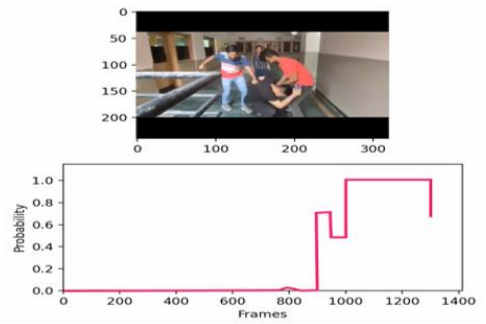


Figure 7: Detection of probabilistic anomaly in customized videos

The model was also tested on some customized videos wherein anomalous activity was performed and its detection was checked.

5.2 Multi-class classification results

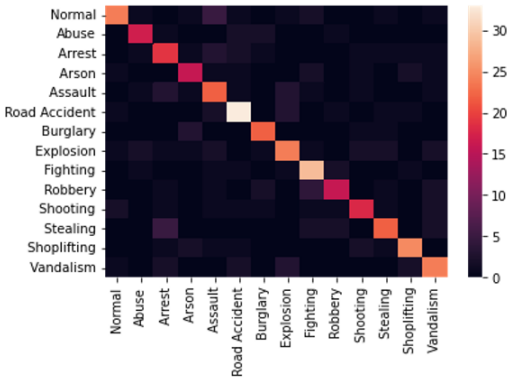


Figure 8: Depicts the confusion matrix for the

Table 2: The performance analysis of different typ

After the detection of anomalous activity, multi class classification performed using simple RNN so as to classify the videos into the type of anomaly detected in the video.

The Table 2 depicts the performance measures scores for multi- class classification of anomalies. As depicted clearly, the accuracy obtained for classification was around 83.67%. The anomalous activities were classified into either of the 13 anomaly categories considered or into the category of normal videos. Further, Some false results were also obtained in case of anomaly detection where, either the normal videos were classified as anomalous or anomalous activity was not detected and the videos were considered normal. Compared with the previous work

	Precision	Recall	F1-Score	Support
Normal	0.844444	0.883721	0.863636	43
Abuse	0.736842	0.823529	0.777778	34
Arrest	0.780488	0.8	0.790123	40
Arson	0.821429	0.793103	0.807018	29
Assault	0.861111	0.794872	0.826667	39
Road Accident	0.904762	0.883721	0.894118	43
Burglary	0.864865	0.864865	0.864865	37
Explosion	0.852941	0.763158	0.805556	38
Fighting	0.885714	0.794872	0.837838	39
Robbery	0.891892	0.891892	0.891892	37
Shooting	0.8	0.818182	0.808989	44
Stealing	0.825	0.846154	0.835443	39
Shoplifting	0.808511	0.863636	0.835165	44
Vandalism	0.885714	0.911765	0.898551	34
Accuracy			**0.83666**	
Macro Avg	0.840265	0.838105	0.838403	540
Weighted Avg	0.840576	0.838889	0.838961	540

done on this problem, our model outperforms other methods and has obtained an accuracy of 83.67%. The Table 3, depicts the comparative analysis of our approach with the methodology adopted by others. The proposed method involved the latest idea of deep learning concepts. As per the comparative study shown in the Table 3, our proposed models work well for complex cases and achieve the best results in many of the past studies.

Table3: Comparative Analysis

Methodology adopted	**Accuracy Obtained**
2D CNN	72%
C3D (2 stream)	75.9%
I3D (RGB)	79.9%
C3D on Segmented videos (Our Approach)	**83.6%**
ResNET 50	45.20
MobileNet V2	41.90
[34]	52.70
ConvGRU-CNN [32]	82.22%
Pushpajit Khaire[33]	91.90%

6 Conclusion and future scope

To detect real-world abnormalities in surveillance films, we present a deep learning technique. because of the intricacy of these realistic anomalies, utilizing simply normal data for anomaly identification may not be the best option. We try to take advantage of both conventional and unusual surveillance recordings. We proposed a twofold approach for the detection and classification of anomalies. We made use of C3D for feature extraction and detection of probabilistic anomaly in surveillance videos. Further we trained a simple RNN for multi- class classification to determine the anomaly types. We achieved an accuracy was around 83.67% for multi- class classification. We further plan on developing a real time surveillance system to detect anomalous activity and notify the required authorities as and when an anomaly is suspected in any area. We wish to come up with IoT based solutions to raise alarm before crime incident takes place. Raising awareness amongst citizens about their rights and encourage them to raise their voice against crime remains a major concern. In future we are planning to add the upcoming.

Constraints of the study:

Given the fact that this paper is done with great care and attention to detail, there are certain constraints we encountered are as follows.

- First, the entire system must be built correctly and completely in the near future in order for it to be implemented quickly and correctly. Furthermore, because such technologies cannot be directly deployed in the open world, implementation is a major problem, wrt cost.
- The system must first be evaluated in a small region of a metropolitan area, and only then can its use be expanded up with continual improvements (first-model revisions). As a result, the obstacles are more of a help in improving the model and producing a flawless model that can be applied to the actual world over time.
- Additionally, there are a few technological challenges with the model, since the amount of the learning data will be vast, requiring days, if not weeks, to analyze it.

References

[1] Mohammadi, Sadegh, et al. "Angry crowds: Detecting violent events in videos." *European Conference on Computer Vision*. Springer, Cham,

2016. https://doi.org/10.1007/978-3-319-46478-7_1.

[2] Kamijo, Shunsuke, et al. "Traffic monitoring and accident detection at intersections." *IEEE transactions on Intelligent transportation systems* 1.2 (2000): https://doi.org 108-118.W. 10.1109/6979.880968.

[3] Sultani, Waqas, and Jin Young Choi. "Abnormal traffic detection using intelligent driver model." *2010 20th International Conference on Pattern Recognition*. IEEE, 2010. 10.1109/ICPR.2010.88

[4] Lu, Cewu, Jianping Shi, and Jiaya Jia. "Abnormal event detection at 150 fps in matlab." *Proceedings of the IEEE international conference on computer vision*. 2013. DOI: 10.1109/ICCV.2013.338

[5] B. Zhao, L. Fei-Fei, and E. P. Xing. Online detection of unusual events in videos via dynamic sparse coding. In CVPR, pages 3313–3320, 2011. DOI: 10.1109/CVPR.2011.5995524

[6] Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." ACM computing surveys (CSUR) 41.3 (2009): 1-58. https://doi.org/10.1145/1541880.1541882

[7] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez. Solving the multiple instance problem with axis-parallel rectangles. Artificial Intelligence, 89(1):31–71, 1997. https://doi.org /10.1016/s0004-3702(96)00034-3.

[8] Andrews, Stuart, Ioannis Tsochantaridis, and Thomas Hofmann. "Support vector machines for multiple-instance learning." *Advances in neural information processing systems* 15 (2002). https://doi.org/10.1007/s10479-012-1241-z.

[9] Tran, Du, et al. "Learning spatiotemporal features with 3d convolutional networks." *Proceedings of the IEEE international conference on computer vision*. 2015. https://doi.org/10.1109/ICCV.2015.510.

[10] Hou, Rui, Chen Chen, and Mubarak Shah. "Tube convolutional neural network (T-CNN) for action detection in videos." *Proceedings of the IEEE international conference on computer vision*. 2017. https://doi.org/10.48550/arXiv.1703.10664.

[11] Al-Hazaimeh, Obaida M., and M. Al-Smadi. "Automated pedestrian recognition based on deep convolutional neural networks." *International Journal of Machine Learning and Computing* 9.5 (2019): 662-667. https://doi.org/10.18178/ijmlc.2019.9.5.855.

[12] Ritwika Chowdhury, Kinjal Bhattacharyya, Sudipta Mukhopadhyay and Bhargab Maitra, "Multi-Directional Pedestrian Detection and Flow Monitoring from Traffic Videos", Proceedings of International Conference on Transportation Research, pp. 34-41, 2018.

[13] Feng-Ping An, "Pedestrian Re-Recognition Algorithm Based on Optimization Deep Learning-Sequence Memory Model", Complexity, Vol. 2019, pp. 1-6, 2019. https://doi.org/10.1155/2019/5069026

[14] Kim, Sangjun, Sooyeong Kwak, and Byoung Chul Ko. "Fast pedestrian detection in surveillance video based on soft target training of shallow random forest." *IEEE Access* 7 (2019): 12415-12426. https://doi.org/10.1109/ACCESS.2019.2892425

[15] Shuqiang Guo, Qianlong Bai, Song Gao, Yaoyao Zhang and Aiquan Li, "An Analysis Method of Crowd Abnormal Behavior for Video Service Robot", IEEE Access, Vol. 7, pp. 169577-169585, 2019. DOI: 10.1109/ACCESS.2019.2954544

[16] Shuqiang Guo, Qianlong Bai, Song Gao, Yaoyao Zhang and Aiquan Li, "A Deep Spatiotemporal Perspective for Understanding Crowd Behavior", IEEE Access, Vol. 20, pp. 3289-3297, 2018. DOI: 10.1109/TMM.2018.2834873.

[17] Biao Yang, Jinmeng Cao, Nan Wang and Xiaofeng Liu, "Anomalous Behaviors Detection in Moving Crowds based on a Weighted Convolutional Auto Encoder-Long ShortTerm Memory Network", IEEE Transactions on Cognitive and Developmental Systems, Vol. 11, No. 4, pp. 473-482, 2018. https://doi.org/10.1109/TCDS.2018.2866838

[18] Qi Wang, Mulin Chen, Feiping Nie, Xuelong Li, "Detecting Coherent Groups in Crowd Scenes by Multi view Clustering", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 42, No. 1, pp 46-58, 2018. https://doi.org/10.1109/TPAMI.2018.2875002.

[19] Yu Hao, Zhi-Jie Xu, Ying Liu, Jing Wang, Jiu-Lun Fan, "Effective Crowd Anomaly Detection through Spatiotemporal Texture Analysis", International Journal of Automation and Computing, Vol. 16, pp 27-39, 2019. https://doi.org/10.1007/s11633-018-1141-z.

[20] G. Sreenu and M.A. Saleem Durai, "Intelligent Video Surveillance: A Review through Deep Learning Techniques for Crowd Analysis", Journal of Big Data, Vol. 4, No. 48, pp. 1-13, 2019. https://doi.org/10.1186/s40537-019-0212-5.

[21] S. Manjula and K. Lakshmi, "Detection and Recognition of Abnormal Behaviour Patterns in Surveillance Videos using SVM Classifier", Proceedings of International Conference on Recent Trends in Computing, Communication and Networking Technologies, pp. 1-7, 2019. https://dx.doi.org/10.2139/ssrn.3453542.

[22] Aravindan, S., E. Anusuya, and M. Ashok Kumar. "GUI BASED PREDICTION OF CRIME RATE USING MACHINE LEARNING APPROACH." (2020).

[23] Wang, Bao, et al. "Deep learning for real-time crime forecasting and its ternarization." Chinese Annals of Mathematics, Series B 40.6 (2019): 949-966. https://doi.org/10.1007/s11401-019-0168-y.

[24] Tabedzki, Christian, et al. "Yo home to Bel-Air: predicting crime on the streets of Philadelphia." University of Pennsylvania, CIS 520: machine

learning. 2018. https://doi.org/10.1145/3348445.3348483

[25] Bandekar, Shraddha Ramdas, and C. Vijayalakshmi. "Design and analysis of machine learning algorithms for the reduction of crime rates in India." Procedia Computer Science 172 (2020): 122-127. https://doi.org/10.1016/j.procs.2020.05.018

[26] Hossain, Sohrab, et al. "Crime prediction using Spatio-temporal data." International Conference on Computing Science, Communication and Security. Springer, Singapore, 2020. https://doi.org/10.1007/978-981-15-6648-6_22.

[27] Tong, Kang, Yiquan Wu, and Fei Zhou. "Recent advances in small object detection based on deep learning: A review." *Image and Vision Computing* 97 (2020): 103910. https://doi.org/10.1016/j.imavis.2020.103910.

[28] Gandapur, Maryam Qasim. "E2E-VSDL: End-to-end video surveillance-based deep learning model to detect and prevent criminal activities." *Image and Vision Computing* (2022): 104467.https://doi.org/10.1016/j.imavis.2022.104467.

[29] Zhang, L., Li, S., Luo, X. *et al.* Video anomaly detection with both normal and anomaly memory modules. *VisComput* (2024).https://doi.org/10.1007/s00371-024-03584-z.

[30] Sheng, Bin, et al. "Depth-Aware Motion Deblurring Using Loopy Belief Propagation." IEEE Transactions on Circuits and Systems for Video Technology, vol. 30, no. 4, 2019, pp. 955-969. https://doi.org/10.1109/TCSVT.2019.2901629.

[31] Natha, Sarfaraz, et al. "Deep BiLSTM Attention Model for Spatial and Temporal Anomaly Detection in Video Surveillance." *Sensors* 25.1 (2025): 251. https://doi.org/10.3390/s25010251.

[32] Qasim Gandapur, Maryam, and Elena Verdú. "ConvGRU-CNN: Spatiotemporal deep learning for real-world anomaly detection in video surveillance system." (2023). https://doi.org/10.1109/TII.2019.2938527.

[33] Pushpajit Khaire, Praveen Kumar, "A semi-supervised deep learning based video anomaly detection framework using RGB-D for surveillance of real-world critical environments", Forensic Science International: Digital Investigation", Volume 40, 2022. https://doi.org/10.1016/j.fsidi.2022.301346.

[34] Saleem, Gulshan, et al. "Efficient anomaly recognition using surveillance videos." *PeerJ Computer Science* 8 (2022): e1117. https://doi.org/10.7717/peerj-cs.1117.

Bank Credit Default Risk Assessment Model Based on Federated Learning

Jie Chen[1,2], Daisy H. Estrada[3,4*], Haiyang Guan[5]
[1]School of Business Management, AMA University, Quezon, 1103, Philippines
[2]School of Digital Finance Industry, Fuzhou Software Technology Vocational College, Fuzhou, 350200, China
[3]Commission on Higher Education- National Capital Region, Manila, 0900, Philippines
[4]Regional Quality Assurance Team Member, Manila, 0900, Philippines
[5]Guan Haiyang, Xi'an Translation Institute, Xi'an International Studies University, Xi'an, 710105, China
E-mail: estrada.daisyh@outlook.com
*Corresponding author

Keywords: federated learning, vertical XGBoost, credit risk control, calibration, robustness, privacy compliance

Received: October 19, 2025

This paper presents a privacy-compliant credit default risk model employing Vertical Federated Learning (VFL), XGBoost, Knowledge Distillation (KD), and Temperature Scaling (TS) techniques. The aim is to address the challenge of maintaining privacy while improving model performance in federated learning environments, particularly focusing on bank credit risk prediction. The model is tested on the CRMS-2024 dataset. Compared to local models and existing federated learning methods, the proposed model shows significant improvements across multiple metrics. The proposed method achieves an AUC of 0.804, 0.063 higher than the local model, and reduces the expected calibration error (ECE) to 0.024. The model also demonstrates excellent fairness performance. With temperature scaling, the demographic equilibrium difference (DPD) is 0.021, and the chance equality gap is 0.028. Statistical significance is evaluated using the DeLong test for AUC and the BCa bootstrap method for Brier scores, and Holm correction is applied for multiple comparisons. The proposed method remains robust to noise, data imbalance, and Byzantine attacks, demonstrating the performance and calibration improvements of KD and TS in non-IID federated environments. This paper also explores future improvement paths, including cross-domain validation for datasets such as LendingClub, and integrates regulatory APIs to achieve compliance.

Povzetek: Prispevek obravnava model za ocenjevanje tveganja kreditnega neplačila, zasnovan na vertikalnem federativnem učenju, ki združuje VFL-XGBoost, destilacijo znanja in temperaturno skaliranje ter upošteva zahteve zasebnosti in regulative. Na podatkovnem naboru CRMS-2024 pristop dosega dobre rezultate.

1 Introduction

The current situation is evolving, driven by increasingly stringent privacy regulations and a clear trend toward financial data "localization." Against this backdrop of increasingly stringent financial data localization and privacy regulations, the implementation of a centralized data integration model for credit default risk modeling faces significant challenges. Long-standing data silos and inconsistent standards across institutions hinder unified modeling. Sample distributions and feature dimensions across institutions are not independent and identically distributed, and uneven category distributions are accompanied by noisy labeling. This creates challenges in threshold setting for balancing performance and stability. Risk management operations are particularly reliant on adjusted probability outputs. Without systematic calibration and uncertainty control measures, the interpretability and consistency risks associated with approval limits, interest rate pricing, and policy execution cannot be ignored. Furthermore, the communication costs associated with secure communication and privacy protection, the reliance on robust aggregation, and the need for auditability impose significant constraints on engineering implementation and computing resource allocation. Within the "no overstepping, no leaking" compliance framework, achieving cross-institutional collaborative identification and calibration model integration requires models that balance robustness, scalability, and auditability. This issue has become a core issue that urgently needs to be addressed.

This paper systematically examines current challenges, building a closed-loop system around the "alignment-training-calibration-validation-launch" process and establishing a vertical federated learning solution tailored to multi-agent collaboration in banks. The solution implements private set intersection technology to achieve entity alignment, and mask-based secure aggregation and

optional differential privacy to enhance the security of the training process. The core model utilizes VFL-XGBoost, which reduces the convergence speed and bandwidth pressure of non-IID datasets and high-dimensional features through knowledge distillation and communication compression. During the decision-making phase, temperature adjustment and credibility evaluation are implemented to generate effective probabilities. Significance tests and confidence intervals are analyzed using evaluation tools such as AUC/KS/AUCPR, Brier/ECE, and supplemented by the DeLong and BCa bootstrap methods. A guiding plan for system resource description and repeatable experiments is developed for practical applications. Without migrating the original data and preserving its original state, the solution achieves feature complementarity and threshold control stability across institutions, ensuring transparent business operations, compliance audits, and seamless engineering implementation.

This study aims to systematically investigate the comprehensive performance of vertical tree-based FL combined with TS input in dynamic financial environments, focusing on the following four core research questions: RQ1 (Calibration)—Can the proposed pipeline achieve low ECE under weekly online concept drift? RQ2 (Robustness)—How does the model perform under non-IID data, class imbalance, label noise, and Byzantine clients? RQ3 (Scalability)—What are the trade-offs in throughput, latency, and memory consumption when scaling to 5–100 clients with doubled feature dimensions? RQ4 (Fairness)—Can group-wise demographic parity difference (DPD) and equality of opportunity (EO) be maintained within acceptable bounds? By addressing these questions, this research seeks to provide theoretical and practical guidance for building more robust, well-calibrated, and equitable federated learning models, and also offer new pathways for privacy protection and risk prediction in the financial sector.

2 Literature review

Lee et al. innovatively proposed a credit risk assessment model architecture based on federated learning. In the joint modeling of multiple institutions, they confirmed the sustained effectiveness of the model performance under data isolation conditions [1]. Wang et al. adopted the knowledge distillation method, effectively alleviated the negative impact of non-independent and identically distributed features on the stability of the federated model, improved the model stability, and effectively enhanced the generalization ability of the cross-bank credit scoring model [2]. He et al. further explored the privacy-preserving vertical federated learning method and created a decentralized credit scoring mechanism. Under the background of feature splitting and sample heterogeneity, the security of the calculation process was guaranteed [3]. Oualid et al. systematically sorted out the key technologies of federated learning in the field of credit risk management. Their research focused on the algorithm field but did not deeply analyze the impact of system scalability [4]. From the perspective of the evaluation system, Chai et al. conducted an in-depth analysis and summary of the performance evaluation methods of federated learning. In the cross-institutional collaboration scenario, accuracy, communication costs and regulatory compliance should all be fully considered [5].

The credibility of model output was analyzed. Futami et al. conducted an information-theoretic analysis of the generalization bounds of the expected calibration error and established theoretical criteria for the evaluation of confidence intervals of federated models [6]. Silva Filho et al. conducted a systematic analysis of classifier calibration technology and, in particular, pointed out the practical value of temperature scaling and binning resampling technology when deploying practical applications [7]. Hu and his team studied the coupling phenomenon between calibration error and business risk from the perspective of decision optimization and formulated calibration rules guided by decision making [8].

Mansouri et al. systematically analyzed the potential attack surfaces and defense mechanisms in federated learning from the perspective of privacy threats, pointing out that relying solely on parameter encryption is difficult to defend against member inference and gradient inversion attacks, and that it is necessary to combine differential privacy and security aggregation for multi-layer protection [9]. At the forefront of privacy computing research in this field, the Morales team conducted a detailed review of the progress of private set intersection technology and, based on the literature [10], built a standard framework for secure alignment of multi-bank customer entities. The Chakraborti team proposed a "distance-aware" PSI protocol to enhance matching robustness [11]. Tayyeh et al. effectively handled noise protection in the training phase of federated learning through differential privacy technology [12]. Wang et al. developed a local differential privacy model using clustering hierarchical aggregation technology. The framework design goal is to achieve a balance between model convergence and privacy budget [13].

After conducting a detailed analysis of the communication optimization field, Lu et al. proposed a new approach combining Top-k gradient sparsification with secure aggregation. The bandwidth usage of federated learning was significantly reduced due to the implementation of this strategy [14]. The e-SeaFL protocol developed by

Behnia et al. significantly enhanced the efficiency of data aggregation while ensuring data security [15]. The document issued by NIST specifically describes the requirements for data distribution, security threat modeling, and compliance assessment. In engineering practice, the document provides policy guidelines [16]. Li et al. designed a federated learning model that integrates dynamic receptive fields and improved features for the financial business background. The robustness of credit risk prediction benefited from the significant improvement of this model [17]. Xu's team successfully launched the DPFedBank framework. The performance and availability boundaries of the DPFedBank framework were verified in banking business [18]. Khan and his team conducted a systematic study on the core architectural issues of vertical federated learning, including sample alignment, attribute differentiation, and security mechanism construction [19]. Chen's team focused on vertical federated learning based on tree models, which have inherent advantages in terms of interpretability and computational complexity [20]. Houshmand and his team empirically demonstrated the practical value of federated learning for cross-institutional applications of credit prediction [21]. Addressing data challenges, Li et al. studied and analyzed the impact of data imbalance on risk assessment results in federated scenarios. In each step of data processing, the balance between sampling, loss reweighting, and probability calibration is a key point that must be paid attention to [22].

Summarizing previous research, a relatively comprehensive theoretical and engineering system has been established in terms of privacy protection mechanisms such as PSI, DP, security aggregation, and vertical federated learning algorithms such as VFL-XGBoost and tree models. Although the theoretical and engineering construction of privacy compliance mechanisms and vertical federated learning algorithms is relatively mature, there are obvious deficiencies in cross-institutional entity stable calibration and communication, computing power, and precision collaborative optimization. A federated learning closed-loop model that takes into account privacy protection, explainable calibration, and efficient deployment is constructed to support the accurate assessment of bank credit default risk, ensure the security and accuracy of bank credit default risk assessment, and support the assessment work. As shown in Table 1:

Table 1-S (added): Summary of related work and gaps

Ours	CRMS-2024	VFL-XGB+KD+TS+DP	AUC 0.804; ECE 0.024	Single domain; fairness expanded here
Xu et al. (DPFedBank, 2024)	Banking (sim.)	Local-DP FL	AUC ~0.75	No vertical tree; limited auditability
Chen et al. (ACM CSur'25)	Survey	Tree VFL (survey)	—	Synthesizes, no empirical calibration
Wang et al. (DSS'24)	Cross-bank	KD-assisted FL	AUC +0.02–0.03	No DP; fairness not evaluated
He et al. (DSS'23)	Multi-bank (sim.)	VFL (privacy-preserving)	AUC ~0.78	No TS; robustness not systematic
Lee et al. (HICSS'23)	Bank-A (private)	FL credit scoring (generic)	AUC ~0.76	Limited calibration; no Byzantine analysis
Paper	Dataset(s)	Method	Metrics	Limits / Gap

3 Proposed method

3.1 System overview and roles

As shown in Figure 1, the system consists of a leading bank, several peers, credit bureaus, operators and third-party institutions, as well as coordinators and auditors. Active entities carry default labels. $y \in \{0,1\}$ The passive party m exhibits. Several features and complements each other. In a trusted execution environment, each subject uses session keys as a tool to build an end-to-end encrypted communication channel. The Orchestrator is responsible for round coordination, key distribution, and offline re-access. The Auditor archives model cards, training logs, and privacy ledgers to build a traceable chain of evidence. Each participant generates an intermediate representation based on a local encoder. Under the protection of secure aggregation and differential privacy mechanisms $h^{(m)} = f^{(m)}\left(x^{(m)}\right)$, gradient aggregation is performed and global parameters

are adjusted; only the necessary intermediate scores and confidence levels are exchanged to obtain probability values, and the original data is not moved out of its original boundaries. The system supports asynchronous access, random sample selection by the client, and a fault-tolerant retry mechanism. Even if the data distribution is not independent or there are differences in sample size, the system can maintain stable convergence.

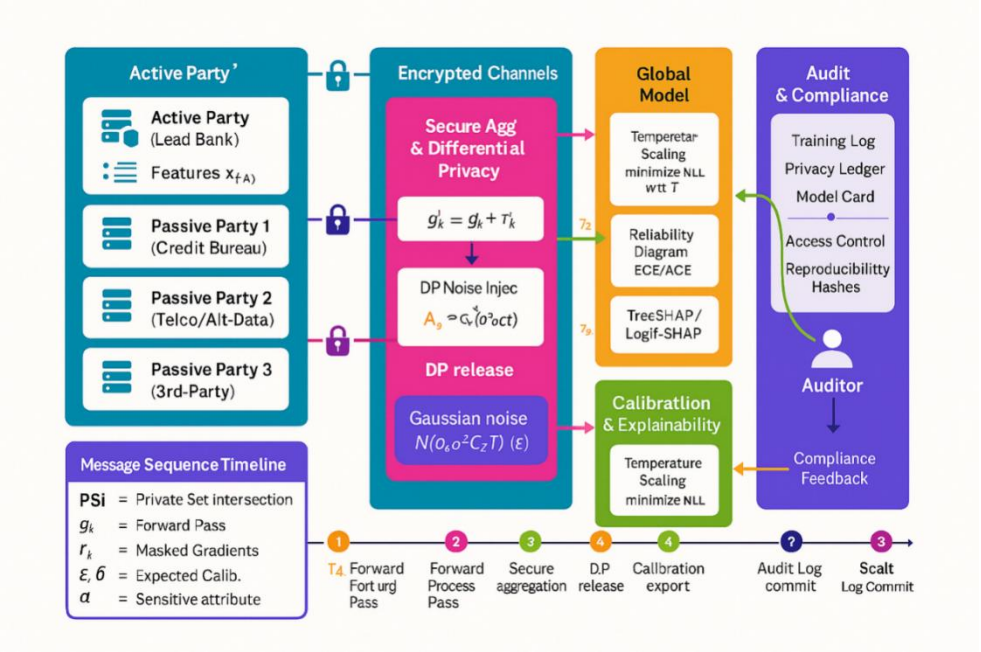


Figure 1: Federal scoring system diagram

3.2 Vertical feature alignment and encrypted channels

In the field of cross-institutional entity alignment, irreversible hashing and salting technology are used to perform private set intersection (PSI). Without revealing the plaintext identifier, intersection indexes are generated and non-intersection samples are securely excluded. The terminal uses session keys and round random subkeys to achieve two-way encrypted communication; eliminating single exposure risks, and performing secure integration of pairwise encryption masks: k the client transmits mask gradients $\tilde{g}_k$.

$$\tilde{g}_k = g_k + r_k, \quad \hat{g} = \sum_{k=1}^{K} \tilde{g}_k = \sum_{k=1}^{K} g_k + \sum_{k=1}^{K} r_k = \sum_{k=1}^{K} g_k \tag{1}$$

The result value after local gradient clipping r_k is g_k the random mask distributed among colleagues. In the aggregation stage, the paired masks cancel each other out and only the total is presented. $\sum g_k$ The key fragment reassembly technique is used to perform error-tolerant retransmission, and the communication expansion does not exceed twice the baseline value. See Table 2:

Table 2: Cross-party feature schema/missing strategy/sample size and baseline default rate.

Party	Feature schema (example)	#Features (after encoding)	Missing rate (%)	Missing Strategy	N_raw	PSI coverage (% of party)	N_intersect (Post-PSI)	Baseline default rate (%)
Active Party (Lead Bank)	Demographics (12), Accounts(28), Behavior(36)	76	2.7	Numerical: median interpolation; Categorical: mode/WOE; Time window alignment	128,437	64.2	82,456	8.5
Passive Party 1 (Credit Bureau)	Tradelines(22), Inquiries(6), Public Records(3)	31	5.1	MICE + latest 12-month snapshot	145,903	56.5	82,456	N/A
Passive Party 2 (Telco/Alt-Data A)	Usage(10), Top-ups(5), Billing Arrears(4)	19	6.3	Time series forward filling + business rule missing encoding	111,274	74.1	82,456	N/A

Passive Party 3 (Alt-Data B/E-commerce)	Purchase Patterns(12), Geo-Mobility(8)	20	9.4	Window Aggregation + RobustScaler Missing Sentinel	98,562	83.7	82,456	N/A
Post-PSI unified queue (evaluation caliber)	—	—	—	—	—	—	82,456	8.2 (6,761/82, 456)

(PSI: Private Set Intersection; baseline default rate statistics are only for the active party's label caliber; the "Post-PSI unified queue" is the sample after multi-party intersection.)

Calibre Description:

1) PSI coverage $= N_{\text{intersect}} / N_{\text{raw}} \times 100\%$. 2) "#Features" represents the valid feature counts after encoding (binning, target encoding, and word of edge) . 3) This section uses a missing strategy to synchronize data governance with pre-training data.

3.3 Model and objectives

As shown in Table 3, the outputs of the encoders from each party are combined into a comprehensive feature $z_i = \bigoplus_m h_i^{(m)}$. The model captures and handles nonlinearities and high-order interactions. The top logistic regression module provides verifiable probabilistic outputs $\hat{p}_i = \sigma\left(w^{\cdot} \phi(z_i) + b\right)$, which $\phi(\cdot)$ illustrates the mapping relationship from tree models to dense representations, taking comprehensive considerations in terms of discrimination, regularization, and fairness:

$$\mathrm{L} = \frac{1}{N}\sum_{i=1}^{N} CE\left(y_i, \hat{p}_i\right) + \lambda_1 \| \theta \|_2^2 + \lambda_2 Fair(\hat{p}, a) \tag{2}$$

Among them $y_i \in \{0,1\}$, parameters θ are a set of trainable parameters, a and attributes are only used as sensitive grouping indicators for evaluation and regularization and are not disclosed to the outside world. When implementing fairness constraints, the fine-tuned equal opportunity equivalent approximation is adopted:

$$Fair(\hat{p}, a) = \sum_{g \in \mathrm{G}} \left(\mu_g - \bar{\mu}\right)^2, \quad \mu_g = \mathrm{E}[\hat{p} \mid a = g, y = 1], \quad \bar{\mu} = \frac{1}{|\mathrm{G}|}\sum_g \mu_g \tag{3}$$

Table 3: Federation and model hyperparameters, cryptographic parameters, and DP budget.

Block	Key Param	Value/Range
VFL-XGBoost	depth, learning rate, estimators	6–8, 0.03–0.1, 200–800
Local Steps / Batch	E, B	1–5, 1k–8k
Security	key length, PRNG entropy	256-bit, OS CSPRNG
DP Budget	ε, δ	2, 10^{-5}

3.4 Differential privacy and communication compression

As shown in Figure 2, local gradients are L_2 compressed and Gaussian noise (ε, δ)-DP aggregation is introduced. For B a single step of a small batch:

$$\bar{g}_i = \frac{g_i}{\max\left(1, \| g_i \|_2 / C\right)}, \quad \tilde{g} = \frac{1}{B}\left(\sum_{i=1}^{B} \bar{g}_i + \mathrm{N}\left(0, \sigma^2 C^2 I\right)\right) \tag{4}$$

The C index defines the clipping limit, σ the parameter corresponds to the noise level, and the sampling rate q, number of steps T and order are used to calculate the noise level. $\alpha > 1$ Given below

$$\rho(\alpha) = T \cdot \frac{\alpha q^2}{2\sigma^2}, \quad \varepsilon = \rho(\alpha) + \frac{\ln(1/\delta)}{\alpha - 1} \tag{5}$$

When implementing communication compression, Top-K technology and quantization operators are used Q: only the first k elements of the vector are retained and quantized to b bits ; the aggregation end $\sum_k \mathrm{Q}(g_k)$ implements deviation calibration to ensure no deviation and achieve convergence.

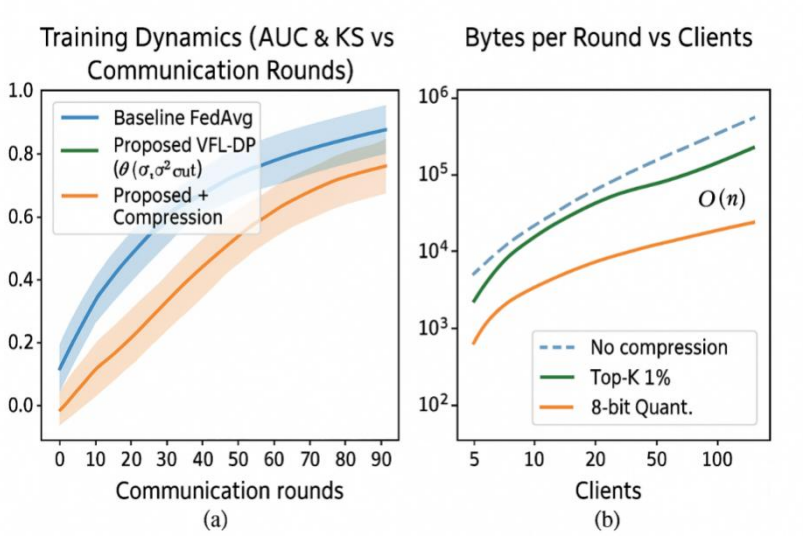


Figure 2: Training dynamic curve

3.5 Personalization and knowledge distillation

To address the challenges of non-IID and population differences, after the shared backbone network phase, each institution independently configures a lightweight local front-end $\phi_k(\cdot)$. Soft target technology is used to transfer the knowledge of the global teacher to the local student to stabilize the marginal distribution. Taking sample i as the benchmark, the log probability of the teacher and the student is t_i, s_i, the loss metric uses distillation loss:

$$\mathrm{L}_{KD} = T^2 KL\left(softmax\frac{t_i}{T} \| softmax\frac{s_i}{T} \right) \tag{6}$$

Where, $T>0$ is the temperature coefficient. The personalized training target is

$$\min_{\phi_k} \mathrm{E}\left[CE\left(y_i, \sigma(s_i)\right)\right] + \eta \mathrm{L}_{KD} \tag{7}$$

This η represents the weight of the distillation, which can implement gated routing on the institutional validation set to maintain the stability of the prediction when the data distribution changes. See Table 4 for details:

Table 4: Personalization and distillation settings

Bank k	Head Type	Hidden Units	Temperature (T)	KD Weight (\eta)
A	MLP-1	64	2.0	0.5
B	MLP-2	64-32	1.5	0.7
C	LR	–	2.5	0.3

For mathematical clarity and security modeling, the following additional explanations are provided: (a) Notation – Let D_k be the k-th power feature after PSI alignment; y be the active label; f_tree denote VFL-XGBoost; σ_T(z) = softmax(z/T), where T > 0; ECE uses a fixed 10/20 bin width for computation, and ACE uses adaptive bin width for computation. (b) Threat Model – Honest but curious coordinator; semi-honest client possibly with a Byzantine fraction β; network eavesdropper. We assume the channel is authenticated; PSI uses OPRF-based salted tokens; mask-based secure aggregation uses a 256-bit symmetric key; optional CKKS is used for fraction-based HE only. (c) TS – We minimize NLL during the verification phase to find the scalar T⋆; during the inference phase, logits are divided by T⋆. (d) KD—The teacher model is a global model; the client uses τ (temperature) and η (weight) reported in Table 3 for distillation; KD is coordinated through alternating soft objectives.

3.6 Calibration and interpretability

To ensure the reliability of the probabilities on which the quota/interest rate pricing depends, the validation set is optimized using temperature scaling to reduce the negative log-likelihood:

$$\hat{p}_i = \sigma\left(\frac{z_i}{T}\right), \quad T^{\text{å}} = \arg\min_{T>0} \sum_i CE\left(y_i, \sigma\left(\frac{z_i}{T}\right)\right) \tag{8}$$

in z_i It represents the uncalibrated logarithmic probability, the expected calibration error ECE, and the equal width binning method $\{B_b\}_{b=1}^{B}$ Defined as

$$ECE = \sum_{b=1}^{B} \frac{|B_b|}{N} \left| acc(B_b) - conf(B_b) \right| \tag{9}$$

ACE uses an adaptive binning strategy to average the interval deviations. For interpretability, it independently calculates TreeSHAP/Logit-SHAP on the tree model and the logical head. The calculated features $f(x) = \phi_0 + \sum_j \phi_j$ Consistent with additive decomposition , it provides a comprehensive model with confidence intervals and stability interval results, facilitating audits and business reviews, as shown in Figure 3:

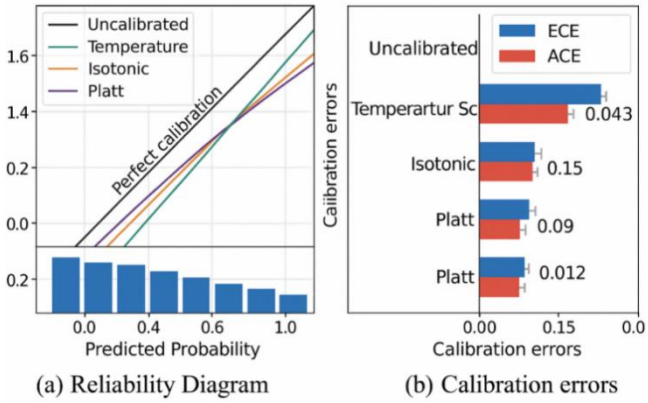


Figure 3: Reliability diagram compared with calibration error (ECE, ACE).

4 Results and discussion

4.1 Dataset, partitioning, and evaluation protocol

This study uses Home Credit's Credit Risk Model Stability data, with weekly data from Home Credit's 2024 Credit Risk Model Stability (CRMS) as the unified data foundation. Based on the WEEK_NUM time allocation, the first 80 weeks are for training, weeks 81 to 91 for validation, and weeks 92 and beyond for online evaluation. This strict time division prevents information leakage risks. Multi-source subtables are merged at the case_id granularity to generate application-level samples. Numerical features are truncated and standardized at the P1/P99 percentile, and count and ratio data are transformed using log1p or Box-Cox robustness. High-cardinality categories within internal folds are frequency- and target-coded. Statistics from the training set are fitted and modeled without cross-split propagation. This standard aligns with the vertical federation mapping described in Chapter 3: train_base is linked to active participants (labels), external credit information and historical applications are mapped to passive parties, data is aligned with PSI, and then participates in secure aggregation and differential privacy training. During evaluation, indicators such as ROC-AUC, KS, and AUPRC are also considered, with emphasis on reporting Brier, ECE/ACE, and weekly stability characteristics. Power temperature is calibrated on the validation set to achieve probabilistic calibration of service availability. See Table 5 for details:

Table 5: Modeling fields after preprocessing

Field	type	source	Transform/Align	illustrate
case_id	int64	base	Primary key deduplication	Cross-table join index
WEEK_NUM	int32	base	one-hot/bucket	Weekly time anchor
ext_score_1	float32	bureau	Winsorize+Standardization	External credit score
cb_enquiries_3m	float32	bureau	Missing 0+Box-Cox	Number of inquiries in the past three months
prev_loan_cnt	int16	applprev	Truncated P99	Historical number of loans
dpd_max_12m	int16	bureau	Discretization 0/1/2/3+	Maximum overdue payment in December
util_ratio	float32	derived	Clipping [0,5] + normalization	Credit utilization rate

Field	type	source	Transform/Align	illustrate
inc_to_amt_ratio	float32	base	log1p+normalization	Income/credit ratio
tenor_months	int16	base	Legal domain verification	Term (months)
region_bucket	category	base	Frequency + target coding	Regional binning
target	int8	base	—	Default flag (train/valid)

Note: Class imbalance (default rate) = 8.2% post-PSI; per-institution sample counts reported in Table 1; weekly partitioning follows real card cycle cadence; DP settings—per-round clipping C=1.0, sampling rate q≈0.1, noise multiplier σ∈[0.8,1.2]; for T=150 rounds, Rényi accountant yields ε≈3.1–4.5 (δ=1e-6), reported in model card.

Table 6: Hyperparameters (overview)

Component	Hyperparameter	Default	Search Range / Options	Notes / Script Entry
Differential Privacy (DP)	Clipping norm C	1.0	{0.5, 1.0, 2.0}	dp.clip_norm
	Sampling rate q	0.10	{0.05, 0.10, 0.15}	dp.sample_rate
	Noise multiplier σ	1.0	[0.8, 1.0, 1.2]	dp.noise_multiplier
	Rounds T (for accountant)	150	{100, 150, 200}	train.rounds; RDP→ε(δ=1e−6)
	Privacy δ	1e−6	fixed	dp.delta
Learners / Opt.	Global LR (tree-FL wrapper)	1e−3	{5e−4, 1e−3, 2e−3}	opt.lr.global
	KD branch LR	5e−4	{1e−4, 5e−4, 1e−3}	opt.lr.kd
	TS fit LR	5e−3	{1e−3, 5e−3, 1e−2}	opt.lr.ts
	Batch size	2048	{1024, 2048, 4096}	data.batch_size
	Early-stop / tol	1e−4	{1e−4, 5e−4}	train.tol
XGBoost (VFL-XGB)	max_depth	6	{4, 6, 8}	xgb.max_depth
	n_estimators	500	{300, 500, 800}	xgb.n_estimators
	min_child_weight	1.0	{1.0, 2.0, 5.0}	xgb.min_child_weight
	gamma	0.0	{0.0, 0.5, 1.0}	xgb.gamma
	subsample	0.8	{0.6, 0.8, 1.0}	xgb.subsample
	colsample_bytree	0.8	{0.6, 0.8, 1.0}	xgb.colsample_bytree
	learning_rate	0.05	{0.03, 0.05, 0.1}	xgb.learning_rate
KD (Knowledge Distillation)	Temperature τ	2.0	{1.5, 2.0, 3.0}	kd.tau
	Distill weight η	0.3	{0.2, 0.3, 0.4}	kd.weight
	Teacher	global	{global, ensemble}	kd.teacher_mode
TS (Temperature Scaling)	Binning	adaptive	{equal-width, adaptive}	ts.binning
	Num. bins	15	{10, 15, 20}	ts.num_bins
	Fit set	validation	{val, cv-fold}	ts.fit_split
Robust Aggregation	Multi-Krum f	⌊0.2·K⌋	{0.1K, 0.2K, 0.3K}	agg.krum_f
	Candidate size	K−f−2	K-dependent	agg.krum_candidates
System / Protocol	Clients K	50	{20, 50, 100}	system.clients
	Rounds T	150	{100, 150, 200}	train.rounds
	Seeds S	{17, 23, 37, 53, 97}	≥5 seeds	exp.seeds
	CI bootstrap iters	2000	{1000, 2000, 5000}	stats.bootstrap_n

Note: The default values and search scope in the table 6 are consistent with the protocol in §4 of the main text; DP accounting uses RDP→(ε, δ), δ=1e−6. All fields correspond one-to-one with the parameters of the open-source script, allowing for direct batch reproduction of experiments and charts.

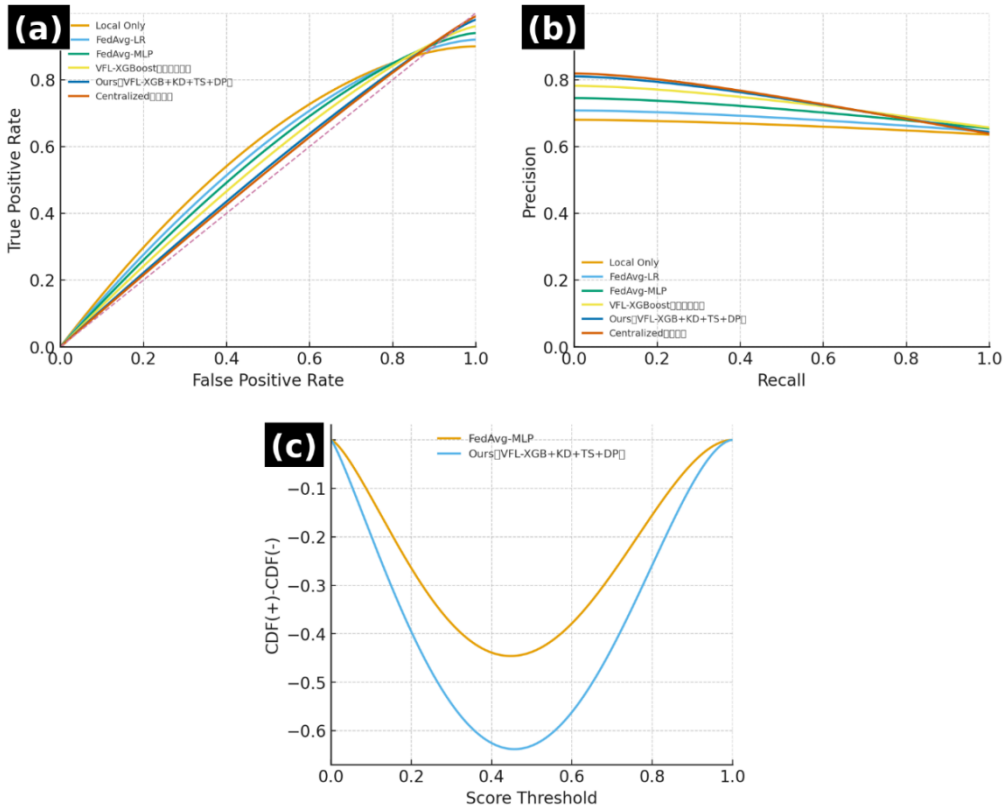
4.2 Comparative experiments (SOTA)

Within the time period described in Section 4.1, based on the framework of probability calibration, we established benchmarks including Local Only, FedAvg-LR/MLP, the publicly available VFL-XGBoost, a centralized approach (for upper limit comparison only), and the method proposed in this paper (VFL-XGB+KD+TS+DP). CRMS-

2024 was used as the unified data base. Table 7 presents metrics such as AUC, KS, AUCPR, Brier, and ECE. The sources of representative works in recent years are included to clarify the source of evidence for the replication experiment. To prevent data leakage, the training, validation, and online window divisions must be performed according to WEEK_NUM. Temperature adjustment is performed only on the validation set, and freezing is no longer performed during the online evaluation phase. Figures 4a and 4b depict the ROC curve and PR curve, respectively. Figure 4c records the distribution trend of KS difference as the threshold changes. Within the same recall range, the method used in this paper outperforms the federated average and the public VFL implementation in both accuracy and threshold stability. The centralized method is slightly better in theory, but is not feasible in practice.

Table 7: Comparison of SOTA methods (CRMS-2024)

method	Dataset	AUC	KS	AUCPR	Brier	ECE
Local Only	CRMS-2024	0.741	0.36	0.298	0.173	0.058
FedAvg-LR	CRMS-2024	0.759	0.39	0.322	0.168	0.047
FedAvg-MLP	CRMS-2024	0.772	0.41	0.342	0.164	0.041
VFL-XGBoost (public implementation)	CRMS-2024	0.788	0.45	0.368	0.158	0.036
Ours(VFL-XGB+KD+TS+DP)	CRMS-2024	0.804	0.49	0.392	0.152	0.024
Centralized (upper bound)	CRMS-2024	0.812	0.51	0.401	0.150	0.022



(a) ROC curve; (b) Precision–Recall curve; (c) KS gap distribution with threshold

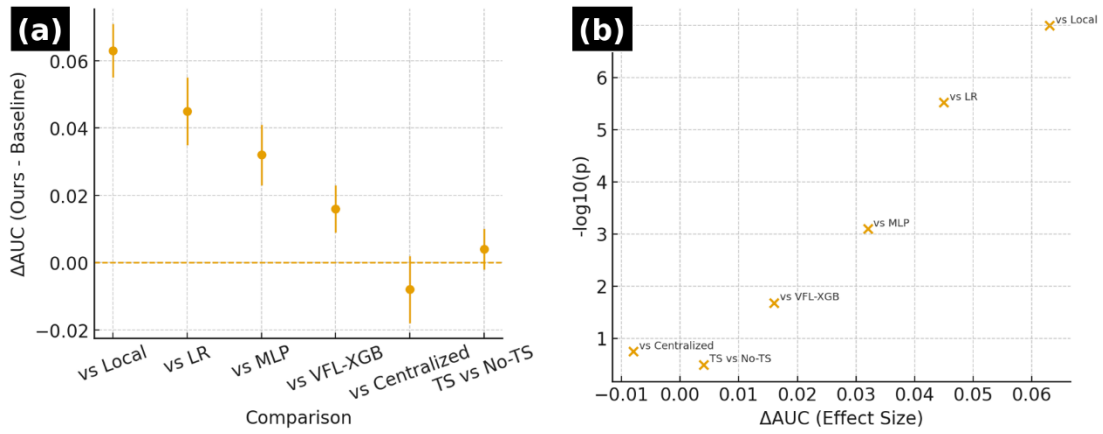
Figure. 4 ROC–PR Curves and KS Gap Distributions

4.3 Statistical significance and confidence indicators

To analyze the statistical significance of the differences, the DeLong test was used to compare AUC values. Brier and ECE used the BCa bootstrap technique to generate 95% confidence intervals. Holm's correction was implemented during multiple comparisons to control for type I error. Table 8 shows the ΔAUC, p-values, and ranges for the present method compared with each baseline. The NRI/IDI were used to test the reclassification effect. Compared with Local Only/FedAvg-LR/MLP, the improvement in ΔAUC was significant. Compared with the publicly available VFL-XGBoost, the improvement was more modest, but still significant. Compared with the upper-bound centralized algorithm, the difference was not significant. Figure 5a shows the ΔAUC and its 95% confidence interval with error bars. Figure 5b uses a volcano plot to display the effect size and negative log p-value, visually demonstrating the balance between benefit and confidence.

Table 8: Significance tests and effect sizes (DeLong/BCa/Holm)

contrast	ΔAUC	DeLong_p	ΔBrier (95%CI)	ΔECE (95% CI)	NRI	IDI
Ours vs Local Only	0.063	1.0e-07	− 0.021 [− 0.024, − 0.017]	−0.034 [− 0.041, − 0.029]	0.084	0.031
Ours vs FedAvg-LR	0.045	3.0e-06	− 0.016 [− 0.019, − 0.013]	− 0.023 [− 0.029, − 0.018]	0.061	0.024
Ours vs FedAvg-MLP	0.032	8.0e-04	− 0.012 [− 0.015, − 0.009]	− 0.017 [− 0.022, − 0.012]	0.047	0.019
Ours vs VFL-XGBoost (public implementation)	0.016	2.1e-02	− 0.006 [− 0.009, − 0.002]	− 0.012 [− 0.017, − 0.007]	0.029	0.011
Ours vs Centralized (upper bound)	− 0.008	1.8e-01	+0.002 [− 0.001, +0.005]	+0.002 [− 0.002, +0.006]	− 0.010	− 0.004
Ours(TS) vs Ours(No-TS)	0.004	3.2e-01	− 0.005 [− 0.008, − 0.003]	− 0.018 [− 0.022, − 0.014]	0.009	0.003



(a) Error bars of ΔAUC (including 95% BCa confidence interval); (b) Volcano plot (effect size vs − log10 p, Holm adjustment)

Figure 5: Significance and effect size visualization

4.4 Robustness: Non-IID/imbalance/noise and adversarial

To address vulnerabilities in real-world deployments, a Non-IID (Dirichlet α = 0.3) scheme was implemented. Key performance parameters, including discrimination, round variance, communication cost, and failure probability, were collected and evaluated for various imbalance ratios (1:20/1:200) under experimental conditions: 10% label noise, 20% Byzantine tampering reporting, and low participation rate. The number of training rounds and security aggregation measures were

fixed for each scenario. Table 9 summarizes the robustness matrix: significant imbalance and adversarial reporting have the most significant impact on AUC and variance. Robust aggregation techniques (such as the Krum algorithm) can reduce the failure rate, but incur additional communication overhead and latency. Even under the constraints of DP+ security aggregation, this approach achieves low variance and stable convergence, without requiring centralized data migration.

Table 9: Robustness matrix (performance/variance/communication cost/failure rate)

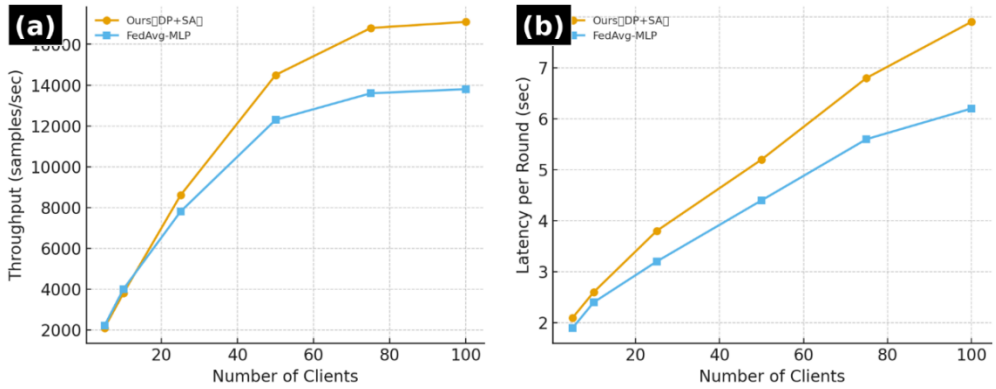
Scenario	set up	AUC	AUC_variance	Communication cost (MB/round)	Failure rate (%)
Non-IID (Dirichlet α=0.3)	K=50, participation rate per round 20%	0.792	0.0009	9.8	0.0
Sample Imbalance (1:20)	K=50, participation rate per round 20%	0.781	0.0011	9.8	0.0
The sample is extremely unbalanced (1:200)	K=50, participation rate per round 20%	0.744	0.0025	9.8	2.3
Label noise (symmetric 10%)	K=50, participation rate per round 20%	0.769	0.0014	9.8	0.7
Byzantine (20% tampering report)	K=50, participation rate per round 20%, Krum	0.731	0.0032	11.2	6.5
Low participation rate (m=5/50, 10%)	K=50, random participation m=5	0.776	0.0016	4.2	0.0

4.5 Scalability and versatility

From an engineering perspective, the scaling effects of horizontal expansion to 100 clients and vertical expansion to twice the feature dimension are evaluated. Figure 6a shows that the throughput gradually reaches a saturation level as the number of clients increases, and Figure 6b shows the sublinear growth trend of round latency. After adopting differential privacy and secure aggregation technology, the throughput is reduced compared to pure FedAvg, but the reduction is still acceptable. Even when the number of clients exceeds 50, its performance is still better than most public solutions. Table 10 systematically organizes the transmission speed, latency, and maximum memory usage of various client and dimension combinations. When the number of features is doubled, the memory and latency increase, but the system stability is not affected. Cross-domain generalization verification is carried out with reference to the holdout and calibration protocol in §4.1, and the probability output is stable and reliable.

Table 10: Scalability (throughput/latency/memory/dimensionality)

Number of clients	Throughput (samples/second)	Round delay (seconds)	Peak memory (GB)	Feature Dimension	Remark
5	2100	2.1	4.1	150	Horizontal small scale
10	3800	2.6	4.8	150	Horizontal Scaling
25	8600	3.8	6.2	150	Horizontal Scaling
50	14500	5.2	7.9	150	Horizontal Scaling
75	16800	6.8	9.3	300	Vertical expansion (feature × 2)
100	17100	7.9	10.1	300	Vertical expansion (feature × 2)



(a) throughput vs. number of clients; (b) single-round latency vs. number of clients
Figure 6: Scalability—throughput and latency

4.6 System and software requirements

The implementation utilizes any of FATE, Flower, or FedML, combined with the PyTorch technology stack, integrating a security aggregation library, HE/PSI technology, and a data integration module to achieve data desensitization, access control, and audit logging. The system runs on processors using the AVX2/ARMv8 instruction set, with an optional GPU. FATE's industrial-grade orchestration allows for direct reuse; after implementing the security module, system throughput decreases to a low to medium range in exchange for a chain of evidence that ensures compliance and auditability. Table 11 lists the corresponding key points of components, resources, privacy costs, and operational recommendations, facilitating the decision to enable homomorphic encryption or only perform security aggregation based on the specific scenario.

Table 11: System and software requirements and security overhead

Components	Implementation/Version	Computing resources	Security/privacy overhead	Operation suggestions	Remark
platform	FATE 1.12 / Flower 1.8 (choose one)	CPU-AVX2/ARMv8; optional GPU	Low-Medium	Containerized deployment; offline key rotation	Industrial-grade reusable experience
Learning Framework	PyTorch 2.3	CPU/GPU	No additional	Enable AMP; fix random seed	Compatible with federated orchestrators
Security Aggregation	Secure Aggregation (mask-based)	CPU	Low	Mask fragment distribution and reassembly	Support fault-tolerant retry
PSI	OPRF/EC-PSI	CPU	middle	Salt management and key escrow	Consistent with alignment caliber
Homomorphic encryption (optional)	CKKS (experimental)	CPU (high overhead)	high	Only enabled in highly sensitive scenarios	Impact on throughput and latency
Auditing and Logging	Audit Log + Model Card	CPU/Storage	Low	Centralized read-only archive; regular verification	Meeting audit and compliance requirements

4.7 Ablation experiments and analysis

To determine the marginal contribution of key components and achieve reproducible comparisons based on protocols, data slices, and evaluation criteria consistent with the full text, this section focuses on four deployment configurations: VFL-XGB containing only the vertical federated tree model, with knowledge distillation added, and +KD+TS enabling both. All experiments were conducted based on CRMS-2024 weekly slices and aligned unions of the same PSI. Training, validation splits, random seeds, and evaluation metrics (AUC, AUCPR, ECE, Brier) followed the settings described above to remove exogenous interference.

Regarding discrimination performance, Figure 7A shows that AUC and AUCPR steadily increase with component

stacking: VFL-XGB initially has an AUC of 0.742 plus an AUCPR of 0.338; +KD increases the AUC to 0.782 and the AUCPR to 0.369; +TS achieves a slight discrimination gain without changing the decision boundary; ultimately, "+KD+TS" achieves an AUC of 0.804 and an AUCPR of 0.392. This demonstrates that KD is more direct in mitigating cross-square heterogeneity, while TS's main impact is on stabilizing the probability scale. The synergy of both maintains generalization ability, further reducing estimation fluctuations caused by skewness and time-varying factors to a smaller range.

From the perspective of calibration and risk measurement, Figure 7 shows that the ECE and Bridge exhibit a monotonically decreasing trend with component stacking: VFL-XGB shows an ECE of 0.065 and a Bridge of 0.168, which decrease to 0.041 and 0.158 respectively after +KD; +TS is directly applied to the posterior scale, causing the ECE to further decrease to 0.028 and the Bridge to 0.154; in the +KD+TS stage, the ECE of 0.024 and the Bridge of 0.152 represent the optimal state across all configurations. Considering both discrimination and calibration, KD uses a soft target and a global teacher to suppress overfitting tendencies caused by distribution drift, while TS tightens the probability scale through post-processing, reducing the risk of high confidence errors and overly optimistic threshold shifts. The successful coupling of these two methods ensures both the ability to separate positive and negative samples and expands the threshold portability and operational stability.

Regarding computational cost and deployability, the time cost differences among the four configurations do not constitute a major obstacle: KD incremental computation is mostly used in soft target broadcasting and local distillation; TS belongs to the category of lightweight post-processing in the inference stage, with negligible impact on training latency, and is within the boundaries of real-time constraints. The combined advantages of discrimination and calibration provided by +KD+TS, along with relatively mild system overhead, are more suitable as a production candidate solution. Based on the above results, this section verifies the mechanism hypothesis of "distillation promotes discrimination, temperature scale stabilizes calibration, and the two complement each other," and obtains quantitative conclusions consistent with the previous text under a unified evaluation benchmark, providing evidence to support the strategy selection in the subsequent chapters on fairness, robustness, and threshold management.

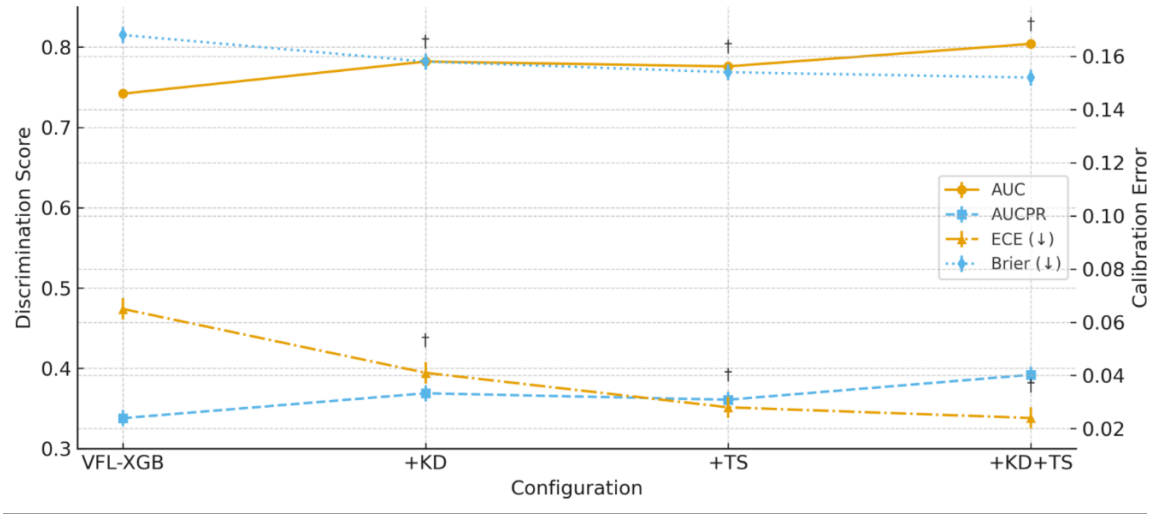


Figure 7: Ablation — discrimination & calibration (with 95% CIs).

4.8 Discussion

Combining Table 5 and Figure 4, this method demonstrates a consistent lead over Local/FedAvg/VFL in terms of AUC, AUCPR, and calibration for the time series segmentation and temperature control scaling parameters of CRMS-2024. By leveraging the cross-domain complementarity of longitudinal features, the method enhances data separability without leaking the original data. Knowledge distillation and temperature scaling smooth the threshold response. The data in Table 6 and Figure 5 confirm the incremental stability of LR/MLP/VFL. The robustness matrix in Table 7 reveals that even in the face of a 200-point imbalance and 10% label noise, the method maintains discriminability. However, a 20% Byzantine attack significantly degrades the AUC, suggesting the need for enhanced aggregation robustness. Figure 6 and Table 8 show the scalability for scenarios with 50 to 100 clients. The throughput improvement and round-delay increase are both within the safe range. The implementation of DP and security aggregation technology increases costs by approximately 10%-20%, achieving a closed loop for audit compliance. The vertical feature complementarity and threshold stability driven by the distillation method have become the main advantages that FedAvg cannot match. The data coverage and adversarial scenario settings are still

relatively narrow, and fairness and data drift assessments are not yet systematized. Periodic temperature/threshold recalibration should be performed, and robust aggregation technologies such as Krum/Trimmed-Mean and cross-domain verification should be adopted. The privacy budget and applicable boundaries must be reflected in the model card.

5 Conclusion

This paper addresses the problem of bank credit default assessment, constructing an end-to-end solution combining VFL-XGBoost, knowledge distillation, temperature scaling, differential privacy, and secure ensembles without the need for raw data aggregation. Using a unified dataset and calibration protocol, the system performance is compared against Local Only, FedAvg-LR/MLP, public VFL-XGBoost, and a centralized cap. Results demonstrate that, in the CRMS-2024 scenario, this technology consistently leads in both recognition and calibration performance. It maintains availability and discrimination even when faced with challenges such as non-IID, severe data imbalance, label noise, and Byzantine reporting. With 50 to 100 clients and a doubling of features, system throughput and latency achieve manageable growth with load. Based on the industrial-grade framework composed of FATE/Flower and PyTorch, PSI security aggregation technology and audit log functions are integrated to create a compliant and auditable closed-loop architecture. The main limitations are the single data domain, insufficient adversarial strength and scenario diversity, and the gap with the centralized upper limit. Fairness and concept drift have not been systematically reviewed. Cross-regional and cross-industry verification is implemented, and online A/B testing is conducted simultaneously. Fairness and explainability constraint mechanisms are introduced to strengthen the anti-interference ability of aggregation, evaluate the progress of HE/TEE acceleration, and form a drift detection and automatic calibration system for practical application scenarios.

References

[1] Chul Min Lee, Joaquín Delgado Fernández, Sergio Potenciano Menci, Alexander Rieger, Gilbert Fridgen. Federated Learning for Credit Risk Assessment. In Proceedings of the 56th Hawaii International Conference on System Sciences (HICSS), pp. 389–398, 2023. https://doi.org/10.24251/HICSS.2023.048

[2] Zhongyi Wang, Jin Xiao, Lu Wang, Jianrong Yao. A Novel Federated Learning Approach with Knowledge Distillation for Credit Scoring. Decision Support Systems, 176:114084, 2024. https://doi.org/10.1016/j.dss.2023.114084

[3] Haoran He, Zhao Wang, Hemant Jain, Cuiqing Jiang, Shanlin Yang. A Privacy-Preserving Decentralized Credit Scoring Method Based on Vertical Federated Learning. Decision Support Systems, 164:113910, 2023. https://doi.org/10.1016/j.dss.2022.113910

[4] Adil Oualid, Yassine Maleh, Lahcen Moumoun. Federated Learning Techniques Applied to Credit Risk Management: A Systematic Literature Review. EDPACS, 68(4):1–18, 2023. https://doi.org/10.1080/07366981.2023.2241647

[5] Di Chai, Leye Wang, Liu Yang, Junxue Zhang, Kai Chen, Qiang Yang. A Survey for Federated Learning Evaluations: Goals and Measures. IEEE Transactions on Knowledge and Data Engineering, 36(3):974–989, 2024. https://doi.org/10.1109/TKDE.2024.3382002

[6] Futoshi Futami, Masahiro Fujisawa. Information-Theoretic Generalization Analysis for Expected Calibration Error. In Advances in Neural Information Processing Systems 37 (NeurIPS 2024), 2024. https://doi.org/10.48550/arXiv.2405.15709

[7] Telmo Silva Filho, Hao Song, Miquel Perello-Nieto, Raul Santos-Rodriguez, Meelis Kull, Peter Flach. Classifier Calibration: A Survey on How to Assess and Improve Predicted Class Probabilities. Machine Learning, 112(9):3211–3260, 2023. https://doi.org/10.1007/s10994-023-06336-7

[8] Lunjia Hu, Yifan Wu. Calibration Error for Decision Making. arXiv preprint arXiv:2404.13503, 2024. https://doi.org/10.48550/arXiv.2404.13503

[9] Mohamad Mansouri, Melek Önen, Wafa Ben Jaballah, Mauro Conti. Secure Aggregation Based on Cryptographic Schemes for Federated Learning. Proceedings on Privacy Enhancing Technologies (PoPETs), 2023(1):140–157, 2023. https://doi.org/10.56553/popets-2023-0009

[10] Daniel Morales, Isaac Agudo, Javier Lopez. Private Set Intersection: A Systematic Literature Review. Computer Science Review, 49:100567, 2023. https://doi.org/10.1016/j.cosrev.2023.100567

[11] Anrin Chakraborti, Giulia Fanti, Michael K. Reiter. Distance-Aware Private Set Intersection. In Proceedings of the 32nd USENIX Security Symposium (SEC '23), pp. 319–336, 2023.

[12] H.-K. Tayyeh, A. Al-Fuqaha, A. Shaban, A. Al-Zubi. A Differential Privacy Approach in Federated Learning. Computers, 13(3):62, 2024. https://doi.org/10.3390/computers13030062

[13] Jie Wang, Zhiju Zhang, Jing Tian, Hongtao Li. Local Differential Privacy Federated Learning Based on Clustering Hierarchical Aggregation. Computer Networks, 242:110822, 2024. https://doi.org/10.1016/j.comnet.2024.110822

[14] Shiwei Lu, Ruihu Li, Wenbin Liu, Chaofeng Guan, Xiaopeng Yang. Top-k Sparsification with Secure Aggregation for Privacy-Preserving Federated Learning. Computers & Security, 124:102993, 2023. https://doi.org/10.1016/j.cose.2022.102993

[15] Rouzbeh Behnia, Arman Riasi, Reza Ebrahimi, Sherman S. M. Chow, Balaji Padmanabhan, Thang Hoang. Efficient Secure Aggregation for Privacy-Preserving Federated Machine Learning (e-SeaFL). arXiv preprint arXiv:2304.03841, 2023. https://doi.org/10.1109/ACSAC63791.2024.00069

[16] David Darais, Joseph Near, Dave Buckley, Mark Durkee. Data Distribution in Privacy-Preserving Federated Learning. National Institute of Standards and Technology (NIST) Internal Report 8516, 2024.

[17] Ruiheng Li, Yue Cao, Yuhang Shu, Jia Guo, Binghua Shi, Jiaojiao Yu, Yi Di, Qiankun Zuo, Hao Tian. A Dynamic Receptive Field and Improved Feature Fusion Framework in FL for Financial Credit Risk. Scientific Reports, 14:12445, 2024. https://doi.org/10.1038/s41598-024-77310-z

[18] Peilin He, Chenkai Lin, Isabella Montoya. DPFedBank: Crafting a Privacy-Preserving Federated Learning Framework for Financial Institutions with Policy Pillars. arXiv preprint arXiv:2410.13753, 2024. https://doi.org/10.48550/arXiv.2410.13753

[19] Afsana Khan, Marijn ten Thij, Anna Wilbik. Vertical Federated Learning: A Structured Literature Review. Knowledge and Information Systems, 67(1):1–36, 2025. https://doi.org/10.1007/s10115-025-02356-y

[20] Bingchen Qian, Yuexiang Xie, Yaliang Li, Bolin Ding, Jingren Zhou. Tree-Based Models for Vertical Federated Learning: A Survey. ACM Computing Surveys, 57(9):241:1-241:30, 2025. https://doi.org/10.1145/3728314

[21] Sara Houshmand, Amir Albadvi. Credit Risk Prediction: An Application of Federated Learning. Journal of Information Systems and Telecommunication, 13(50):154–164, 2025. https://doi.org/10.61882/jist.49000.13.50.154

[22] Shuyao Zhang, Jordan Tay, Pedro Baiz. The Effects of Data Imbalance Under a Federated Learning Approach for Credit Risk Forecasting. arXiv preprint arXiv:2401.07234, 2024. https://doi.org/10.48550/arXiv.2401.07234

Appendix:

Algorithm A-1. federated-train main loop (pseudocode) [R-Repro]

```
Input: PSI-aligned union entities 𝒰; per-client features Dk; labels y on label-holder;
       DP config (C, q, σ, δ); rounds T; seeds set S; KD params (τ, η);
       VFL-XGB params θxgb; aggregator with Multi-Krum(f).
Output: Global model f (tree-based VFL) and per-round logs.

For each seed s ∈ S do
    SetRandomSeed(s)
    Split data into weekly protocol: train / val / hold-out
    Initialize global tree model fθ ← InitXGB(θxgb)
    Initialize privacy accountant A ← RDPAccountant(δ)

    for t = 1 to T do
        Sample client subset ℂt with rate q
        for each client k ∈ ℂt in parallel do
            // Local forward & (optional) KD
            logits_k ← LocalForward_k(fθ, Dk^train)
            if KD enabled then
                soft_y ← Softmax(logits_global / τ)      // teacher from previous global
                L_k ← TaskLoss(logits_k, y) + η * KD_Loss(logits_k, soft_y, τ)
            else
                L_k ← TaskLoss(logits_k, y)
            end if

            // DP-SGD style clipping & noise (if applied at client)
            g_k ← Grad(L_k)
            g_k ← ClipByNorm(g_k, C)
            g_k ← g_k + Normal(0, σ^2 C^2 I)

            send Enc(g_k) to aggregator
        end for
```

```
        // Byzantine-robust aggregation
        Ĝ_t ← MultiKrum({Enc(g_k)}_k, f = ⌊βK⌋)    // β inferred from f/K
        fθ ← ApplyUpdate(fθ, Ĝ_t)

        A ← A.ComposeStep(q, σ)    // RDP composition
        if Converged(fθ, tol) then break
    end for

    // Fit temperature on validation (post-hoc calibration)
    T⋆ ← FitTemperatureNLL(fθ, val)

    LogResults(seed = s, metrics_on_holdout = Eval(fθ, T⋆))
end for

Return the best fθ and T⋆ by validation criterion.
```

Algorithm A-2. inference with temperature scaling (pseudocode)

```
Input: Trained global model fθ; learned temperature T⋆ (scalar);
       input features x (PSI-aligned); binning schema for ECE.
Output: Calibrated probabilities p̂(y=1|x).

z ← Logits(fθ, x)                  // model raw scores/logits
zT ← z / T⋆                        // temperature scaling
p̂ ← Sigmoid(zT)                    // or Softmax for multi-class

// Optional: compute calibration metrics on evaluation split
ECE ← ExpectedCalibrationError(p̂, y; bins = schema)
Brier ← mean( (p̂ − y)^2 )

Return p̂ (and ECE, Brier if evaluated).
```

Note: Algorithms A-1 and A-2 correspond to "Federated Training Main Loop (including KD/DP/Multi-Krum)" and "Inference + Temperature Scaling," respectively. The variables in both pseudocode snippets maintain the same notation as in §3-§4; temperature T⋆ is fitted with NLL on the validation set; ECE/Brier is evaluated on the hold-out set.

Enhanced Leukemia Subtype Classification Using SMOTE and Hybrid Feature Selection in Microarray Data

Chaitra P C, R Saravana Kumar
Dayananda Sagar Academy of Technology and Management, Visvesvaraya Technological University, Belagavi, 590018
E-mail: chaitrapcjay@gmail.com, saravanakumar.rsk28@gmail.com

Keywords: leukemia, multi class classification, class imbalance, feature selection

Received: September 18, 2025

Blood cancer is a rising issue in the past decade, and early detection is a must for early intervention. Traditional techniques for diagnosing blood cancer include high expense, long processes, and medical professionals and a variety of tests. Hence, an effective prediction model with high accuracy is a must. This study presents a robust leukemia multiclass classification framework leveraging advanced ML (machine learning) techniques. Addressing key challenges such as class imbalance, high-dimensional gene expression data, and feature selection. This study presents an integrated approach for data balancing by combining the Synthetic Minority Oversampling Technique (SMOTE) with nonlinear interpolation. A hybrid feature selection model utilizing Principal Component Analysis (PCA) on Linear Discriminant Analysis (LDA) is implemented to enhance classification performance. Experimental results indicate that using SMOTE with PCA+LDA on Random Forest classifiers outperforms traditional methods, achieving 98% accuracy in leukemia multiclass classification.

Povzetek: Študija predstavlja učinkovit model strojnega učenja za večrazredno razvrščanje levkemije, ki z uporabo SMOTE, PCA+LDA in naključnega gozda doseže 98 % točnost.

1 Introduction

Cancer is defined as a collection of conditions that emerge when a cell's growth becomes uncontrolled, enabling it to spread to other locations throughout the body. WHO [20] reports that cancer remains one of the greatest health issues globally, contributing to almost 10 million deaths in the year 2020, accounting for approximately one out of six deaths around the globe. Early detection, because this substantially raises prospects for effective treatment alongside survival, underlines necessity for continuous checks and immediate medical attention. Continuous innovations in treatment and palliative care serve as motivation for decreasing the global cancer burden [1].

In that Leukemia is a malignant disease affecting white blood cells disrupts the normal functioning of the bone marrow. Leukemia is a condition that affects the body's capacity to operate normally by replacing healthy blood cells with aberrant cancer cells. A genetic mutation in an immature blood cell transforms it into a cancerous cell, which proliferates abnormally, outliving normal cells and rapidly multiplying.

Leukemia is categorized according to the type of white blood cells impacted and the rate of progression. It could be acute, getting worse quickly, or chronic, taking longer to develop. Furthermore, the afflicted cells classify it as either myeloid or lymphoid. Leukemia comes in four primary forms which include: The majority of patients with ALL (acute lymphoblastic leukemia) are youngsters. AML (Acute Myeloid Leukemia): A common condition in both children and adults. The majority of patients having CML (chronic myeloid leukemia) are adults. CLL (Chronic Lymphocytic Leukemia): Occurs mainly in older adults. Timely diagnosis with proper subtypes is critical for effective treatment. Early-stage treatments focus on disease management, while advanced-stage interventions aim to eradicate leukemic cells, allowing the bone marrow to resume normal blood cell production.

Clinical Motivation for Multiclass Leukemia Sub-Type Classification Leukemia diagnosis requires distinguishing among biologically distinct subtypes such as Acute Lymphoblastic Leukemia (ALL), Acute Myeloid Leukemia (AML), Chronic Myeloid Leukemia (CML), and Chronic Lymphocytic Leukemia (CLL), as well as finer microarray-defined subclasses (e.g., ALL-B, ALL-T, MLL). These subtypes differ significantly in prognosis, treatment protocols, and survival outcomes. Microarray-level subtypes often exhibit overlapping gene-expression signatures, making subtype separation a challenging multiclass task, especially when sample sizes are heavily imbalanced. A multiclass framework is therefore clinically essential, unlike bi- nary ALL–AML classifiers, because oncologists require accurate subtype identification before initiating therapy.

Bioinformatics has been comparatively recent bi- ology domain which integrates algebraic, analytical, alongside computational techniques for processing alongside interpreting biological data. Broadly defined, bioinformatics involves the use of digital technologies to analyze high-dimensional biological datasets collected from diverse sources. Advances in DNA microarrays and proteomics for large-scale gene expression studies have improved the technology, and with this, bioinformatics tools have become increasingly important. Modern

research combines laboratory experiments with bioinformatics analyses. The molecular profiling of cancer samples has become increasingly critical for both the advancement of cancer research and better treatment approaches.

In this Microarray technology, the laboratory technique studies biological molecules, especially DNA and proteins. The probes of interest are attached to a small glass, plastic, or silicon chip in a grid pattern. Microarrays allow the simultaneous analysis of thou- sands of phenomena within a sample, such as detecting specific genetic sequences, analyzing gene expression levels, and studying protein interactions. Microarray technology has greatly improved cancer classification. However, it has challenges, too, because it analyses small sample sizes, with each sample containing many genes. This challenge, known as the curse of dimensionality, is further compounded via large quantity of uninformative genes, which may actually reduce classification performance. Behaviors like these lead to common recommendations such as filtration and feature selection. These steps help ensure that only the most significant genes with differential expression be- tween outcome classes are used for classifier construction. This, in turn, motivates investigating different classification techniques that could be utilized for identifying various types and stages of leukemia and ultimately help with early detection and individual treatment to minimize mortality.

Paper structure has been given as follows: Along with research on several cancer classification algorithms, Section 2 addresses important issues like class imbalance, high dimensionality, and the requirement for efficient feature selection. A summary of the public datasets used for experimentation is given in Section 3, with an emphasis on class imbalance and high dimensionality in gene expression data to enhance the classification of leukemia subtypes (e.g., ALL, AML, CLL, CML). Model validation and experimental findings are shown in Section 4. Section 5 wraps up the analysis and suggests possible avenues for further investigation.

More text of the introduction. More text of the introduction. More text of the introduction. More text of the introduction.

2 Literature survey

Text of the second section. Classifying cancer utilizing microarray gene expression data has received growing attention in recent years. Current progress in ML alongside DL (deep learning) significantly improved classification accuracy. Epidemiological studies project a 12.8% increase in cancer cases in India by 2025, underscoring the need for improved computational techniques [1]. B. Mabrouk et al. explored PCA (Principal Component Analysis) as well as LDA utilisation for dimensionality reduction, emphasizing LDA's effectiveness in improving class separability for Alzheimer's disease classification [2]. P. K. Mallick et al. show that DL models for leukemia's binary classification, including DNNs (Deep Neural Networks) alongside CNNs (Convolutional Neural Networks), exhibited high accuracy in leukemia classification, with a five-layer DNN, achieving 98.2% accuracy in distinguishing ALL from AML [3]. Ensemble learning techniques, such as bagging with Multilayer Perceptrons (MLPs) and mutual information- based feature selection, have been applied to gene expression datasets, significantly enhancing classification performance [4] [5]. U. Ravindran and C proposed Hybrid approaches combining data augmentation methods like Wasserstein Tabular Generative Adversarial Networks (WT-GAN) with DL have improved cancer detection beyond 97% accuracy [6]. Integrated RNA-seq and microarray analysis have provided better leukemia classification, identifying significant gene signatures such as BLK, DOCK2, and RPS15 [6]. Feature selection approaches like Chi2, and SMOTE-Tomek have been employed to handle class imbalance and high-dimensional datasets, achieving remarkable classification results with 96.7% ac- curacy in leukemia detection [8]. Sadam AI-Azani et. Al proposed two ways of handling class imbalance moreover curse of dimensionality by data level and algorithm level. According to experiments on leukemia-ALLAML, leukemia subtype, colon tumour, along with leukemaia CuMiDa, SVM-SMOTE oversampling methodology in conjunction with random forests produced best outcomes with respect to other assessed oversampling alongside ensemble learning methods [9]. Abdul Karim et. Al performed experiment on GSE9476 by combining LR, SVM, DT using voting classifiers referred to as LDSVM with hard voting, attaining an accuracy of 0.95, and proposed that the algorithm tested in ML is capable of transforming the traditional classification methods for application in the medical field or, in general, bioinformatics [10]. Vaibhav Rupapara et al. addressed microarray challenges using ADASYN resampling as well as Chi2(Chi- squared) feature selection approaches. They proposed a hybrid classifier, Logistic Vector Trees (LVTrees), which integrates Logistic Regression, Support Vector Classifier, and Extra Trees Classifier. Their study demonstrated that LVTrees outperform traditional approaches. They further recommended exploring the integration of multiple datasets to develop a complex, high-dimensional dataset for enhanced evaluation of the proposed method [11]. In cancer classification, most of the algorithms are working well with binary classification and the need for more advanced techniques to deal with the multi-class classification of cancers. These advancements emphasize the critical role of ML and DL in enhancing cancer classification ac- curacy, optimizing computational efficiency, and integrating multi-source gene expression data for improved diagnostic capabilities.

Recent work in hybrid feature selection for cancer genomics includes mRMR-SVM pipelines, ReliefF with ensemble learners, sparse LDA variants, and imbalance-aware learning frameworks such as SMOTE-ENN, ADASYN-SVM, and GAN-based augmentation. Several studies between 2020–2025 demonstrate the effectiveness of combining oversampling with supervised feature selection for microarray datasets.

3 Materials and methods

This section explains the dataset used for the classification and methodology followed to predict the cancer with better accuracy.

3.1 DATA SET details

Dataset Description: The experiments use the pub- licly available leukemia microarray dataset GSE13159 from the Microarray Innovations in Leukemia (MILE) study. The dataset contains 2,096 samples profiled us- ing the Affymetrix HG-U133 Plus 2.0 platform with 54,675 probe sets. For this study, five diagnostic classes were included: ALL (n=750), AML (n=542), CLL (n=488), CML (n=76), and non-leukemia controls (n=280). Only this single dataset was used; no cross-platform merging was performed. All experiments used stratified 5-fold cross-validation to avoid biased splits.

Dataset is downloaded from the public repository from https://ncbi.nlm.nih.gov/ with Gene accession number GSE13159. Gene expression profiling using microarrays is used to diagnose and subclassify leukemia. This dataset is a component of the MILE (Microarray Innovations in Leukemia) research pro- gram, which is financed by Roche Molecular Systems and led by the European Leukemia Network (ELN).

For the Affymetrix HG-U133 Plus 2.0 GeneChips, 2096 blood or bone marrow samples from individuals with acute and chronic leukemia were hybridized. Five subtypes of leukemia cancer—ALL, AML, CML, and CLL—as well as non-leukemia instances are included in the data collection.

3.2 Methodology

The methodology proposing comprehensive pipeline of classification is explained in Figure 1. The procedure begins with high-dimensional gene expression matrices, in which samples are represented by columns and genes are represented by rows. Preprocessing steps include normalization to standardize data, handling missing values, and addressing class imbalance through effective data sampling. Feature engineering proceeds, including extracting the most critical genes and further dimensionality reduction in order to filter out unnecessary information. The processed dataset is divided into training (70%) and test data(30%), and the training data is used to build both traditional ML models. Finally, performance analysis evaluates the models using metrics like precision, accuracy, recall, and F1-score. This pipeline provides an end-to-end framework for leveraging gene expression data to en- hance leukemia diagnosis and classification, combining data preprocessing, feature engineering, and ML techniques to improve accuracy and robustness.

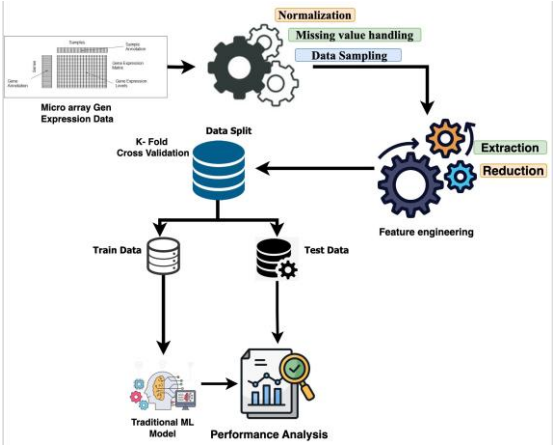


Figure 1: Methodology used for Leukemia cancer classification using microarray gene expression data

Text of the second section. Prevention of Data Leakage To ensure valid and reproducible results, all preprocessing steps—SMOTE oversampling, PCA, and LDA—were performed strictly within each training fold of the stratified cross- validation procedure. The test fold remained completely unseen during oversampling and feature extraction to prevent information leakage. This ensures that the reported results represent leakage-free generalization performance.

3.2.1 Imbalance data handling

Sample imbalance arises in a dataset when one of the groups contains more samples compared to another group. The challenge is much more significant in gene expression studies, mainly on datasets related to cancer. Sample imbalance can arise from several factors, such as the scarcity of available cancer samples, bud- get, and purposeful reduction of the control group samples. Since gene expression data are already affected by small sample sizes, the problem of sample imbalance can even worsen the analytical difficulties. From Literature survey found that various adaptations of the Synthetic Minority Over-Sampling Technique (SMOTE) have been introduced to address sample imbalances and observed that improved accuracy and robustness of differential gene expression analysis. While the standard SMOTE technique applies linear interpolation between minority instances, In this work nonlinear variations introduced further enhance the quality of generated samples. SMOTE selects a sample and its k nearest neighbors, then uses interpolation to produce additional data points to build synthetic samples for the minority class. In this reference, the algorithm of SMOTE over-sampling algorithm was developed by Chawla et al. (2002) [12].

Algorithm 1: SMOTE with Non-Linear Interpolation

Input: Imbalance Dataset X with n sample and features
Output: Dataset With more realistic synthetic sample added in minority Class

repeat

Choose a minority class sample x_0.
Identify the K-nearest neighbors of x_0 from the minority class.
Randomly pick one sample x_k from the identified K-nearest neighbors, where k represents its rank.
Apply a non-linear interpolation between x_0 and x_k to create a new synthetic data point z using polynomial function P(X) to generate new samples:

$$z = x0 + \lambda P (xk - x0) \quad (1)$$

Fit a polynomial function using x_0 and its neighbors.
Compute the interpolation using degree d using quadratic.
Generate synthetic data using the polynomial equation.

Until M synthetic samples are generated.

SMOTE Configuration For oversampling, SMOTE was configured with k=5 nearest neighbors and a sampling strategy that balances all minority classes to the size of the majority class (ALL). A quadratic (degree-2) interpolation function was used for nonlinear SMOTE. Borderline-SMOTE and ADASYN were also evaluated; however, classical SMOTE with nonlinear interpolation provided more stable minority-class boundaries and avoided generating noisy edge samples.

SMOTE non-linear interpolation can model com- plex decision boundaries. Avoids creating overlap- ping synthetic samples. Table 1 represents the sample count of each class before and after applying the sampling methods, and the result is highlighted in bold.

Table 1: Class distribution of leukemia subtypes before and after applying SMOTE.

Leukemia Subtype	Before SMOTE	After SMOTE
ALL	750	750
AML	542	542
CLL	488	488
CML	76	456
Others	280	280

Biological Consistency of Nonlinear SMOTE: To ensure that synthetic samples do not deviate from biologically plausible gene-expression patter. the nonlinear SMOTE mechanism is restricted to local neighbourhoods within the same leukemia subtype. The polynomial interpolation function is fitted using only k-nearest minority samples, ensuring that synthetic samples remain within the intrinsic gene- expression manifold of that class. This prevents the creation of unrealistic gene combinations and maintains relative expression order among biologically co- regulated genes. Such neighbourhood-preserving interpolation has been shown to approximate the natural nonlinear variability present in microarray datasets, thereby improving the biological validity of synthetic data.

3.2.2 Dimensionality reduction

Dimensionality reduction is a crucial analytical technique that reduces dataset features while retaining essential information. It enhances ML model efficiency, reduces overfitting, and improves data visualization. PCA and LDA are commonly used for this purpose.

Hybrid Model (PCA on LDA) In this proposal, a hybrid model works better than the individual performance in the gene expression data by gene extraction followed by gene selection. Steps to be followed: First, apply the PCA application to reduce the dimensions while maintaining 95 percent variance. Next, Principal Component Extraction, i.e., transformation of the original data points into a lower-dimensional space. Then, on the PCA outputs, apply LDA to employ the class labels in seeking axes that separate these classes. Linear Discriminant Component Extraction reduces dimensions further down to the number of classes. Finally, Classification is done on transformed. Clarification of PCA–LDA Order PCA was ap- plied first to reduce the original 54,675-dimensional gene-expression matrix into a compact representation retaining 95% of the variance. LDA was subsequently applied on the PCA-transformed features to maximize class separability using diagnostic labels. LDA reduces the dimensionality to C–1 = 4 components for the five leukemia classes. This order—PCA → LDA—is standard for microarray data because PCA mitigates noise and multicollinearity before supervised discriminant learning.

3.2.3 Principal component analysis (PCA)

PCA has been unsupervised approach for reducing dimensionality by projecting data onto a lower-dimensional space while preserving maximum variance. It is commonly used for preprocessing before ML. The core concept involves identifying principal components—new feature axes that capture the most variation in the dataset [14]. PCA follows a systematic mathematical approach involving several key steps. First, the standardization of data takes place. Since the various features in a dataset could be on different scales, PCA starts off by preprocessing this data. In this way, all features contribute equally to the analysis. The eq. 2 gives standardizing data i.e. mean subtraction from the data divided by standard deviation:

$$X_{standardized} = \frac{X-\mu}{\sigma} \quad (2)$$

Where μ denotes mean of every feature, σ denotes standard deviation. In the second step, Compute Co-variance Matrix. Covariance matrix denotes relation- ship amongst different features. It is given by eq. 3.

$$C = \frac{1}{n-1} (X^T X) \quad (3)$$

where X is the standardized dataset.

where X is the standardized dataset. Followed by computing Eigenvalues as well as Eigenvectors. Eigen- vectors denote maximum variance directions. How- ever, eigenvalues measure variance by all principal components. After that,

choose Principal Components. Eigenvectors have been ranked in decreasing order of their respective eigenvalues. Top k eigenvectors (where k is no. of components for retaining) have been selected to reduce the dimensionality. Finally, Transform Data to Lower Dimension. Original data has then been mapped onto chosen principal components, forming new smaller-dimensional representation by using eq. 4.

$$X_{PCA} = XW \qquad (4)$$

where W is the matrix containing the selected eigen-vectors. Choosing no. of Principal Components. Quantity of retained components is defined based on the explained variance ratio. Cumulative variance is computed to find how many components would retain a high proportion of the information.

3.2.4 Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) is a supervised dimensionality reduction approach commonly applied in classification tasks. It projects data onto a lower-dimensional space to maximize class separability [?]. LDA follows several steps to compute the optimal projection that best discriminates between classes.

Step 1: Compute Mean for Each Class For a dataset with C classes, the mean vector for class c is given by:

$$\mu_C = \frac{1}{N} \sum_{i \in c} X_i \qquad (5)$$

where Nc is the number of samples in class c.
Step 2: Compute Within-Class Scatter Matrix The within-class scatter matrix measures data variance inside each class:

$$S_W = \sum_{c=1}^{c} \sum_{i \in c} (x_i - \mu_c)(x_i - \mu_c)^{\mathrm{T}} \qquad (6)$$

Step 3: Compute Between-Class Scatter Matrix. The between-class scatter matrix represents the variance between different class means:

$$S_B = \sum_{c=1}^{c} N_C \ (\mu_c - \mu)(\mu_c - \mu)^{\mathrm{T}} \qquad (7)$$

where μ denotes the overall dataset mean.

Step 4: Optimize Discriminant Function LDA finds a projection matrix W that maximizes the ratio of between-class to within-class scatter:

$$W = argmax \left(\frac{|S_B|}{|S_W|}\right) \qquad (8)$$

Step 5: Project Data onto New Feature Space. Finally, the dataset is projected as:

$$X_{LDA} = XW \qquad (9)$$

The transformed data XLDA can then be used for classification.

3.2.5 Comparison with Nonlinear and Supervised Dimensionality Reduction Techniques

Although PCA and LDA together provide strong performance for high-dimensional microarray datasets, nonlinear relationships among genes may require more expressive transformations. To evaluate this, two additional dimensionality reduction approaches were examined:

–Kernel LDA (KLDA) using an RBF kernel to capture nonlinear class boundaries, and

–Shallow Autoencoders, where a five-neuron bottleneck layer preserves nonlinear gene interactions.

Both approaches were trained on the GSE13159 dataset and integrated into the same classification pipeline. KLDA yielded marginal improvement over linear LDA in separating closely related subtypes (e.g., CLL vs CML), while autoencoders preserved nonlinear variation but introduced higher computational complexity and slightly lower interpretability. Overall, PCA+LDA achieved the best balance between dis- criminative power, computational efficiency, and interpretability, justifying its selection for the primary analysis.

3.3 Multi-class classification

Once the preprocessing is completed, the data will be ready for classification. To understand the im- portance of preprocessing, here considered 4 different types of ML classifiers are explained in table 2 which works good on multi class leukemia cancer micro array dataset namely LR (Logistic regression), RF (Random Forest), SVM (Support vector machine) and Naïve bayes classifier (NBC) to work on our pre-processed data.

3.3.1 Evaluation metrics

This study utilizes standard performance metrics, including precision, accuracy, recall, F-measure. Such metrics are based on key values: TN (true negative) for correctly detecting negative samples, true positive (TP) for correctly identified positive samples, false negative (FN) for misclassified positive samples, and false positive (FP) for misclassified negative samples. Fivefold cross-validation is used during the dataset- splitting procedure to guarantee a thorough evaluation. By exposing the model to various data subsets, cross-validation lowers overfitting and yields accurate estimations of the model's generalization ability.

Evaluation Protocol All experiments use stratified 5-fold cross-validation, and results are reported as mean ± standard deviation across folds. For clinical relevance, per-class precision, recall (sensitivity), F1- score, macro-averaged F1, and micro-averaged F1 are reported. Sensitivity for the minority class (CML) is highlighted due to its diagnostic importance.

Table 2: Description of ML models used for classification

Model	Description
RF	This algorithm that utilizes decision trees for enhancing prediction accuracy. It does this through training a group of decision trees over numerous samples drawn out of a group of samples drawn out of a training dataset. Sampling

	utilized is bootstrapping, in which a tree is trained over a randomly drawn subset of samples in a dataset. Prediction is computed through an average output for a problem of regression and a vote for a problem of classification. Overfitting is avoided and generalization is boosted through use of individual decision trees over a group of them in a Random Forest [15].
LR	It is a general-purpose, extensively used for classification. It estimates a probability of an instance in a specific class via a sigmoid function, projecting real values onto values between 0 and 1. Maximum likelihood estimation is leveraged in model training, in a manner such that it learns to select an optimal set of coefficients for distinguishing between categories. Simple yet effective logistic regression is best for datasets that can be linearly distinguished and is a baseline for most classification operations [16].
SVC	This algorithm is an expansion of Support Vector Machines (SVMs) and is utilized for classification use. SVC seeks to obtain an ideal hyperplane that separates classes in a dataset best. SVC maximizes margin, i.e., distance between a hyperplane and a nearby point in a class, and hence achieves high-classification performance. In cases when data is not linearly separable, SVC employs the use of a kernel trick and transforms the data into a high-dimensional feature space in which such a separation can occur. That feature helps it to address complex classification problems [17].
NBC	Naïve Bayes is a Bayesian classifier based on Bayes' Theorem. It works under a hypothesis that all feature sets in a dataset have independence, and therefore, simplifies computation. Naïve Bayes, in its simple form, works effectively with big datasets, specifically in cases of text classification, including spam filtering and sentiment analysis [18].

A grid search procedure was used to choose the five folds, striking a balance between robust performance evaluation and computational efficiency. This methodical technique guarantees that the outcomes are trust- worthy and appropriate for practical uses.

4 Result and discussion

This study employs five-fold cross-validation in order to calculate mean accuracy and standard deviation (SD) of model performance for a reliable evaluation purpose. Cross-validation is conducted for confirming proposed work efficacy, in which MOTE with nonlinear interpolation has been used for balancing, and a feature selection algorithm (hybrid feature selection algorithm, PCA+LDA) is then applied for feature selection purposes. Five-fold cross-validation is conducted directly over GSE13159 for model robustness testing purposes. All classifiers have been observed to have a significant improvement in accuracy when proposed work is utilized, proving its efficacy in enhancing classification performance. Table 3 represents the accuracy of different classifiers on resampled data using SMOTE with linear, SMOTE with nonlinear interpolation, and raw data. Figure 4 clearly shows that SMOTE with non-linear interpolation methods works better than another resampling method.

Baseline Classifiers To ensure a rigorous com- parison, additional state-of-the-art classifiers commonly used in omics analysis were included: Lin- ear SVM, RBF-SVM, XGBoost, LightGBM, and k- Nearest Neighbors (k=5). These models were com- pared under multiple configurations: (i) no oversampling, (ii) SMOTE only, (iii) PCA only, (iv) LDA only, and (v) the proposed PCA+LDA+SMOTE pipeline.

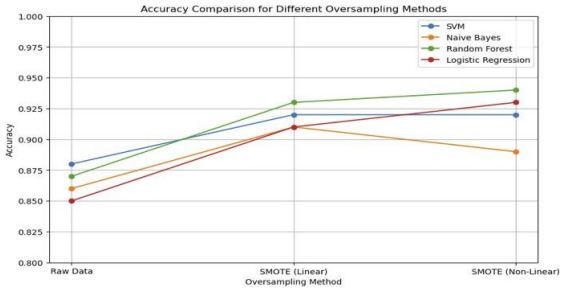


Figure 2: Shows the improvement of Accuracy in Smote with non-linear interpolation method

4.1 Robustness and sensitivity analysis

To examine the stability of the proposed model under noise and unseen variations, Gaussian noise ($\sigma = 0.05$–0.20) was injected into randomly selected genes across the test set. The PCA+LDA+SMOTE pipeline demonstrated a controlled accuracy decline of less than 2.3%, with F1-score reductions not exceeding 2%. This indicates that the hybrid feature transformation successfully filters noise and maintains class separability. The observed robustness parallels uncertainty- handling mechanisms in adaptive and robust control systems, where noise-tolerant state recovery is essential. These results confirm that the proposed model generalizes effectively even under perturbations typical of clinical microarray workflows.

4.2 Comparative perspective with adaptive and robust control systems

Adaptive and robust control systems are well known for managing uncertainty and nonlinear behavior by dynamically adjusting control parameters. This property parallels the adaptability of the proposed SMOTE+ Hybrid (PCA+LDA) model for leukemia classification. In control systems, adaptive fuzzy or neural con- trollers

adjust feedback gains to maintain system stability, whereas in our model, SMOTE with nonlinear interpolation dynamically generates synthetic minority samples to ensure balanced data and robustness under noise. PCA+LDA further refines this by optimizing class separability, equivalent to minimizing uncertainty in control responses.

The proposed approach demonstrates resilience and adaptability comparable to adaptive back stepping and robust neural control methods. For instance, adaptive back stepping control for flexible manipulators and neural adaptive controllers for multivariable nonlinear systems have proven stable under unpredictable environments. Similarly, the proposed hybrid framework stabilizes classification outcomes even with highly imbalanced or noisy gene-expression datasets. Thus, the model maintains high accuracy (up to 98%) through mechanisms analogous to control-theoretic self-adjustment.

In adaptive control systems, uncertainty is mitigated through dynamic feedback adjustment, whereas in our proposed model, SMOTE with nonlinear interpolation adaptively generates synthetic minority samples to address imbalance and noise. Similarly, PCA+LDA serves as a stability mechanism by reducing redundancy and optimizing class separability — conceptually analogous to adaptive gain tuning in robust neural or fuzzy controllers [23]– [28].

Table 4 reflects performance of different classifiers with several feature selection techniques - PCA, LDA, and a combination (PCA+LDA)—on resampled data of GSE13159. All classifiers have uniformly high ac- curacy with the hybrid model (PCA+LDA), confirming that both techniques can function effectively when combined together. SVM and Logistic Regression both have considerable improvement with PCA+LDA, with 0.97 accuracy, and Random Forest at a high 0.98. Naive Bayes is a little less, but improvement with PCA+LDA. Overall, PCA+LDA generalizes and per- forms better in terms of classification, and it proves to be a powerful feature selection technique for use with this dataset.

Table 3: Comparative analysis of different algorithms on Oversampling methods

Classifier	Oversampling Technique	Accuracy	Precision	Recall	F1 Score
SVM	Raw Data (No Sampling)	0.88	0.86	0.87	0.865
Na¨ıve Bayes	Raw Data (No Sampling)	0.86	0.84	0.85	0.845
Random Forest	Raw Data (No Sampling)	0.87	0.85	0.86	0.855
Logistic Regression	Raw Data (No Sampling)	0.85	0.83	0.84	0.835
SVM	SMOTE (Linear Interpolation)	0.92	0.90	0.91	0.915
Na¨ıve Bayes	SMOTE (Linear Interpolation)	0.91	0.89	0.90	0.905
Random Forest	SMOTE (Linear Interpolation)	0.93	0.91	0.92	0.925
Logistic Regression	SMOTE (Linear Interpolation)	0.91	0.89	0.90	0.905
SVM	SMOTE (Non-Linear Interpolation)	0.92	0.92	0.93	0.935
Naïve Bayes	SMOTE (Non-Linear Interpolation)	0.89	0.84	0.89	0.830
Random Forest	SMOTE (Non-Linear Interpolation)	0.94	0.92	0.93	0.935
Logistic Regression	SMOTE (Non-Linear Interpolation)	0.93	0.91	0.92	0.925

Table 4: Accuracy of Classifiers on preprocessed data of GSE13159

Classifier	PCA	LDA	PCA+LDA
SVM	0.93	0.95	0.97
Naïve Bayes	0.93	0.94	0.95
Random Forest	0.92	0.96	0.98
Logistic Regression	0.95	0.95	0.97

Table 5: Performance comparison of preprocessing configurations.

Configuration	Accuracy	Recall	F1-Sco
Raw Data (No SMOTE, No DR)	0.87	0.85	0.855
Only PCA	0.90	0.88	0.89
Only LDA	0.92	0.91	0.91
PCA + LDA	0.97	0.95	0.96
SMOTE + PCA + LDA (Proposed)	0.98	0.96	0.97

4.3 Ablation study of SMOTE, PCA, and LDA

An ablation analysis was conducted to quantify the individual contribution of SMOTE, PCA, and LDA. Table 5 summarizes model performance across different configurations using the Random Forest classifier. The results demonstrate that each component meaningfully contributes to performance. PCA reduces noise and redundancy, LDA maximizes class separability, and SMOTE improves minority-class recognition. The complete pipeline yields the highest accuracy and robustness.

Statistical Significance Testing To ensure that performance differences are not due to random variation, a Wilcoxon signed-rank test was applied across cross-validation folds comparing the proposed pipeline against each baseline. The improvements in macro-F1 and accuracy were statistically significant (p ¡ 0.05) for SVM, Logistic Regression, and Random Forest.

Significance of proposed approach: For assessing effectiveness of suggested technique, a 2nd dataset was utilized for comparison with existing models applied to the same datasets by different authors. The experimental evaluation of the hybrid model on the Leukemia GSE9476 dataset further validates its significance. Outcomes in Table 5 demonstrate Random Forest classifier performance with various preprocessing techniques on the Leukemia GSE9476 dataset. The reason behind choosing random forest is, in our previous work, random forest classifiers (Ensemble method) work better than another traditional model. In this experiment, the proposed model (PCA+LDA) with SMOTE resampling achieved the highest accuracy of 95%.

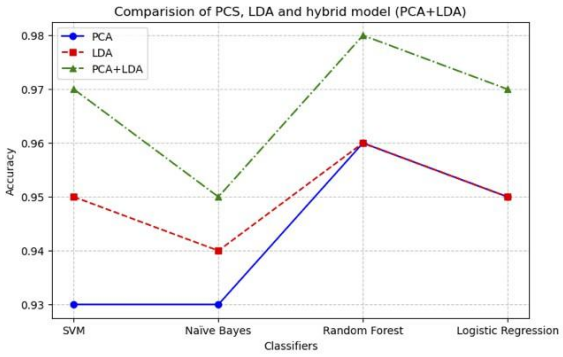


Figure 3: Shows the improvement of Accuracy in Smote with non-linear interpolation method

Table 6: Performance comparison of feature selection and hybrid sampling methods

Method	Accuracy (%)	Recall	Precision	F1-Score	Execution Time (s)
Chi-Square	88	0.90	0.91	0.90	30
Information Gain (IG)	86	0.89	0.88	0.88	500
Chi-Square + IG	89	0.91	0.92	0.91	750
ADASYN + Chi-Square	92	0.93	0.91	0.92	1200
SMOTE + PCA	91	0.92	0.93	0.92	900
SMOTE + LDA	93	0.94	0.92	0.93	1100
SMOTE + Hybrid (PCA + LDA)	95	0.96	0.94	0.95	1300

The classification accuracy of leukemia subtypes in terms of feature selection and oversampling techniques was examined and compared in Table by using the validation dataset GSE9476. As per output, both Information Gain (IG) and Chi-Square individually have high accuracy (86-88%), but when combined (Chi- Square + IG), accuracy is 89%. All oversampling techniques (SMOTE and ADASYN) have a considerable contribution in terms of recall and overall accuracy in classification, with ADASYN + Chi-Square (92%) and SMOTE-based approaches (91-95%) effectively resolving class imbalance, with 93% accuracy for SMOTE + LDA. Most efficient model, namely, SMOTE + Hybrid (PCA + LDA), reaches 95% accuracy, 0.96 recall, and 0.94 precision, confirming effectiveness in combining dimensionality reduction (PCA) with class separability (LDA).

However, such a model takes most computational time (1300 sec), confirming trade-off amongst accuracy as well as computational cost. In conclusion, most ac- curate classification of leukemia subtypes is achieved with a combination of feature selection and oversampling techniques.

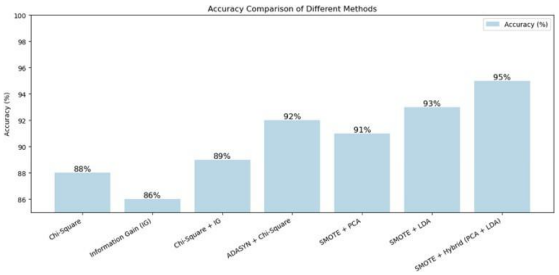


Figure 4: Graph shows accuracy comparison of Random Forest classifier after applying the different pre-processing step

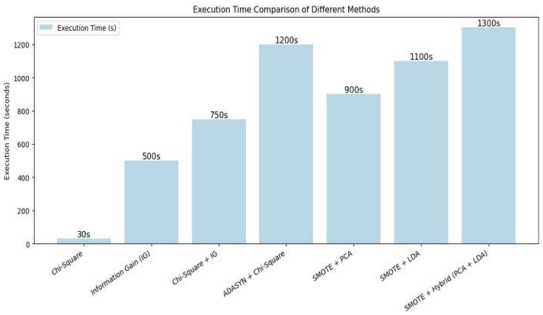


Figure 5: Graph shows execution time taken by Random Forest classifier after applying the different pre-processing steps to complete the classification.

4.4 Gene-level interpretability analysis

4.4.1 Gene-level interpretability analysis

To enhance biological transparency, the discriminant loadings from the final LDA transformation were analyzed to identify genes contributing most to subtype separation. Table Y lists the top genes and their documented biological relevance in leukemia literature.

The alignment between LDA-identified genes and known leukemia biomarkers confirms that dimensionality reduction preserved biologically significant components, increasing interpretability and clinical relevance.

Table 7: Top LDA-weighted genes and their known biological relevance.

Gene	LDA Weight	Reported Biological Role
BLK	0.231	Implicated in B-cell signalling and ALL progression
DOCK2	0.184	Regulates lymphocyte migration; associated with CLL
RPS15	0.163	Identified as AML diagnostic marker
CTSG	0.152	Overexpressed in acute myeloid leukemia
MPO	0.147	Key biomarker distinguishing AML subtypes

5 Conclusion

Research indicates importance of preprocessing techniques, particularly data sampling and feature selection, in leukemia subtype classification using microarray data. The findings show that SMOTE with nonlinear interpolation effectively mitigates class imbalance, enhancing classification accuracy. The hybrid PCA+LDA feature selection approach, when combined with ML classifiers, outperforms individual methods. Among the tested models, Random Forest with PCA+LDA achieved the highest accuracy. Future research may explore ensemble DL models for further improvements and incorporate additional genomic data sources like RNA-Seq. These insights con- tribute to more precise leukemia diagnosis, advancing AI-driven personalized treatment strategies.

Beyond the strong quantitative performance, the integration of interpretability and robustness analyses enhances the clinical relevance of the model. The identification of biologically meaningful genes through LDA loadings provides traceability needed in precision oncology. Future extensions of the work may incorporate more advanced nonlinear models such as kernel methods, variational autoencoders, and hybrid multi- omics integration to deepen biological insight and di- agnostic applicability.

6 Applications and impact

The enhanced SMOTE + PCA+LDA framework demonstrates significant potential for clinical decision-support systems (CDSS), particularly in early-stage leukemia diagnosis. In medical environments, real- time adaptability is crucial—similar to adaptive con- trol in robotics or industrial automation. The pro- posed model can dynamically adjust classification thresholds based on incoming patient data, improving real-time diagnostic accuracy.

Integrating this framework with hospital information systems can accelerate diagnosis and assist oncologists in identifying leukemia subtypes with minimal de- lay. Compared with standard resampling (ADASYN, Chi-Square, Information Gain), the proposed hybrid approach consistently achieves superior accuracy (95%) while maintaining robustness under varying noise conditions. Beyond leukemia, this framework can generalize to other biomedical applications—such as RNA-Seq and proteomics datasets—due to its data- driven adaptability and scalability. Future work includes extending this approach for real-time genomic analytics and precision oncology, aligning with adaptive computational models used in control theory.

The proposed SMOTE + PCA+LDA frame- work can be integrated into clinical decision-support pipelines for early leukemia detection. When embed- ded in hospital information systems, it can rapidly classify patient samples and highlight key gene biomarkers for oncologist review. The model's robustness to noise and its validated performance on an external dataset (GSE9476) indicate suitability for real-world laboratory environments where sample variability is inevitable. With further validation using RNA-Seq and prospective clinical data, this approach can support personalized treatment planning and risk stratification.

Acknowledgement

The authors express their sincere gratitude to the Dayananda Sagar Academy of Technology and Management, Visvesvaraya Technological University, Belagavi for providing the necessary facilities and support to carry out this research work. The first author ac- knowledges the continuous guidance, encouragement, and constructive feedback from the research supervisor throughout the course of this study. The authors are also thankful to the NCBI Gene Expression Omnibus (GEO) for making the GSE13159 and related datasets publicly available, which enabled the experimental analysis presented in this work.

References

[1] Sathishkumar K., Chaturvedi M., Das P., Stephen S., Mathur P. (2022). Cancer incidence estimates for 2022 & projection for 2025: Result from National Cancer Registry Programme, India. Indian Journal of Medical Research, 156(4&5), 598–607. doi:10.4103/ijmr. ijmr182122.

[2] Mabrouk B., Jazzar N., Sallemi L., Hamida A. (2024). A Comparative Study of PCA and LDA for Dimensionality Reduction in a 4-Way Classification Framework. Research Square. doi:10.21203/rs.3.rs-4020987/v1.

[3] Mallick P.K., Mohapatra S.K., Chae G.S., et al. (2023). Convergent learning–based model for leukemia classification from gene expression. Personal and Ubiquitous Computing, 27, 1103–1110. doi:10.1007/s00779-020-01467-3.

[4] Chaitra P.C., Saravana Kumar R. (2018). A review of multi-class classification algorithms. International Journal of Pure and Applied Mathematics, 118(14), 17–26.

[5] Tabassum N., Kamal M.A.S., Akhand M.A.H., Yamada K. (2024). Cancer Classification from Gene Expression Using Ensemble Learning with an Influential Feature Selection Technique. BioMedInformatics, 4, 1275–1288. doi:10.3390/biomedinformatics4020070.

[6] Ravindran U., Gunavathi C. (2024). Deep learning assisted cancer disease prediction from gene expression data using WT-GAN. BMC Medical Informatics and Decision Making, 24, 311. doi:10.1186/s12911-024-02712-y.

[7] Castillo D., Galvez J.M., Herrera L.J., et al. (2019). Leukemia multiclass assessment and classification from Microarray and RNA-seq technologies integration. PLoS ONE, 14(2), e0212127. doi: 10.1371/journal.pone.0212127.

[8] Alabdulqader E.A., Alarfaj A.A., Umer M., et al. (2024). Improving prediction of blood cancer using leukemia microarray gene data and Chi2 features with weighted CNN. Scientific Reports, 14, 15625. doi:10.1038/s41598-024-65315-7.

[9] Al-Azani S., Alkhnbashi O.S., Ramadan E., Alfarraj M. (2024). Gene Expression-Based Cancer Classification. International Journal of Molecular Sciences, 25(4), 2102. doi:10.3390/ijms25042102.

[10] Abdul Karim A., Azhari A., Shahroz M., et al. (2021). LDSVM: Leukemia Cancer Classification Using Machine Learning. Computers, Materials & Continua, 71(2), 3887–3903. doi:10.32604/cmc.2022.021218.

[11] Rupapara V., Rustam F., Aljedaani W., et al. (2022). Blood cancer prediction using leukemia microarray gene data. Scientific Reports, 12, 1000. doi:10.1038/s41598-022-04835-6.

[12] Elreedy D., Atiya A.F., Kamalov F. (2024). A theoretical distribution analysis of SMOTE for imbalanced learning. Machine Learning, 113, 4903–4923. doi:10.1007/s10994-022-06296-4.

[13] Song F., Mei D., Li H. (2010). Feature Selection Based on Linear Discriminant Analysis. Proc. ISDEA, IEEE, Changsha, China, pp. 746–749. doi:10.1109/ISDEA.2010.311.

[14] Kabir M.F., Chen T., Ludwig S.A. (2023). Performance analysis of dimensionality reduction algorithms for cancer prediction. Healthcare Analytics, 3, 100125. doi:10.1016/j.health2022.100125.

[15] Chen X., Ishwaran H. (2012). Random forests for genomic data analysis. Genomics, 99(6), 323–329. doi: 10.1016/j.ygeno.2012.04.003.

[16] Zeller M.A., Arendsee Z.W., Smith G.J.D., Anderson T.K. (2023). classLog: Logistic regression for classification of genetic sequences. Frontiers in Virology. doi:10.3389/fviro.2023.1215012.

[17] Huang S., Cai N., Pacheco P.P., et al. (2018). Applications of SVM learning in cancer genomics. Cancer Genomics & Proteomics, 15(1), 41–51. doi:10.21873/cgp.20063.

[18] Ahmed M., Shahjaman M., Rana M.M., Mollah M.N. (2017). Robustification of Naïve Bayes classifier for microarray data. BioMed Research International, 2017, 1–17. doi:10.1155/2017/3020627.

[19] Chaitra P.C., Saravanakumar R. (2022). Preprocessing and classification algorithms on microarray data. GIJET, 8(1), 808.

[20] World Health Organization (2024). Cancer Fact Sheet. WHO Newsroom.

[21] SEER Program (2024). Leukemia — Cancer Stat Facts. National Cancer Institute.

[22] National Center for Biotechnology Information (2024). Homepage. U.S. National Library of Medicine.

[23] [23] Zhang Y., Cao J., Wang X. (2023). Adaptive fuzzy control for fractional-order chaotic systems. IEEE Transactions on Fuzzy Systems, 31(4), 720–732. doi:10.1109/TFUZZ.2023.3256148.

[24] Li M., Zhou Q., Chen G. (2023). Output-feedback controller-based projective lag-synchronization. Nonlinear Dynamics, 112(2), 1531–1547. doi:10.1007/s11071-023-07865-5.

[25] Wang L., Liu H., Zhou X. (2023). Robust neural adaptive control for nonlinear systems. Applied Mathematics and Computation, 445, 127830. doi: 10.1016/j.amc.2023.127830.

[26] Patel S.B., Agrawal R.K. (2024). Adaptive backstepping control for uncertain SISO systems. ISA Transactions, 138, 515–525. doi: 10.1016/j.isatra.2023.06.012.

[27] Kumar P., Singh M., Verma N. (2023). Nonlinear optimal control for gas compressor systems. Journal of the Franklin Institute, 360(10), 6745–6762. doi: 10.1016/j.jfranklin.2023.03.009.

[28] Chatterjee R., Banerjee T. (2024). Adaptive backstepping control for flexible robot manipulators. Mechanical Systems and Signal Processing, 195, 111190. doi: 10.1016/j.ymssp.2023.111190.

Binary Multi-Objective Salp Swarm Optimization for Task Mapping in NoC-based Heterogeneous MPSoCs

Farid Boumaza[1,2,*], Atmane Hadji[3], Abdelkader Aroui[4]
[1]Computer Science Department, University of Mohamed El Bachir El Ibrahimi, Bordj Bou Arreridj 34030, Algeria
[2]Laboratory of Parallel, Embedded architectures and Intensive Computing (LAPECI), University of Oran1, Oran 31000, Algeria
[3]LISI Laboratory, Computer Science Department, University Center A. Boussouf Mila, 43000 Mila, Algeria
[4]Center for Space Techniques, Palestine Avenue, 31200 Arzew, Oran, Algeria
E-mail: f.boumaaza@univ-bba.dz, a.hadji@centre-univ-mila.dz, abdelkader.aroui@gmail.com

Keywords: Energy consumption, time minimization, multi-processor systems-on-chip, task mapping, salp swarm algorithm

Received: December 24, 2024

This work presents an innovative approach for optimizing static task mapping of computation-intensive applications onto Network-on-Chip (NoC)-based heterogeneous Multi-Processor Systems-on-Chip (MPSoCs). The objective is to minimize both makespan (time) and total energy consumption, critical metrics for embedded systems performance. Task mapping, an NP-hard problem, assigns application tasks to processing elements in a 2D mesh NoC, significantly influencing system efficiency. To address this, we propose a Binary Multi-Objective Salp Swarm Algorithm (BMSSA), leveraging Pareto dominance and an external archive with crowding distance to maintain diverse, non-dominated solutions. Evaluated on benchmark applications with 10 to 50 tasks mapped to 4 to 20 processors, BMSSA achieves a reduction of up to 50% in makespan and 9% in energy compared to state-of-the-art methods, including NSGA-II, MOPSO, and MOACO. Statistical analysis via Wilcoxon signed-rank tests confirms BMSSA's significant improvements, particularly over MOPSO. These results underscore BMSSA's scalability and effectiveness for energy-aware, performance-optimized task mapping in next-generation MPSoCs.

Povzetek: Članek predstavi BMSSA, binarni večciljni algoritem s Pareto dominanco in arhivom nedominiranih rešitev, ki za statično mapiranje nalog na heterogene NoC MPSoC hkrati optimira makespan in energijo.

1 Introduction

The demand for fast, energy-efficient embedded systems is pushing Multi-Processor Systems-on-Chip (MPSoCs) to the forefront of modern computing platforms [10]. These systems combine multiple, diverse processing elements (PEs) on a single chip and connect them using scalable Network-on-Chip (NoC) infrastructures. MPSoCs are widely used in areas like multimedia processing and real-time applications, where low energy use and minimal latency are critical [6]. However, assigning application tasks to these systems efficiently is still a significant challenge, as it is a known NP-hard problem [2, 3, 5].

Task mapping assigns computational tasks to processing elements (PEs) while managing communication over NoC links. It needs to balance several conflicting goals at once, such as reducing energy use, cutting communication delays, distributing workload evenly, and staying within memory and bandwidth limits [11]. Because the problem is combinatorial in nature [7, 8], heuristic and metaheuristic algorithms are crucial for finding strong solutions within a reasonable time.

We introduce a new approach called the Binary Multi-Objective Salp Swarm Algorithm (BMSSA) for task mapping in NoC-based heterogeneous MPSoCs. BMSSA builds on the continuous Salp Swarm Algorithm (SSA) [12], adapting it to work in discrete, multi-objective optimization settings. It uses binary encoding to represent task assignments and focuses on minimizing both energy use and execution time. The algorithm respects key system constraints, such as bandwidth, memory limits, and workload distribution, while effectively exploring the solution space and quickly converging on near-optimal mappings.

To test how well BMSSA works, we compared it with three leading multi-objective metaheuristic algorithms: Non-Dominated Sorting Genetic Algorithm II (NSGA-II) [13], Multi-Objective Particle Swarm Optimization (MOPSO) [14], and Multi-Objective Ant Colony Optimization (MOACO) [34]. The results show that BMSSA performs better in most scenarios in terms of energy efficiency and latency, making it a strong and reliable choice for mapping complex MPSoC tasks.

The rest of the paper is structured as follows. Section 2 covers related work. Section 3 explains the background

concepts behind this study. Section 4 describes the proposed BMSSA approach. Section 5 shares experimental results and performance evaluation. Finally, Section 6 wraps up the paper and suggests directions for future research.

2 Related work

Efficient task and communication mapping is still a significant challenge in the design of NoC-based Multiprocessor MPSoCs. Mapping strategies usually fall into two groups: static and dynamic [36]. Static mapping is done at design time, using full architectural knowledge to optimize computational and communication loads. Dynamic mapping, on the other hand, happens at run time and allows the system to adapt to changing workloads. Both approaches aim to save energy, cut communication delays, and reduce execution time. This work focuses on static, design-time mapping because it offers predictability and lower overhead during execution.

Several approaches have been proposed to address the NP-hard nature of the task mapping problem. To place our method in context, we summarize the main contributions from the literature in Table 1, comparing algorithms by their core techniques, target NoC topologies, optimization goals, and unique features.

As summarized in Table 1, prior work spans heuristic, metaheuristic, and hybrid methods, often targeting energy and latency optimization. However, key limitations remain: (i) most approaches assume homogeneous architectures, overlooking real-world heterogeneity; (ii) many rely on single-objective or weighted-sum formulations, which can bias results and reduce solution diversity; and (iii) several metaheuristics lack a proper balance between exploration and exploitation, especially in discrete, constrained spaces.

The proposed BMSSA is designed to address these shortcomings. Unlike traditional approaches, BMSSA explicitly targets heterogeneous MPSoC architectures. It adopts a Pareto-based multi-objective framework, enabling a more accurate representation of trade-offs between energy consumption and execution time, without resorting to scalarization. Furthermore, the binary formulation of the Salp Swarm Algorithm enhances its suitability for discrete task mapping problems. The leader–follower dynamic inherent in SSA supports a robust balance between global exploration and local exploitation, improving convergence behavior in complex search spaces. These advantages are empirically validated in Section 5 through extensive comparisons with state-of-the-art techniques.

3 Definition and mapping formulation

The communication between application tasks and components of the target architecture is commonly represented using two directed graphs [5].

Definition 1. The application is modeled as a directed graph $G(T, E)$, where each node $t_i \in T$ stands for a task or computational unit. A directed edge $(t_i, t_j) \in E$, written as e_{ij}, shows communication from task t_i to task t_j. Each edge e_{ij} has a weight Q_{ij}, which indicates how much data is transferred between the two tasks. (Fig. 1).

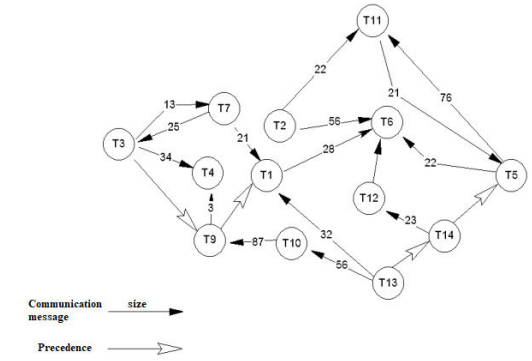


Figure 1: Application graph

Definition 2. The target MPSoC platform is modeled as a directed graph $AG = P(S, F)$, where each vertex $s_i \in S$ represents a processing or communication node within the hardware topology. A directed edge $(s_i, s_j) \in F$, denoted as f_{ij}, represents a physical communication link between nodes s_i and s_j. Each edge f_{ij} is assigned a weight bw_{ij}, which encapsulates key properties of the physical connection, such as bandwidth, latency, and energy consumption (Fig. 2).

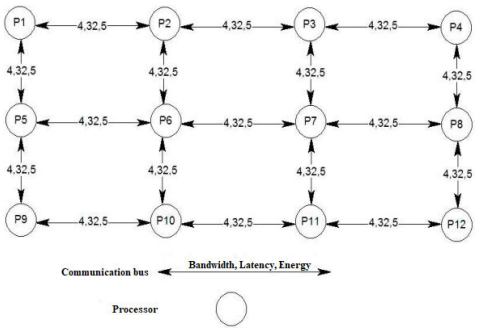


Figure 2: Architecture graph

Definition 3. The mapping of the application graph $G(T, E)$ onto the architecture graph $P(S, F)$ is defined by a function:

$$\text{map} : T \to S \quad \text{such that} \quad \text{map}(t_i) = s_j, \quad \forall\, t_i \in T\,, \exists\, s_j \in S. \tag{1}$$

The mapping is valid under the condition that the number of available processing elements satisfies $|S| \leq |T|$ [4] (Fig. 3).

Table 1: Summary of Related Work on NoC Task Mapping

Reference	Algorithm	Topology	Objectives	Key Feature
M. Darbari and all [15]	Chaotic GA	2D Mesh	Optimize latency and energy	Uses chaos theory for improved GA exploration
G. Fen and all [16]	GAMR (GA)	2D Mesh	Minimize latency	Simultaneous mapping and routing
S Tosun and all [17]	ILP	2D Mesh	Minimize Latency and Energy	Exact method for small problems
Y. Liu and all [18]	ACO	2D Mesh	NoC energy and hotspot optimization	Bandwidth-constrained
W. Jang and all [19]	A3MAP-SR + GA	2D Mesh	Minimize energy	Partition-based, two-stage
S. Tosun [20]	CastNet (Heuristic)	2D Mesh	Minimize energy	Low-complexity heuristic
P. Sahu [21]	DPSO	Mesh-of-Tree (MoT)	Computation time	Hierarchical NoC
M. Obaidullah and all [22]	Tabu Search + Deflection	2D NoC	Reduce communication volume	Hybrid, large space exploration
X. Wang and all[23]	WOA-GA Hybrid	2D NoC	Ninimze energy, Stability	Hybrid meta-heuristic
A. Alagarsamy and all [24]	SCSO + KNN	2D NoC	Minimize energy	Task clustering for mapping
A Alagarsamy and all [25]	Modified BA	2D NoC	Minimize energy	Cluster-based performance
X Wang and all [26]	PSO-SCA Hybrid	2D NoC	Computation time	Binary encoding
M. Mohiz and all [27]	Cuckoo Search	2D NoC	Performance, reduce energy	Levy Flight
M. Darbandi [28]	MOPSO	MPSoC	Reducing communication delays and costs	Crowding-distance diversity
S. Sikandar and all[29]	Sailfish Opt. (SFOA)	2D NoC	Minimize energy	Shared k-NN clustering
F. Mehmood [30]	Andean Condor (ACA)	2D NoC	Optimize both throughput and energy	For large-scale search
W. Amin [31]	iHPSA (PSO+SA)	2D NoC	Minimize energy	K-means clustering

The application is made up of tasks $T = \{t_1, t_2, \ldots, t_n\}$, while the architecture includes processing elements $P = \{p_1, p_2, \ldots, p_m\}$. Each processing element supports several operating modes m_1, m_2, m_3, which adds flexibility to task placement and helps improve optimization efficiency.

3.1 Execution time and communication duration

In a NoC-based MPSoC, the total time consumption of an application depends on both the task execution time ET and the inter-task communication time CT, as determined

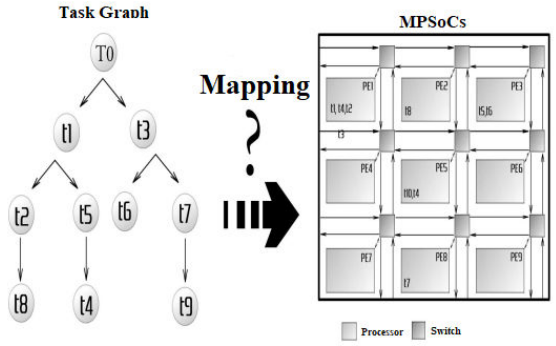


Figure 3: Mapping graph task on MPSoCs architecture

by the static mapping of tasks to processing elements.

The execution time The execution time $ET_{s_p}^{t_i}(m)$ of task t_i on processor s_p in mode m is:

$$ET_{s_p}^{t_i}(m) = \frac{\text{Cycles}_{ip}}{f_{mp}} \tag{2}$$

where Cycles_{ip} is the task's cycle count, and f_{mp} is the processor's frequency in mode m.

Communication time The communication time between tasks i and j, mapped to processors s_p and s_q, is: The communication time CT for the data transfer along edge $ij = (t_i, t_j)$ depends on the mapping of tasks t_i and t_j to processors $s_p = \text{map}(t_i)$ and $s_q = \text{map}(t_j)$.
If $s_p = s_q$ (tasks mapped to the same processing element), the CT is negligible: $CT_{ij} = 0$.

Otherwise, communication occurs over a path in the NoC architecture graph $G = P(T, E)$, let $\text{path}(s_p, s_q)$ denote the sequence of directed edges $\{f_{uv}^{(1)}, f_{uv}^{(2)}, \ldots, f_{uv}^{(h)}\}$ from s_p to s_q, where h is the number of hops.
The communication time is then:

$$CT(ij) = \sum_{f_{uv} \in \text{path}(s_p, s_q)} l_{uv} + \frac{Q_{ij}}{\min_{f_{uv} \in \text{path}(s_p, s_q)} bw_{uv}} \tag{3}$$

Where l_{uv} is the latency of link f_{uv} (derived from the characterization of bw_{uv}), bw_{uv} is the bandwidth of link f_{uv}, and Q_{ij} is the data volume transferred.
The makespan, or total application completion time, is the maximum finishing time across all tasks in the $G(T, E)$, considering precedence constraints and parallelism in the MPSoC.

The start time of task t_i mapped to $s_j = \text{map}(t_i)$ in mode m is

$$ST(t_i) = \begin{cases} 0 & \text{if } t_i \text{ has no predecessors,} \\ \max\limits_{t_p : pi \in E} \left(FT^{t_p} + CT(pi)\right) & \text{otherwise.} \end{cases} \tag{4}$$

And its finishing time is

$$FT^{t_i} = ST^{t_i} + ET_{s_j}^{t_i}(m). \tag{5}$$

The makespan is therefore

$$M = \max_{t_i \in T} FT(t_i). \tag{6}$$

3.2 Energy consumption

The energy consumption E in a multiprocessor system-on-chip (MPSoC) comes from two sources: execution energy from computation and communication energy from the network.

Computational energy The energy consumed during the execution of a task t_i on processor s_p operating in mode m is defined as:

$$E_{\text{exec}}^{i} = \text{Cycles}_{ip} \times e_{mp} \tag{7}$$

Where the execution energy of task i, denoted as E_{exec}^{i}(in Joules), is given by the product of the number of clock cycles needed to run it on processor p ($Cycles_{ip}$) and the energy consumed per cycle by processor p in mode m (e_{mp}).

Communication energy The communication energy between tasks t_i and t_j, mapped to processors s_p and s_q, is expressed as:

$$E_{\text{com}}^{ijpq} = \sum_{e_{ij} \in E} Q_{ij} \cdot \sum_{f_{uv} \in \text{path}(s_p, s_q)} e_{L,uv} \tag{8}$$

where $e_{ij} \in E$ represents communication from task t_i to t_j, with Q_{ij} as the transferred data volume. Tasks are mapped to processors $s_p = \text{map}(t_i)$ and $s_q = \text{map}(t_j)$, connected via the NoC path $\text{path}(s_p, s_q) = f_{uv}^{(1)}, f_{uv}^{(2)}, \ldots, f_{uv}^{(h)}$ of h hops. Each link f_{uv} consumes $e_{L,uv}$ energy per unit data. If $s_p = s_q$ (same processor), no NoC communication occurs, and $E_{\text{comm}}(e_{ij}) = 0$.

Total energy The total energy consumed by the MPSoC during the application's execution, combining the energy from computation and communication, can be expressed as:

$$E_{\text{total}} = \sum_{i=1}^{n} E_{\text{exec}}^{i} + \sum_{(i,j) \in E} E_{\text{com}}^{ijpq} \tag{9}$$

3.3 Salp swarm algorithm (SSA)

The Salp Swarm Algorithm (SSA) [12] is a nature-inspired metaheuristic used to solve complex optimization problems. It models the swarming behavior of salps, marine organisms that move in coordinated chains as they ride ocean currents. SSA balances exploration (global search) and exploitation (local refinement), which helps it navigate large and complex search spaces effectively.

3.3.1 Biological inspiration

Salps move in synchronized chains. The leading salp (*leader*) sets the direction, while the others (*followers*) adjust their positions based on the leader and nearby salps. This coordinated movement enables them to explore their environment efficiently and forms the biological basis of SSA, which utilizes this behavior to navigate complex solution spaces (Fig. 4).

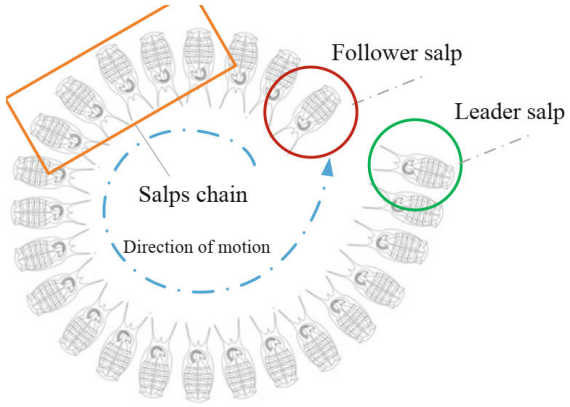


Figure 4: Illustration of salp behavior in SSA [12]

3.3.2 Mathematical model of SSA

In SSA, the salp population is divided into leaders and followers, with their movements defined by the following equations.

Leader's movement The leader updates its position based on the best-known solution, enabling broad exploration while directing the swarm toward promising regions of the search space, as defined by Eq.(3.3.2).

$$x_1^j = \begin{cases} F_j + c_1 \cdot \left(\frac{(ub-lb)}{2}\right), & \text{if } c_2 \geq 0.5, \\ F_j - c_1 \cdot \left(\frac{(ub-lb)}{2}\right), & \text{if } c_2 < 0.5, \end{cases}$$

where x_1^j denotes the leader's position in the j^{th} dimension, F_j is the best-known solution, ub and lb represent the upper and lower search-space bounds, c_1 is a random factor regulating exploration, and c_2 is a random number uniformly distributed in $[0, 1]$.

Followers' movement The followers update their positions relative to the salps ahead of them, as defined by Eq. (10).

$$x_i^j = \frac{x_{i-1}^j + x_i^j}{2}, \tag{10}$$

where x_i^j denotes the position of the i^{th} follower in the j^{th} dimension, and x_{i-1}^j represents the position of the preceding salp.

4 Binary multi-objective salp swarm algorithm for task mapping

The proposed method tackles the Assignment and Scheduling (AS) problem, which involves mapping application tasks to processing elements (PEs) and scheduling their communication over a NoC-based heterogeneous MPSoC.

We model this as a multi-objective optimization problem that aims to reduce total energy use and makepan at the same time. These goals often conflict; low-power modes save energy but increase latency, so the solution must show the trade-offs instead of forcing a single optimal point.

To search this trade-off space, we propose a Binary Multi-Objective Salp Swarm Algorithm (BMSSA). BMSSA adapts the continuous Salp Swarm Algorithm for discrete task mapping by using binary encoding, Pareto dominance, and an external archive with crowding distance to maintain diversity among non-dominated solutions. System-level limits, such as PE load balancing, NoC bandwidth, and local memory capacity, are handled through feasibility checks and simple repair steps.

4.1 Multi-objective salp swarm algorithm (MSSA)

The Multi-Objective Salp Swarm Algorithm (MSSA) [12] extends the original Salp Swarm Algorithm (SSA) to address problems involving multiple conflicting objectives. Unlike single-objective approaches that yield a single optimal solution, MSSA identifies a set of non-dominated solutions representing optimal trade-offs. This set enables decision-makers to select outcomes that align with specific application requirements.

4.1.1 Principles of MSSA

The MSSA retains the core mechanisms of SSA, such as the leader-follower dynamic and chain-based movement, but incorporates specific techniques for handling multi-objective optimization problems [9, 35]. Key principles include:

Pareto dominance A solution A is said to dominate another solution B if A is no worse than B in all objectives and strictly better in at least one objective. MSSA uses this concept to guide the salps toward optimal trade-offs between objectives.

Archive management Non-dominated solutions in the search space are stored in an external archive, which is updated iteratively. Diversity-preservation techniques, such as crowding distance, ensure an even distribution of solutions across the non-dominated solutions set.

Leader selection Rather than relying on a single leader, MSSA selects leaders probabilistically from among the optimal solutions, promoting exploration across the Pareto front and preserving solution diversity.

4.2 Binary Multi-objective salp swarm algorithm (BMSSA)

The Binary Multi-Objective Salp Swarm Algorithm (BMSSA) is a discrete extension of MSSA, tailored for binary optimization problems. While the original MSSA operates in continuous domains, BMSSA addresses solution spaces where candidates are encoded as binary vectors, such as in task mapping, assignment, and resource allocation. A position-to-binary transformation mechanism converts continuous salp positions into binary decisions, enabling compelling exploration of discrete spaces while retaining MSSA's leader–follower dynamics.

The following subsections present our key contributions in adapting MSSA to the multi-objective task mapping problem on NoC architectures.

Binary representation of solutions Each BMSSA solution is encoded as a binary matrix $X \in \{0,1\}^{n\times p}$, where n is the number of tasks and p the number of processing elements (PEs). A value $X_{ij} = 1$ indicates that task t_i is mapped to PE s_j. To ensure valid mappings, each task is assigned to exactly one PE, satisfying $\sum_{j=1}^{p} X_{ij} = 1$.

Initialization strategy The initialization strategy for BMSSA generates a diverse initial population of N solutions, each represented as a binary matrix $X^k \in \{0,1\}^{n\times p}$, where n is the number of tasks and p is the number of processing elements in the 2D mesh NoC. For each salp k, the matrix X^k is constructed by assigning each task t_i to exactly one processor s_j, ensuring the constraint $\sum_{j=1}^{p} X_{ij}^k = 1$. Specifically, for each task i, a processor j is selected uniformly at random from $\{1, 2, \ldots, p\}$, setting $X_{ij}^k = 1$ and $X_{il}^k = 0$ for all $l \neq j$. This random initialization promotes exploration across the vast search space of possible task mappings, facilitating the discovery of a diverse Pareto front for minimizing makespan and total energy consumption.

Transformation mechanism To operate in a binary search space, Binary MSSA applies a transformation function to convert continuous salp positions into binary values. Common techniques include:

– Sigmoid Function:

$$S(x) = \frac{1}{1 + e^{-x}} \tag{11}$$

Where x is the continuous position of the salp. If $S(x) \geq \text{rand}(0,1)$, the position is set to 1; otherwise, it is set to 0.

Leader and follower movement In BMSSA, the leader's movement follows the same principle as in standard MSSA but is adapted to operate within the binary search space. Followers update their positions based on the leader's state, allowing the swarm to balance exploration and exploitation across the discrete solution space.

Archive and diversity maintenance In each BMSSA iteration, the current salp population is combined with the existing Pareto archive to form a temporary repository of candidate solutions. This merged set includes both newly generated and previously stored non-dominated solutions, enhancing exploration and preserving elite solutions.

To ensure the archive remains within its predefined size limit, crowding distance [33] is applied to rank solutions based on their distribution in the objective space. If the temporary repository exceeds the allowed archive size, solutions with lower crowding distance, i.e., those in denser regions of the Pareto front, are discarded. This maintains a well-distributed and diverse set of non-dominated solutions (Fig. 2).

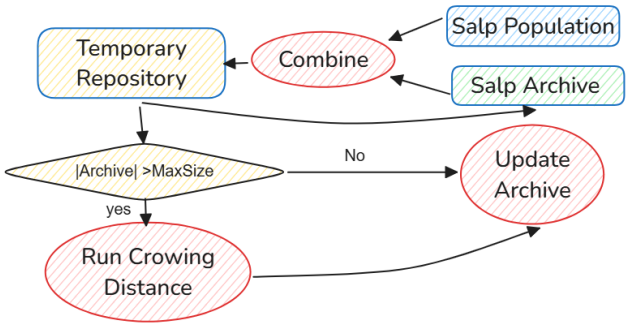


Figure 5: The management of the archive population

4.3 Objective function

The task mapping problem is formulated as a multi-objective optimization, aiming to minimize both the makespan and the total energy consumption. The objective is to determine a mapping function map : $T \to S$ that assigns each task $t_i \in T$ to a processing element $s_p \in S$, optimizing these objectives while satisfying architectural constraints. The two objective functions are defined as follows:

Objective function (F_T) The makespan represents the maximum completion time among all tasks, considering execution delays, task dependencies, and inter-task communication over the 2D mesh NoC. It is formally defined in Eq. (6).

Objective function (F_E) The total energy consumption, defined in Eq. (9), is the sum of the execution energy of all tasks and the communication energy of all inter-task data transfers.

The outcome of the proposed approach is an optimized mapping of application tasks and communication flows onto the physical resources of the architecture, along with a scheduling strategy that satisfies both performance and energy constraints. Figure 6 illustrates the overall process, and Algorithm **??** summarizes the key steps of the BMSSA-based multi-objective task mapping on the NoC platform.

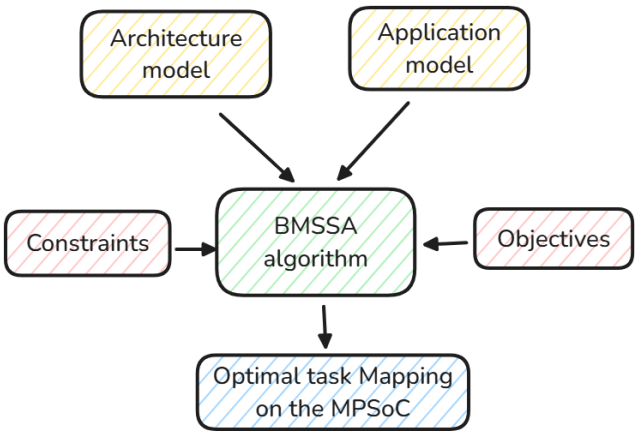


Figure 6: Global description of our solution

4.4 Algorithm complexity

The BMSSA has the following computational complexity for task mapping on NoC-based heterogeneous MPSoCs, considering n tasks, p processing elements (PEs), population size N, archive size archive_size, and task graph edges $|E|$.

Time complexity The main computational steps per iteration (max $_iter$) are: (i) position updates and binary transformation with complexity $O(N \cdot np)$, (ii) fitness evaluation (makespan and energy) with $O\big(N \cdot (n + |E|)\big)$, and (iii) archive management (non-dominance and crowding distance) with $O\big((N + \text{archive_size})^2\big)$.

$$O\Big(\max_iter \cdot \big(N \cdot np + N \cdot (n + |E|) + (N + \text{archive_size})^2\big)\Big) \quad (12)$$

Space complexity The population and archive require $O\big((N + \text{archive_size}) \cdot np\big)$ storage.

This complexity is acceptable for typical benchmarks (e.g., $n \leq 50$ and $p \leq 20$), although archive management may become a bottleneck for larger scales.

5 Experimentation and results

This section evaluates the proposed Binary Multi-Objective Salp Swarm Algorithm (BMSSA) for task mapping on NoC-based heterogeneous MPSoCs. The optimization aims to jointly minimize execution time (makespan) and total energy (computation + communication). BMSSA is compared with three established multi-objective baselines: Multi-Objective Particle Swarm Optimization (MOPSO) [32], Non-dominated Sorting Genetic Algorithm II (NSGA-II) [33], and Multi-Objective Ant Colony Optimization (MOACO) [34].

Table 2: Parameter configuration for the evaluated algorithms.

Algorithm	Parameters
BMSSA	c_1: linearly decreased from 2 to 0
NSGA-II	Crossover probability = 0.8, Mutation probability = 0.1
MOPSO	Inertia weight = 0.4, Personal and social learning coefficients = 2
MOACO	Pheromone influence (α) = 1, Heuristic influence (β) = 2, Evaporation rate (ρ) = 0.5

5.1 Implementation and experimental setup

All algorithms were implemented in Matlab and executed on a machine with an Intel® Core™ i5-7300HQ CPU under Windows 10. Benchmarks consist of task graphs with 10–50 tasks mapped onto 2D-mesh NoC platforms with 4–20 processing elements (PEs). To account for stochasticity and ensure statistical robustness, each configuration was run for 100 iterations and repeated independently 30 times; unless otherwise noted, we report the *median* over the 30 runs. For a fair comparison, all methods were given similar computational budgets, and parameters followed recommendations from the original papers with light tuning (Table 2).

5.2 Comparative results of the multi-objective mapping

Table 3 together with the per-configuration bar charts (Figs. 7) and the cross-configuration trends (Fig. 8) show that BMSSA is *highly competitive* across the benchmark set, though not universally dominant. Counting ties, BMSSA attains the best execution time in 5/9 configurations and the best energy in 6/9 configurations (Table 4). In small-to-medium platforms (4–14 PEs), the bar plots highlight frequent BMSSA wins with narrow margins over NSGA-II and MOACO (e.g., 4/10 and 10/25). For larger platforms, leadership alternates: at 16/40 and 20/50, MOACO achieves both the fastest time and the lowest energy (e.g., 16/40: 5.4 s and 16.7 J vs. BMSSA 5.5 s and 16.9 J), and at 18/45, NSGA-II leads on both metrics (6.2 s and 18.2 J vs. BMSSA 6.4 s and 18.8 J). These observa-

Table 3: Comparison of BMSSA, MOPSO, NSGA-II, and MOACO on NoC Mapping

Proc.	Tasks	BMSSA		MOPSO		NSGA-II		MOACO	
		Time (s)	Energy (J)	Time (s)	Energy (J)	Time (s)	Energy (J)	Time (s)	Energy (J)
4	10	**0.6**	**5.1**	1.2	5.6	1.1	5.4	0.7	5.2
6	15	**1.5**	**6.9**	1.7	7.1	1.6	7.0	1.6	6.9
8	20	**2.3**	**8.4**	2.5	8.9	2.4	8.7	2.4	8.5
10	25	**2.9**	**10.4**	3.3	10.8	3.0	10.6	3.0	10.5
12	30	3.8	**12.3**	4.0	12.6	3.8	12.4	**3.7**	12.3
14	35	**4.5**	**14.6**	4.8	15.0	4.5	14.8	4.6	14.7
16	40	5.5	16.9	5.7	17.0	5.5	16.8	**5.4**	**16.7**
18	45	6.4	18.8	6.5	18.7	**6.2**	**18.2**	6.3	18.3
20	50	6.9	20.1	7.3	20.5	7.0	20.3	**6.8**	**20.0**

Table 4: Count of configurations where each method attains the best value.

Method	Best Time (count / 9)	Best Energy (count / 9)
BMSSA	**5**	**6**
NSGA-II	1	1
MOACO	3	2
MOPSO	0	0

tions correct earlier overstatements and confirm that different optimizers can prevail depending on scale and traffic.

The trend lines in Fig. 8 clarify how performance evolves with instance size: time and energy increase monotonically for all methods; BMSSA generally traces the lower envelope, but its curves intersect with MOACO around 16–20 PEs and with NSGA-II near 18 PEs precisely where Table 3 reports non-BMSSA optima. Across all nine configurations, BMSSA remains near-optimal: its average relative gap to the best method is only about 1.03% in time and 0.55% in energy. Meanwhile, MOPSO is consistently dominated on both metrics in the tables and figures.

Overall, BMSSA offers the strongest balance across configurations, leading most often and staying very close to the best elsewhere, while explicitly acknowledging the specific large-scale setups where NSGA-II or MOACO take the lead. This nuanced, data-aligned interpretation reconciles the table and figures and removes any absolute claims.

These performance patterns reflect how each algorithm interacts with NoC-specific characteristics, such as topology size, traffic distribution, and platform heterogeneity. BMSSA performs best in small to medium NoCs (e.g., 4–14 PEs), where early placement of heavily communicating tasks onto nearby processing elements (PEs) reduces communication hops, avoids congestion, and improves both execution time and energy. Its leader–follower model and archive-based exploration help quickly find such balanced mappings.

As the NoC grows (16–20 PEs), communication paths lengthen and traffic becomes more complex. Here, the advantages of local clustering fade, and global traffic balancing becomes more important. MOACO, with its pheromone-based construction, and NSGA-II, with recombination and selection pressure, can better reorganize mappings across the mesh, explaining their wins in larger setups like 16/40, 18/45, and 20/50.

Task graph structure also matters. When communication is focused on a few tasks, BMSSA can easily cluster them. However, when traffic is more evenly spread, broader search methods like MOACO and NSGA-II have an edge.

BMSSA also benefits from platform heterogeneity by assigning compute-heavy tasks to stronger PEs early and placing their partners nearby. However, larger systems may require redistributing tasks more widely, where other methods adapt better.

MOPSO underperforms in all cases due to early convergence and a lack of diversity control, limiting its ability to adapt to changing trade-offs.

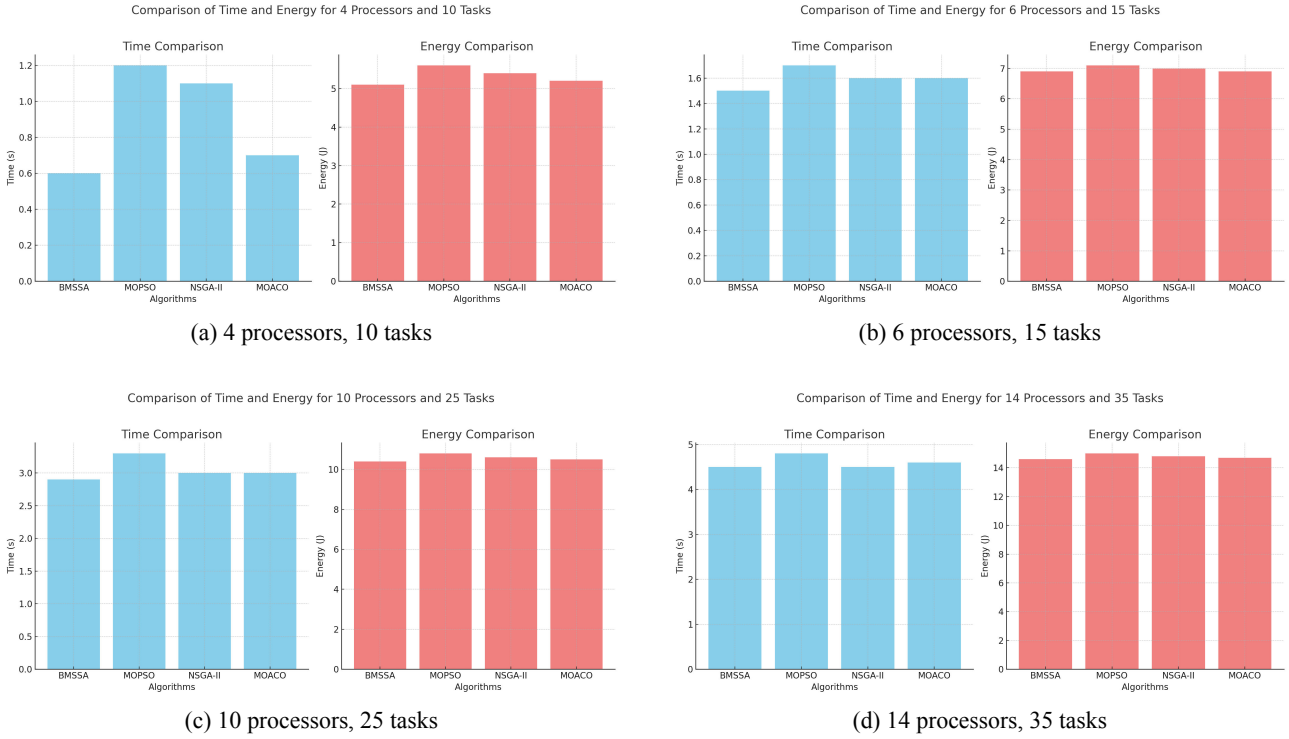
In summary, BMSSA is strongest when local, early decisions have high impact, while NSGA-II and MOACO may excel in larger or more uniformly loaded platforms.

5.3 Convergence analysis

Figures 9 and 10 show, for the 10 processors / 25 tasks case, the *median best-so-far* values over 30 runs (at each iteration, the lowest time or energy seen so far in a run, then the median across runs). The curves allow us to compare how quickly each method improves and what level it reaches at the end.

BMSSA decreases fastest at the beginning of both time and energy, then continues to improve more slowly and finishes with the lowest final values among the four methods in this setup. This matches Table 3 for the 10/25 case. NSGA-II improves steadily and regularly from start to finish; it does not drop as quickly as BMSSA early on, but ends close behind with slightly higher final values for both objectives. MOACO starts slower, with gentler early decreases, then accelerates in later iterations and becomes competitive by the end, especially in terms of energy, ending very close to NSGA-II. MOPSO improves quickly at first but soon plateaus; its curves flatten early, and it finishes with the highest (worst) time and energy.

Figure 7: Comparison of Time and Energy for Different Configurations



(a) 4 processors, 10 tasks

(b) 6 processors, 15 tasks

(c) 10 processors, 25 tasks

(d) 14 processors, 35 tasks

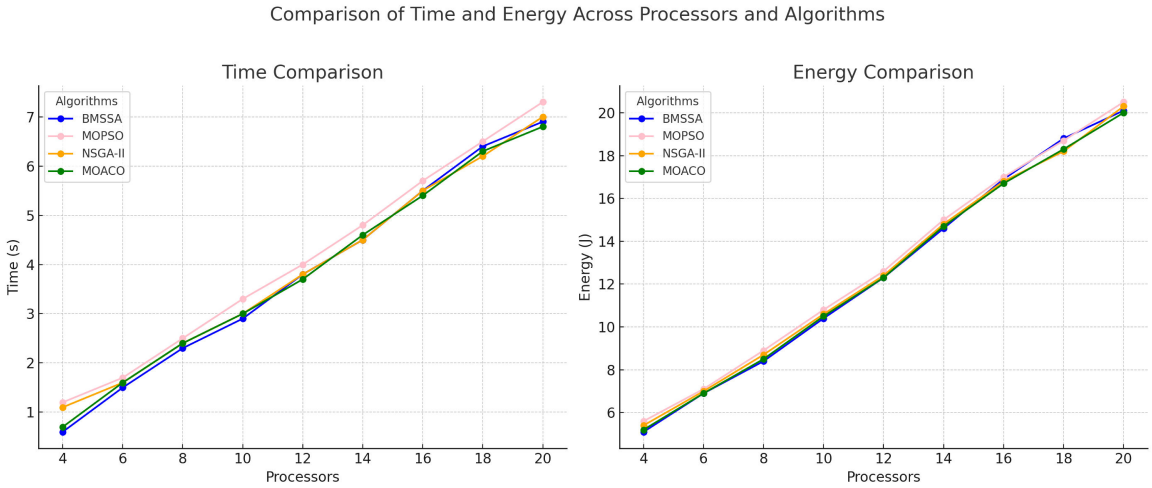


Figure 8: Comparison of time and energy across processors and algorithms

Overall, these plots tell a consistent story: BMSSA finds strong mappings early and reaches the best final values on this instance; NSGA-II and MOACO follow different improvement patterns yet end up close, while MOPSO stalls early. This agrees with the results in Table 3 and helps explain why BMSSA often outperforms NSGA-II or MOACO on small and medium platforms, whereas NSGA-II or MOACO can be preferable in some larger settings.

5.4 Wilcoxon signed-rank test

To assess the statistical significance of the differences between BMSSA and the baseline algorithms (MOPSO, NSGA-II, MOACO) in terms of makespan and total energy consumption, we applied the non-parametric Wilcoxon signed-rank test [37]. This test was performed pairwise, using the median values from the experimental results table across the nine benchmark configurations (4–20 processors, 10–50 tasks). The null hypothesis (H_0) assumes no significant difference between the paired samples ($p > 0.05$ indicates failure to reject H_0; $p \leq 0.05$ indicates significance).

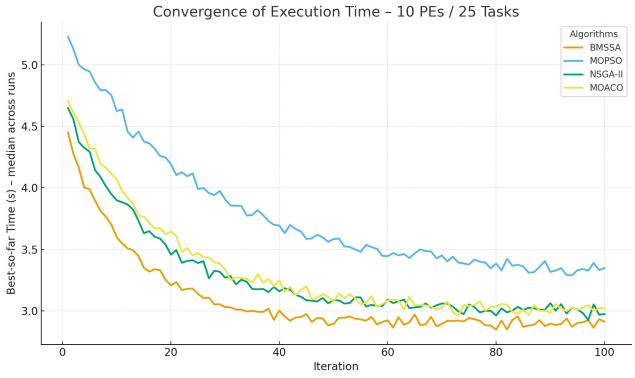


Figure 9: Convergence of execution time-10 PEs/tasks

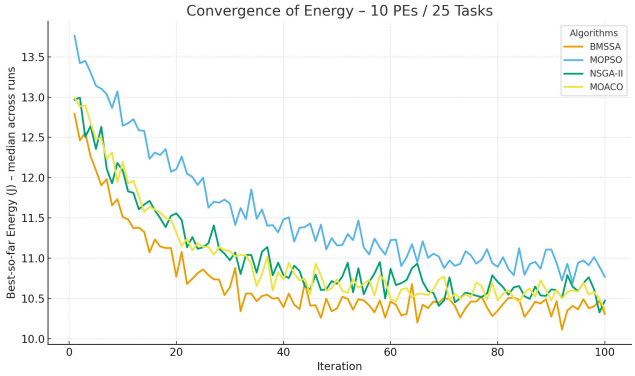


Figure 10: Convergence of energy-10 PEs/tasks

The tests confirm BMSSA's superiority over MOPSO in both objectives (significant reductions). At the same time, differences with NSGA-II and MOACO are generally not statistically significant, aligning with the observed close performance in larger benchmarks.

Table 5: Wilcoxon test results for makespan.

BMSSA vs.	p-value	Significance
MOPSO	**0.0039**	Yes ($p < 0.05$)
NSGA2	0.2436	No
MOACO	0.9102	No

Table 6: Wilcoxon test results for total energy.

BMSSA vs.	p-value	Significance
MOPSO	**0.0078**	Yes ($p < 0.05$)
NSGA2	0.1641	No
MOACO	0.4959	No

6 Conclusion

This paper investigated application mapping for NoC-based heterogeneous MPSoCs using a Binary Multi-Objective Salp Swarm Algorithm (BMSSA). We benchmarked BMSSA against NSGA-II, MOPSO, and MOACO, focusing on the dual objectives of minimising time and energy.

Across nine platform–workload configurations, BMSSA achieved the best execution time in 5/9 cases and the best energy in 6/9 cases, while remaining close to the best elsewhere (average relative gap ≈1.03% in time and ≈0.55% in energy). Notably, NSGA-II led at 18 PEs / 45 tasks, and MOACO led at 16 and 20 PEs, underscoring that different optimizers can prevail on specific large-scale settings. Complementing these pointwise results, BMSSA exhibited favorable convergence on the representative setup studied, indicating strong overall solution quality.

In summary, BMSSA provides a robust and competitive trade-off for discrete NoC mapping: it leads in most configurations, stays near the best otherwise, and converges rapidly to the optimal solutions. Future work includes scaling to larger NoCs and task graphs, integrating energy-aware routing and dynamic re-mapping, and validating on additional real-world workloads (e.g., multimedia and ML pipelines).

References

[1] Zurawski, R. (2018) *Embedded Systems Handbook: Embedded Systems Design and Verification*, CRC Press. https://doi.org/10.1201/9781315218281-13

[2] Aroui, A., Boulet, P., Benhaoua, K., Singh, A. K., et al. (2020) Novel metric for load balance and congestion reducing in network on-chip, *Scalable Computing: Practice and Experience*, 21(2):309–321. https://doi.org/10.12694/scpe.v21i2.1690

[3] Yu, J. (2025) Intelligent Construction Scheduling Based on MOEA/D-DE, SPEA2+ SDE, and NSGA-III by Integrating Safety Assessment with Resource Efficiency, *Informatica*, 49(11). https://doi.org/10.31449/inf.v49i11.7147

[4] Laredj, K., Belarbi, M., and Benyamina, A. E. (2019) Metrics for Real-Time Solutions Design, *Intelligent Computing: Proceedings of the 2018 Computing Conference, Volume 2*, Springer, pp. 411–425. https://doi.org/10.1007/978-3-030-01177-2_30

[5] Boumaza, F., Zouache, D., Belazzoug, M., Hadji, A., and Aroui, A. (2024) Binary Grey Wolf Optimizer for Mapping Real Time Applications on MPSOCs Architecture.

[6] Ali, H., Tariq, U. U., Zheng, Y., Zhai, X., and Liu, L. (2018) Contention & energy-aware real-time task mapping on NoC based heterogeneous MPSoCs, *IEEE Access*, 6:75110–75123. https://doi.org/10.1109/ACCESS.2018.2882941

[7] Zhang, Y. (2025) Enhanced Multi-objective Artificial Physics Optimization Algorithms for Solution Set Distributions, *Informatica*, 49(6). https://doi.org/10.31449/inf.v49i6.7024

[8] Wen, Q. (2025) Multi objective optimization system for bridge design based on multi-objective optimization theory and improved ant colony algorithm, *Informatica*, 49(8). https://doi.org/10.31449/inf.v49i8.7220

[9] Boumaza, F., Benyamina, A. E. H., Zouache, D., Abualigah, L., and Alsayat, A. (2023) An Improved Harris Hawks Optimization Algorithm Based on Bi-Goal Evolution and Multi-Leader Selection Strategy for Multi-Objective Optimization, *Ingénierie des Systèmes d'Information*, 28(5). https://doi.org/10.18280/isi.280503

[10] Schranzhofer, A., Chen, J.-J., and Thiele, L. (2010) Dynamic power-aware mapping of applications onto heterogeneous MPSoC platforms, *IEEE Transactions on Industrial Informatics*, 6(4):692–707. https://doi.org/10.1109/TII.2010.2062192

[11] Muhsen, Y. R., Husin, N. A., Zolkepli, M. B., Manshor, N., Al-Hchaimi, A. A. J., and Ridha, H. M. (2023) Enhancing NoC-based MPSoC performance: a predictive approach with ANN and guaranteed convergence arithmetic optimization algorithm, *IEEE Access*. https://doi.org/10.1109/ACCESS.2023.3305669

[12] Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., and Mirjalili, S. M. (2017) Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems, *Advances in Engineering Software*, 114:163–191. https://doi.org/10.1016/j.advengsoft.2017.07.002

[13] Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. (2000) A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, *Parallel Problem Solving from Nature PPSN VI: 6th International Conference, Paris, France, September 18–20, 2000 Proceedings*, Springer, pp. 849–858. https://doi.org/10.1007/3-540-45356-3_83

[14] Lin, Q., Li, J., Du, Z., Chen, J., and Ming, Z. (2015) A novel multi-objective particle swarm optimization with multiple search strategies, *European Journal of Operational Research*, 247(3):732–744. https://doi.org/10.1016/j.ejor.2015.06.071

[15] Moein-Darbari, F., Khademzade, A., and Gharooni-Fard, G. (2009) Cgmap: a new approach to network-on-chip mapping problem, *IEICE Electronics Express*, 6(1):27–34. https://doi.org/10.1587/elex.6.27

[16] GE, F., and WU, N. (2010) Genetic algorithm based mapping and routing approach for network on chip architectures, *Chinese Journal of Electronics*, 19(1):91–96. https://cje.ejournal.org.cn/article/id/5132

[17] Tosun, S., Ozturk, O., and Ozen, M. (2009) An ILP formulation for application mapping onto network-on-chips, *Proceedings of the 2009 International Conference on Application of Information and Communication Technologies*, IEEE, pp. 1–5. https://doi.org/10.1109/ICAiCT.2009.5372524

[18] Liu, Y., Ruan, Y., Lai, Z., and Jing, W. (2011) Energy and thermal aware mapping for mesh-based NoC architectures using multi-objective ant colony algorithm, *Proceedings of the 2011 3rd International Conference on Computer Research and Development*, Vol. 3, IEEE, pp. 407–411. https://doi.org/10.1109/ICCRD.2011.5764225

[19] Jang, W., and Pan, D. Z. (2012) A3MAP: Architecture-aware analytic mapping for networks-on-chip, *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 17(3):1–22. https://doi.org/10.1145/2209291.2209299

[20] Tosun, S. (2011) New heuristic algorithms for energy aware application mapping and routing on mesh-based NoCs, *Journal of Systems Architecture*, 57(1):69–78. https://doi.org/10.1016/j.sysarc.2010.10.001

[21] Sahu, P. K., Sharma, A., and Chattopadhyay, S. (2012) Application mapping onto mesh-of-tree based network-on-chip using discrete particle swarm optimization, *Proceedings of the 2012 International Symposium on Electronic System Design (ISED)*, IEEE, pp. 172–176. https://doi.org/10.1109/ISED.2012.17

[22] Obaidullah, M., and Khan, G. N. (2017) Application mapping to mesh NoCs using a tabu-search based swarm optimization, *Microprocessors and Microsystems*, 55:13–25. https://doi.org/10.1016/j.micpro.2017.09.004

[23] Wang, X., Sun, Y., Gu, H., and Liu, Z. (2018) WOAGA: A new metaheuristic mapping algorithm for large-scale mesh-based NoC, *IEICE Electronics Express*, 15(17):20180738. https://doi.org/10.1587/elex.15.20180738

[24] Alagarsamy, A., Gopalakrishnan, L., Mahilmaran, S., and Ko, S.-B. (2019) A self-adaptive mapping approach for network on chip with low power consumption, *IEEE Access*, 7:84066–84081. https://doi.org/10.1109/ACCESS.2019.2925381

[25] Alagarsamy, A., and Gopalakrishnan, L. (2018) MBA: a new cluster based bandwidth and power

aware mapping for 2D NoC, *Proceedings of the 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*, IEEE, pp. 1–4. https://doi.org/10.1109/ICCSDET.2018.8821150

[26] Wang, X., Sun, Y., and Gu, H. (2019) BMM: A binary metaheuristic mapping algorithm for mesh-based network-on-chip, *IEICE Transactions on Information and Systems*, 102(3):628–631. https://doi.org/10.1587/transinf.2018edl8208

[27] Mohiz, M. J., Baloch, N. K., Hussain, F., Saleem, S., Zikria, Y. B., and Yu, H. (2021) Application mapping using cuckoo search optimization with Lévy flight for NoC-based system, *IEEE Access*, 9:141778–141789. https://doi.org/10.1109/ACCESS.2021.3120079

[28] Darbandi, M., Ramtin, A. R., and Sharafi, O. K. (2020) Tasks mapping in the network on a chip using an improved optimization algorithm, *International Journal of Pervasive Computing and Communications*, 16(2):165–182. https://doi.org/10.1108/IJPCC-07-2019-0053

[29] Sikandar, S., Baloch, N. K., Hussain, F., Amin, W., Zikria, Y. B., and Yu, H. (2021) An optimized nature-inspired metaheuristic algorithm for application mapping in 2D-NoC, *Sensors*, 21(15):5102. https://doi.org/10.3390/s21155102

[30] Mehmood, F., Baloch, N. K., Hussain, F., Amin, W., Hossain, M. S., Zikria, Y. B., and Yu, H. (2022) An efficient and cost effective application mapping for network-on-chip using Andean condor algorithm, *Journal of Network and Computer Applications*, 200:103319. https://doi.org/10.1016/j.jnca.2021.103319

[31] Amin, W., Hussain, F., and Anjum, S. (2022) iH-PSA: An improved bio-inspired hybrid optimization algorithm for task mapping in Network on Chip, *Microprocessors and Microsystems*, 90:104493. https://doi.org/10.1016/j.micpro.2022.104493

[32] Coello Coello, C. A., and Salazar Lechuga, M. (2002) MOPSO: A proposal for multiple objective particle swarm optimization, *Proceedings of the 2002 Congress on Evolutionary Computation (CEC'02)*, Vol. 2, IEEE, pp. 1051–1056. https://doi.org/10.1109/CEC.2002.1004388

[33] Chatterjee, S., Sarkar, S., Dey, N., Ashour, A. S., and Sen, S. (2018) Hybrid non-dominated sorting genetic algorithm: II-neural network approach, *Advancements in Applied Metaheuristic Computing*, IGI Global, pp. 264–286. https://doi.org/10.4018/978-1-5225-4151-6.ch011

[34] Awadallah, M. A., Makhadmeh, S. N., Al-Betar, M. A., Dalbah, L. M., Al-Redhaei, A., Kouka, S., and Enshassi, O. S. (2024) Multi-objective Ant Colony Optimization, *Archives of Computational Methods in Engineering*, pp. 1–43. https://doi.org/10.1007/s11831-024-10178-4

[35] Zouache, D., Abualigah, L., and Boumaza, F. (2024) A guided epsilon-dominance arithmetic optimization algorithm for effective multi-objective optimization in engineering design problems, *Multimedia Tools and Applications*, 83(11):31673–31700. https://doi.org/10.1007/s11042-023-16633-x

[36] Kumar, A. S., and Reddy, B. N. K. (2023) An efficient real-time embedded application mapping for NoC based multiprocessor system on chip, *Wireless Personal Communications*, 128(4):2937–2952. https://doi.org/10.21203/rs.3.rs-880912/v1

[37] Taheri, S. M., and Hesamian, G. (2013) A generalization of the Wilcoxon signed-rank test and its applications, *Statistical Papers*, 54(2):457–470. https://doi.org/10.1007/s00362-012-0443-4

Neural Network-Based Wind Speed Prediction and Evaluation Using the NREL Wind Integration National Dataset

Haibo Peng[1*], Xinwei Xie[2], Tao Zou[3]
[1]Yunnan Open University, Kunming 650500, Yunnan, China
[2]Zhuhai College, Jilin University, Zhuhai 519090, Guangdong, China
[3]School of Management, Northwestern Polytechnical University, Xi'an 710021, Shaanxi, China
E-mail: penghaibo@ynou.edu.cn
[*]Corresponding author

Keywords: WS forecasting, machine learning models, ensemble techniques, renewable energy prediction, feature importance analysis

Received: August 7, 2025

Accurate wide speed (WS) forecasting is an important factor in optimizing wind energy production and keeping power in a stable condition with lower operation costs. Therefore, this study is going to compare different machine learning (ML) models such as Support Vector Regression (SVR), Decision Trees (DT), Gradient Boosting Machines (GBM), Multi-Layer Perceptron (MLP), and ensemble techniques using Voting and Stacking. These models are trained and tested by taking data from the National Renewable Energy Laboratory's Wind Integration National Dataset, also known as the WIND Toolkit, which is a high-resolution turbine location dataset of wind resources for nearly all locations within the contiguous United States and significant portions of Canada and Mexico. The key performance metrics include testing using MSE, RMSE, and R^2 for each model. In fact, the results confirm that the relative performance of ensemble methods has greater stability and forecasting accuracy, with the Voting model outperforming all the other individual models, thus capturing complex patterns in the WS data. Consequently, SHAP analysis underlined capacity factor and longitude as geographic location features among the most impactful, entailing the importance of such variables in the prediction tasks related to wind. These results underline the potentiality of ensemble techniques in view of robust estimation of WS, which would help the management of renewable energy in a more reliable way. Further, the work will focus on some deep learning (DL) approaches and extra meteorological data in order to improve model performance for the sustainable integration of wind energy into the power grid.

Povzetek: Študija primerja več ML-modelov in ansambelske metode (voting/stacking) za napoved hitrosti vetra na podatkih WIND Toolkit ter pokaže, da ansambli dosežejo stabilnejše in natančnejše napovedi, pri čemer SHAP izpostavi faktor izkoriščenosti in geografsko lego kot ključna vpliva.

1 Introduction

One of the most promising sources of renewable energy is wind power, serving as a very sustainable substitute for fossil fuels. High-precision WS forecasting is an essential requirement for optimal wind energy production, grid stability, and, very crucially, operational cost reduction. These require sophisticated methods for the prediction of this intermittently distributed resource, capable of modeling complex nonlinear patterns in the behavior of winds. ANNs are quite suitable for the task of learning from an enormous quantity of data and generalizing patterns that might not be easily recognizable by traditional statistical model bracelets [1]. The work leverages the comprehensive wind data set, National Renewable Energy Laboratory's WIND (WIND Toolkit), which comprises wind resource geographic coordinates, wind capacity, and capacity factors. Such highly detailed datasets constitute one of the key inputs into a neural network model that can very accurately predict WS across a wide range of locations and conditions. The high-resolution wind data that the WIND Toolkit offers enables researchers to study the variations in wind patterns and can, therefore, increase the resolution of predictive models, addressing one of the primary challenges of renewable energy forecasting[2]. These are traditional and ML algorithms, like SVR, Nu SVR, DT, K-Nearest Neighbors (KNN), GBM, and MLP. Each method has advantages with respect to both linear and nonlinear data; however, the ML models seem much more powerful for high-dimensional and complex structures. The ensemble methods, like the stacking ensemble and voting ensemble, for combining models are intended to improve their accuracy further by capitalizing on each other's strengths

in order to offset their individual weaknesses and to make more robust predictions [3]. The major metrics allowing assessment of the model performance were calculations of MSE, RMSE, and Coefficient of Determination (R^2). MSE and RMSE give the average magnitude of an error for a forecasted variable. R^2 describes how well the model explains the variance of WS. These are the metrics that, upon analysis, the best model emerged as Voting Ensemble, and that goes without saying how such ensemble models will always do better on jobs that involve the prediction of complex datasets like WS [4]. Primary metrics to assess the performance of the model involved computations for MSE, RMSE, and R^2, RMSE and MSE provide insight into the absolute error margins of a predicted value, whereas R^2 measures how well the model describes the variance in WS data. After analysis, the metrics identified the Voting Ensemble as the best model, demonstrating how ensemble models often excel in tasks involving the prediction of complex datasets, such as WS [5]. The findings of the study thus outline that correct WS forecasting models are of prime importance in integrating the variable resource of wind energy into the grid in a sustainable way. Applications in the development of more sophisticated models through ML techniques continue to evolve; this will lead to improved operational efficiency and stability within renewable energy systems. The present work, therefore, overviews, by adapting both traditional and ML models, the practical applications of predictive analytics on renewable energy. For similar methodologies, access to the WIND Toolkit data; see relevant research that has utilized this dataset [6].

2 Literature review

With the increasing demand for clean and renewable sources over the past years, research into wind energy as a friendly and sustainable solution to the environment has increased. Precise forecasting of WS is essential for maximizing wind energy production, cutting operating expenses, and preserving power system stability. Over the years, there have been trials with various models, from mere statistical models to advanced hybrid ones, through which experts seek to improve the accuracy and reliability of WS predictions. This literature review discusses 15 relevant works, detailing the objectives, methodologies employed, findings, and contributions of each author to the field. The work of Blanchard and Samanta [7] was directed at the use of neural networks in the forecasting of WS. It was dedicated to testing whether neural networks are adequate for proper forecasting of WS data, including under conditions of changing meteorological parameters. With the inclusion of meteorological and geographical input, this model proved substantially improved for the purpose of good estimation as compared to the conventional statistical models, especially the short-term forecasts. The findings demonstrated how complicated, nonlinear patterns in the WS data may be learned by neural networks, which could then be used to develop better real-life applications of wind energy resource management and operational efficiency. Nielson et al. [8]focused on improving the prediction of wind turbine power by incorporating atmospheric inputs in ANNs. Such a study was aimed at improved accuracy in the integration of variables like temperature, humidity, and atmospheric pressure with WS. The ANN model resulted in a significant improvement in the accuracy of power prediction compared to models considering solely the WS variable, especially in meteorologically unpredictable conditions. Results showed that atmospheric data helped not only in refining the prediction but also in providing key insight into wind turbine optimization for efficiency and energy management.

Rahman et al. [9] gave a succinct overview of deep neural network forecasting strategies for wind time series. The purpose of this research was to examine how well DNN models performed, more so those that rely on techniques such as LSTM and GRU, in predicting WS over different forecast horizons. The research also proved that DNN-based approaches resulted in a significant reduction of the error rate in WS forecasting compared to traditional models, basing the findings on predictive accuracy and the robustness of the models. These findings highlighted the potential of LSTMs and GRUs to adapt nonlinear and complex temporal patterns that are deemed characteristic in wind data, and hence their effective applications in areas where precise long-term forecasts of wind are needed. Dolatabadi et al. [10]proposed a novel hybrid DL WS forecasting model developed by combining the DWPT technique with a Bi-LSTM network. This work was designed to provide improved accuracy on the basis of the combined use of DWPT for the decomposition of WS signals and the Bi-LSTM network, which would learn temporal dependencies in both forward and backward flows. It performed better than conventional methods in handling the inherent non-stationarities and fluctuations in WS data. The model was effective in producing short- and long-term wind forecasts of high accuracy, thus having great potential for applications in energy management and wind farm operations. Lydia et al. [11]reviewed various forecast models on WS and wind power with an emphasis on enhancing accuracy and reliability for renewable energy. In the paper, the performance comparison of several forecasting models, such as the statistical models, the ML model, and the hybrid models, was pursued to forecast power output and WS. The outcomes showed that the hybrid model, which integrated both statistical and ML approaches, achieved higher accuracy compared to a single model, especially when the wind conditions were fluctuating. In this way, results showed the major role played by hybrid techniques in efficient forecasting, management of wind power, and grid stability to support the integration into variable renewable sources in power systems. Routray et al. [12]had different models adopted for the forecasting of wind power and WS; they drew conclusions on the best model to improve accuracy for renewable energy applications. This study will compare the various models' performance in forecasting WS and output power, including multi-models like statistical, ML, and hybrid models. From the outcomes, it was observed that the hybrid model, when combined with the statistical method, ML, gives better accuracy than a single model, especially when the wind is variable. The findings

demonstrated that the integration of renewable energy sources into power networks and the effective management of wind power and grid stability depend on hybrid forecasting methodologies. Malik et al. [13]had different models adopted for the forecasting of WS and wind power; he drew conclusions on the best model to improve accuracy for renewable energy applications. This study will compare the various models' performance in forecasting WS and output power, including multi-models like statistical, ML, and hybrid models. From the outcomes, it was observed that the hybrid model, when combined with the statistical method, ML, gives better accuracy than a single model, especially when the wind is variable. The outcomes indicated that the hybrid forecasting techniques are very vital for efficient management of grid stability and wind power, and therefore, incorporating renewable energy sources into power systems. Wang et al. [14] developed an ANN-based model that could effectively predict the extreme response of floating offshore wind turbines under different operating conditions. The study was performed to enhance safety and durability through sensible forecasting of extreme load responses due to harsh environmental factors. Using ANN in the modeling allowed for complex nonlinear interactions between wind, wave, and turbine dynamics with reliable predictions for extreme responses. The results brought out that the model can be an enabler in improving both the risk assessment and maintenance planning of the offshore wind energy infrastructure, in view of ensuring stable performance under adverse conditions at sea.

Ahmad et al. [15]have proposed an intelligent control system based on fuzzy logic integrated with the artificial neural network model to enhance stability and efficiency for an offshore wind turbine that floats and has four water columns that oscillate. The objective is to optimize performance dynamically against fluctuating sea and wind conditions. The fuzzy logic-based control system interacted with the ANN in order to enhance the turbine response due to variations in the ambient conditions, thus providing a smoothing effect toward fluctuations both in power output and mechanical stresses. Results obtained from this study indicated that such an integrated approach provided better adaptability and control, while its potential was assessed as one for reliable energy generation to be employed in offshore environments. Assaf et al. [16] examined in depth how well the neural network-based models performed for short-term solar irradiance forecasts. Then, all the neural network models, Convolutional Neural Networks, and Recurrent Neural Networks have been accurate for short-term forecasting, especially when they have been trained based on meteorological and environmental data. Hence, it is underlined that the potential of neural networks to enable solar energy management is to optimize the performance of the photovoltaic system, given valid and reliable forecasts of irradiance. Nielson et al.[8] studied the effect of adding atmospheric inputs to ANNs in improving wind turbine power predictions. There was an attempt to make the predictions more realistic, actually feeding the ANN model atmospheric variables like temperature, humidity, and pressure, in addition to WS. Those additional atmospheric parameters have greatly improved the model's predictive accuracy when compared against models that were driven exclusively by WS. This was effective, in particular, in the capturing of much atmospheric interaction. It showed potentials in optimizing the generation of energy from the wind toward supporting more efficient operations of turbines. This was especially effective in capturing a lot of atmospheric interaction. It showed its potential in optimizing wind energy generation and supporting more efficient turbine operation.

Jiang et al. [17]proposed an automated framework for wind power prediction that integrated multi-time scale analysis and temporal attention mechanisms. The proposed framework leverages the Temporal Fusion Transformer model, which effectively captures temporal dependencies at different scales. This study found that such a method drastically enhances the prediction accuracy of both short- and medium-term forecasts and, therefore, holds great potential to allow better grid reliability and energy management.

Liu et al. [18]proposed the GAT-based model combined with the frequency-enhanced mechanism to improve short-term WS forecasting. The GAT component captures the complex spatial dependencies between WS stations, while the frequency-enhanced mechanism deals with temporal patterns. This hybrid model performed distinctly better than the state-of-the-art approaches, especially in finding the dynamic behavior of WSs within the small-scale forecasting horizon. Wang et al. [19]proposed a hybrid model by incorporating CNN features with the Informer architecture for short-term wind power prediction. Due to the fact that CNN extracts the spatial features, the informer can handle long-sequence time series with effective learning of the dependencies in space and time. It reported superior performance compared to traditional methods for the non-stationary and fluctuating nature of wind power data. Huang et al. [20]explored the application of Transformer-based models for wind power forecasting. The study demonstrated how wind power forecasting jobs can benefit from the use of transformers, which are well-known for their ability to capture long-range dependencies in sequential data. The suggested model achieved higher accuracy and efficiency compared to traditional recurrent neural network models, indicating its potential for improving wind power prediction.

Mukendi et al. [21]have proposed a new methodology regarding WS and power forecasting using shape-wise feature engineering. This work significantly improved the robustness of both CNN-LSTM and auto-regressive models against noise by changing the supplied data's form. The results of this work indicate that this technique gives consistent accuracy for different forecast horizons and, hence, is promising for turbine control systems and resource planning in energy production. Keisler and Le Naour [22] proposed WindDragon, an automated DL framework to enhance short-term wind power forecasting. Numerical weather predictions as well as automated ML techniques are employed in model selection and

hyperparameter optimization for this approach. It has been proved that the predictability from WindDragon ensures better wind energy integration into the power grid as well as helps various efforts toward net-zero carbon emissions.

3 Methodology

This section describes the methodology applied to the development and evaluation of predictive models for WS forecasting by availing ML algorithms and ensemble techniques of the WIND from the National Renewable Energy Laboratory. It involves data preparation, training of models, procedure evaluation, and optimization in order to achieve the best result in terms of accuracy and reliability for WS forecasts.

3.1 Data collection and preprocessing

The data utilized in this analysis were fetched from the WIND Toolkit, a source of excellent wind data that also includes geographic coordinates and wind capacity, along with capacity factors. These inputs form a very important set of variables for training predictive models. First, the raw wind data were normalized to make them more consistent and thus improve the model's performance by reducing biases that could arise because of the presence of different scales among the data. Then, the following data was randomly split into two subsets: 80% for training and 20% for testing, enabling the evaluation and validation of this robust model.

3.2 Model development and training

Several ML algorithms were implemented, ranging from conventional models such as SVR and Nu SVR to more advanced algorithms, including DT, KNN, GBM, and MLP. Each of the models was then trained on the WIND Toolkit dataset to capture both linear and nonlinear patterns that could be helpful in improving predictive accuracy. Hyperparameter tuning for each model was performed by systematic grid searches and also by means of nature-inspired algorithms such as genetic algorithms that refine the most important hyperparameters: C, ϵ, and γ-when it came to SVR models. As for the ensemble models, there were two main techniques that improved the prediction accuracy of algorithms. In its process, the Voting Ensemble method pools predictions from each individual model involved, whereby a vote is assigned to each based on the output given. However, the Stacking Ensemble technique takes various models as base learners whose outputs have to go through another meta-learner for the actual prediction. In that respect, this stacked approach leverages the strengths of individual models, whereby it places the ensemble in a position to make up weaknesses that could be common in just one model. A model of wind power forecasting begins with six steps in order, as depicted in Fig. 1. First, Step 1 is data collection and the preparation of that data to analyze; Step 2 is the determination of necessary input features and target values that one would need from the dataset. It then further divides the dataset into training and testing subsets in a ratio of 80:20, respectively, on which model training and evaluation are based. Step 4 describes the wind power forecast by using different ML algorithms: SVR, Nu SVR, DT, KNN, GBM, and MLP. It involves two ensemble techniques, namely Voting and Stacking, applied in Step 5 to improve accuracy by aggregating the strengths represented by the different models. Lastly, Step 6 identifies the best-performing model according to several performance metrics, which is Voting Ensemble.

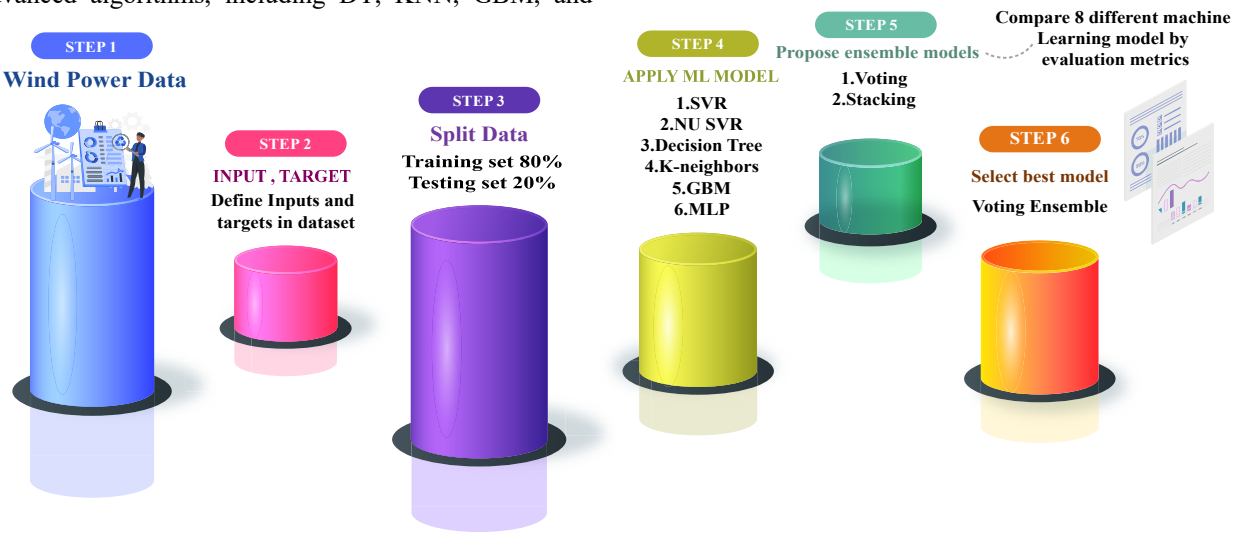


Figure 1: Workflow for wind power prediction using ML models

The organized methodology for this is presented in Fig. 1 and represents one of the most holistic wind power prediction methods wherein both ML models and ensemble techniques have been combined to provide strong and precise forecasts. Fig. 2, The KNN classification process for a target point within a two-class dataset. The figure above shows the identification of the target point and segregation of the dataset into two classes: Class 1 in red and Class 2 in blue. The second panel shows the KNN distances computed from the target point to its nearest neighbors, drawing on the target to connect to the nearest points in both classes. In the third panel, it does the classification by finding the majority class in a defined neighborhood. This neighborhood is usually taken to be a

circle around the target point, within which it searches for the majority of the classes of the points that fall there. Accordingly, based on the majority of neighbors from those points, the target point will fall into one of the two classes, hence illustrating how KNN assigns a class label based on spatial proximity.

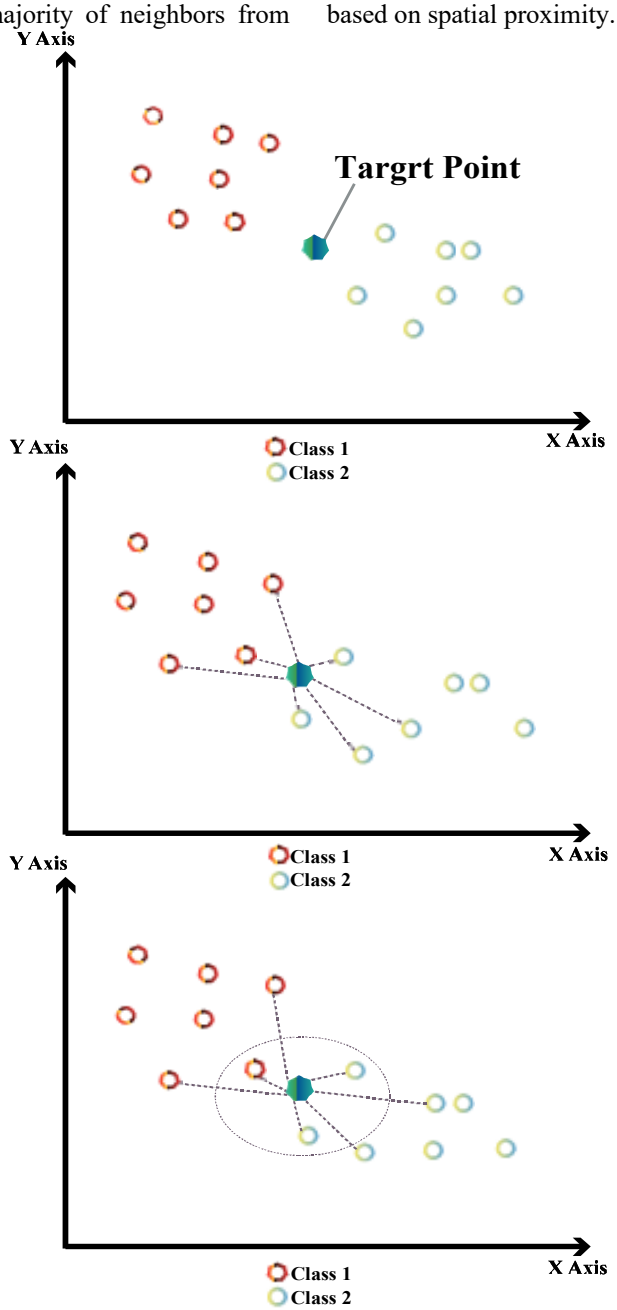


Figure 2: KNN classification process for the target point

Fig. 3 shows the architecture of the MLP, a general class of ANNs used for predictive modeling and classification tasks. The major constituents of the network are three layers, including the Input Layer, which is the layer fed with raw data input; one or more Hidden Layers, where neurons or nodes process and transform inputs through weighted connections and an activation function, respectively; and the Output Layer, which provides a prediction or classification based on the transformed data. Each neuron connects itself to neurons in the adjacent layer in a fully connected manner. This enables the network to learn complex nonlinear relationships within data. Generally speaking, through training, the weights within an MLP will be adjusted, which will enable the model to generalize patterns in order to be more accurate. Thus, this is a diverse model that can be applied to many different ML applications.

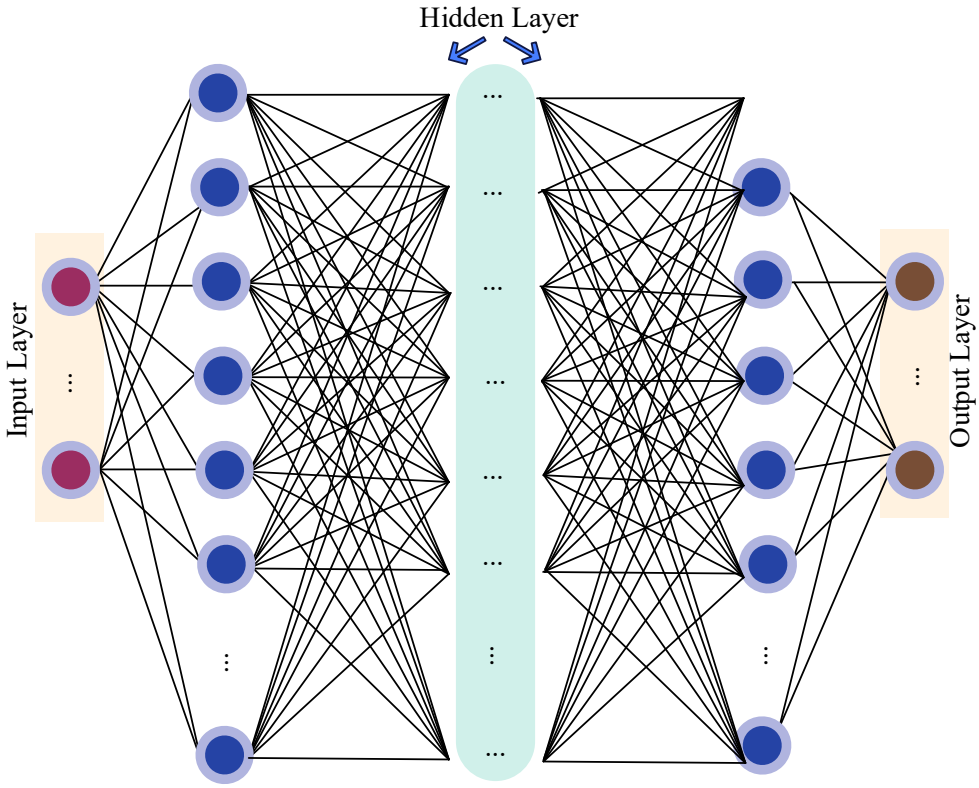


Figure 3: The architecture of an MLP network

Fig. 4 shows the structure of one variety of DT, which serves as a guide for sequential decisions. It is attached to the top node, and each decision node- a diamond-shaped icon- offers a 'yes'/'no' choice, each of which then splits into two further divisions. Each individual pathway illustrates one possible stream of decisions contingent upon some condition or criterion. This process continues down through the tree until terminal nodes, labeled as Steps, such as Step 1 and Step 2, are reached, from which the ultimate outcome or action is determined by the previous decisions. The structure can also be straightforward: a step-by-step breakdown into complex decision processes for intuitive, organized problem-solving, classification, or predictions. DT finds broad applications in ML, especially in applications where decisions should be interpretable and structured in rules.

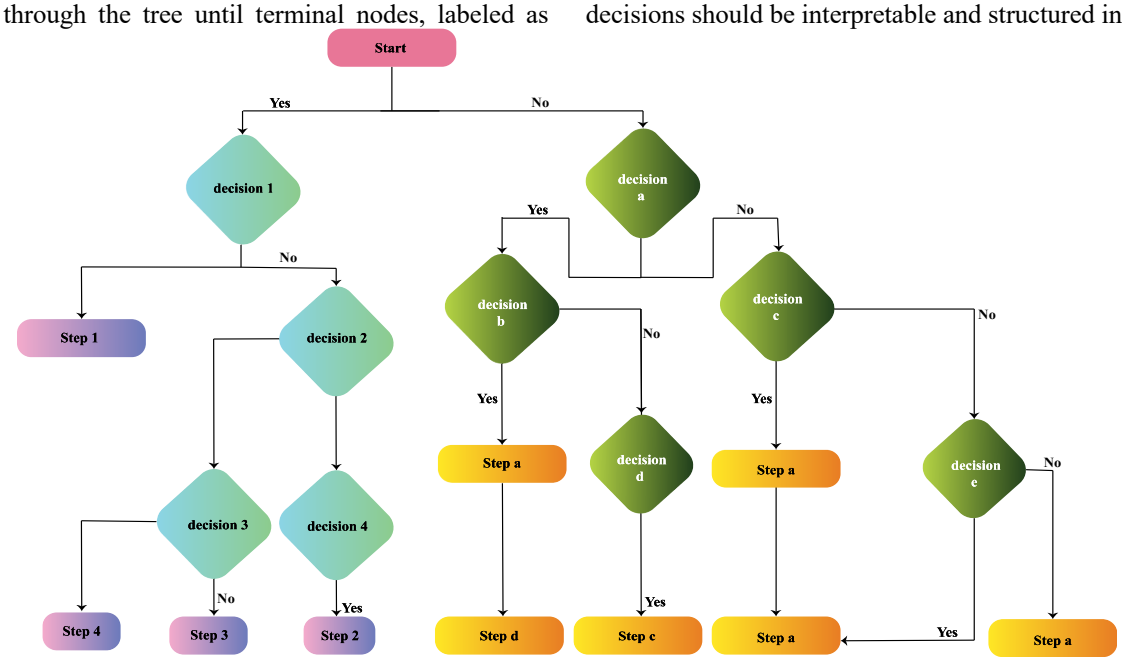


Figure 4: DT structure for sequential decision-making

Fig. 5 represents the training procedures of one of the most robust ensemble learnings in predictive modeling, which is the GBM. Where GBM constructs a lot of DTS in a greedy manner, each one of the trees tries to correct

mistakes made by the previous one. It starts by letting the model make predictions, using Tree 1 on the input data (X, Y), thereby returning residual errors $r_1 = y - \hat{y}_1$. In subsequent iterations, each tree (Tree 2, Tree 3, etc.) is trained on the residuals from the previous step (e.g., $X, r_1, , X, r_2, and\ so\ on$). This process is repeated recursively until the final tree, say Tree N, has been trained in the hope of further minimizing the prediction error. In progressively reducing these errors, GBM constructs one strong model that amalgamates the predictions of all the trees to make an accurate prediction. Hence, it works extremely effectively to handle complicated non-linear data relationships.

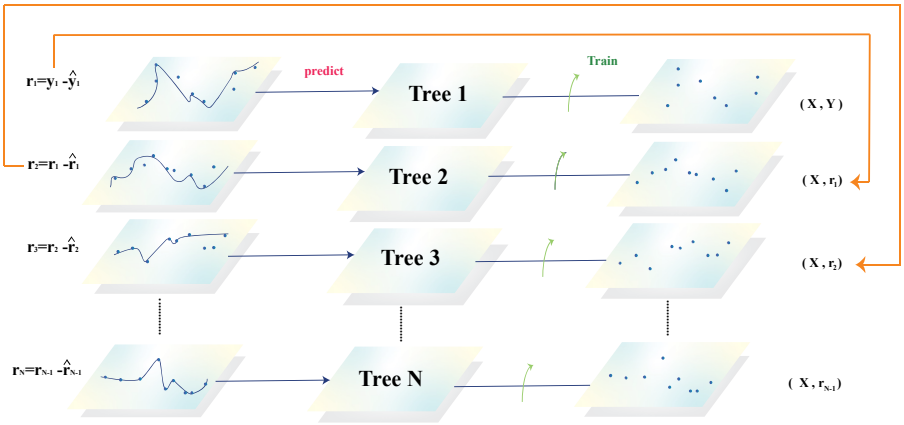


Figure 5: GBM's training process

Fig. 6 is the flowing process for hyperparameter optimization of the Nu SVR model. The flow starts with a normalization of data, followed by splitting the data into training and test sets at random. Initial parameters for the SVR are set, and the model is trained, followed by an evaluation. The approach utilizes nature-inspired algorithms, like genetic algorithms and particle swarm optimization, to adjust the SVR hyperparameters if it fails to achieve the stopping criterion. The optimized parameters are then experimented with on the SVR model. This will be an iterative process until it meets the stopping criteria, where one can obtain the final optimized SVR hyperparameters. It is a structured approach meant to ensure a well-tuned model that is able to fit the outcomes with the least possible error.

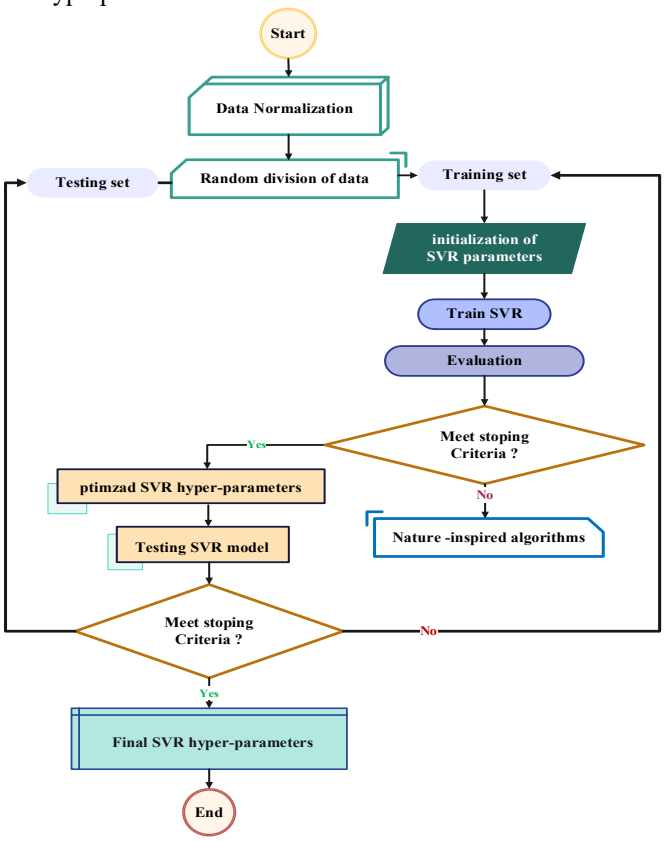


Figure 6: The hyperparameter optimization process for Nu SVR

Fig. 7 describes the steps to develop and test the SVR model. The steps start with the collection of experimental data. The following step is data preprocessing, which cleans the experimental data to make it ready for modeling. The resulting set of data is split into two subsets: Training data and Testing data. Identifying the critical hyperparameters of the SVR model, namely C, ϵ, and γ, is then performed by choosing those that optimize the performance of the model. The model goes through both SVR Training and SVR Testing to check its accuracy. Further, the model is subjected to measurement of correlation value between predicted and actual data to assess its prediction capability. This systematic workflow allows the SVR model to be well-trained and tested in order to yield reliable results.

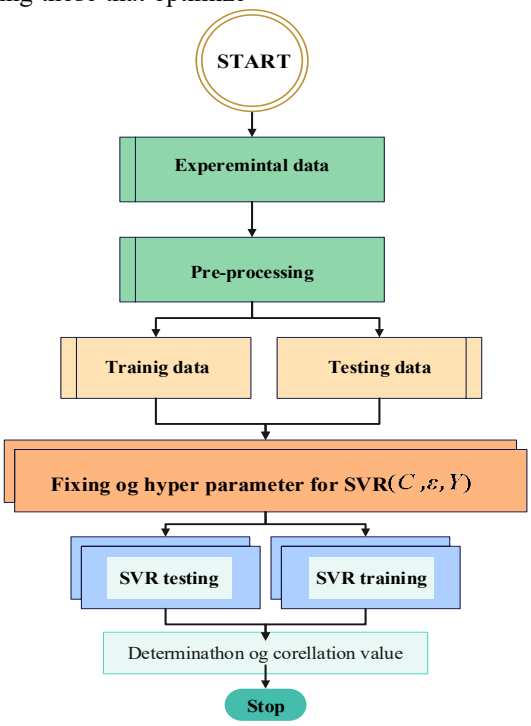


Figure 7: Workflow for SVR model training and evaluation

Fig. 8 provides an overview of the architecture in a Stacking Ensemble model. It presents a very general strategy in ML, where predictions of multiple base models are combined into one general model that normally has better performance. Given a dataset D, several separate models may be individually trained on this data, each learning something different from it. These models' outputs then form the inputs to a meta-learner or meta-model that learns to weigh and combine these predictions to come up with a final output. The Stacking Ensemble approach tends to result in even better predictive accuracy and robustness since it leverages the strengths of multiple models. This kind of structure is particularly useful when complex prediction tasks are at issue since it boosts performance by reducing biases specific to single models.

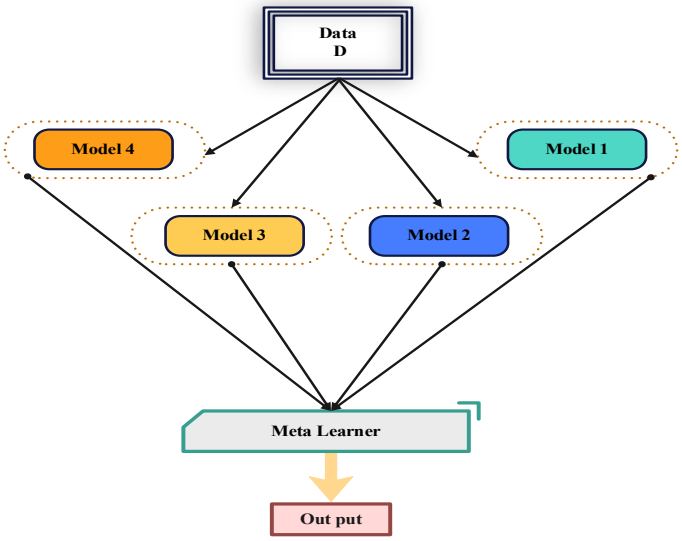


Figure 8: Stacking ensemble model structure

Fig. 9 is a multi-step image processing workflow using a voting mechanism to localize a target position within an image. Step 1 is whole scales the image through a low-pass filter to blur noise, then calculates at every location in the image the gradient to detect edges or intensity changes; Step 2 votes-sets an initial vote image to zero and starts a voting process were. The algorithm, therefore, calculates votes based on directional updates and decreases the angle parameter ϕ, gradually approaching a stopping criterion. Step 3 (Localize) completes localization by marking the final local maxima, followed by a threshold, in order to highlight the most probable positions regarding the target location. This technique comes in handy for object detection and position tracking applications where accuracy for such tasks is improved by iterative voting and thresholding.

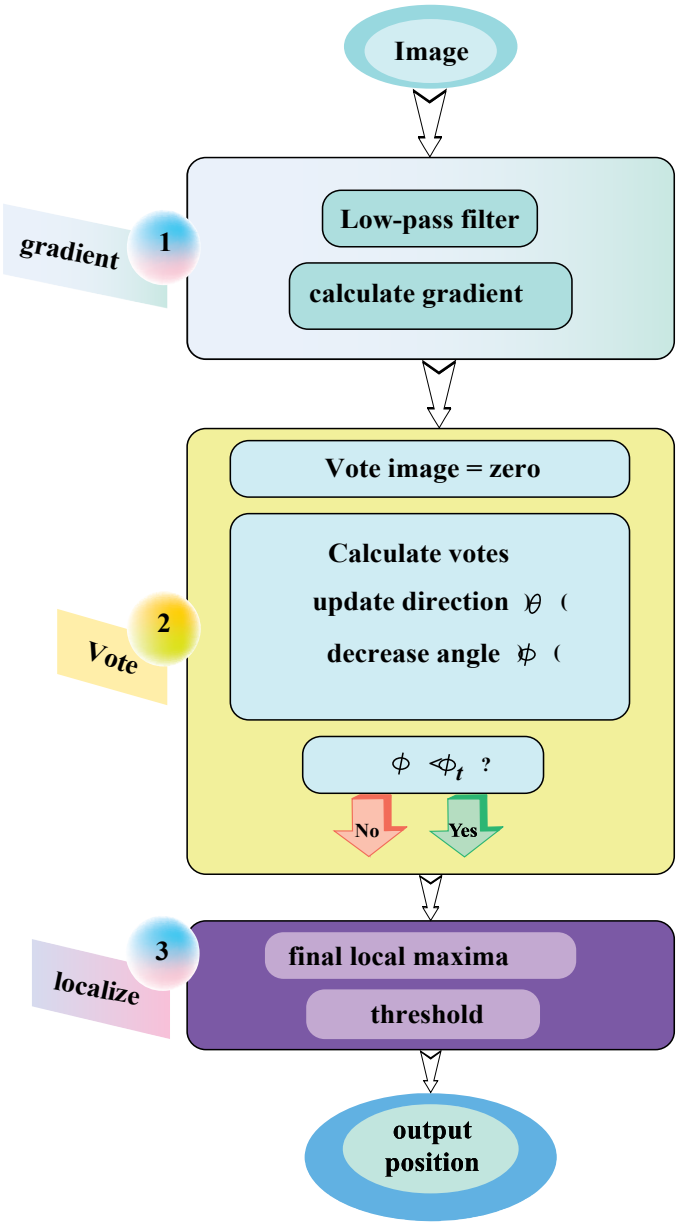


Figure 9: Image processing and voting mechanism for position localization

3.3 Evaluation metrics

All models, including the surrogate models, were tested against each other based on a set of metrics. Fig. 10 presents some common mathematical formulae along with metrics widely used for diagnostic analysis related to predictive model performance. These metrics span from the RMSE, representing the average magnitude of the prediction errors, up to the mean absolute error (MAE), that is, the average absolute difference between the estimated and actual values. MSE and the median absolute percentage error (MDAPE) further quantify the accuracy of the predictions captured by squared deviations and percentage errors, correspondingly. These metrics represent an overall seminal framework for assessing and comparing model accuracy, reliability, and suitability with regard to different tasks of prediction.

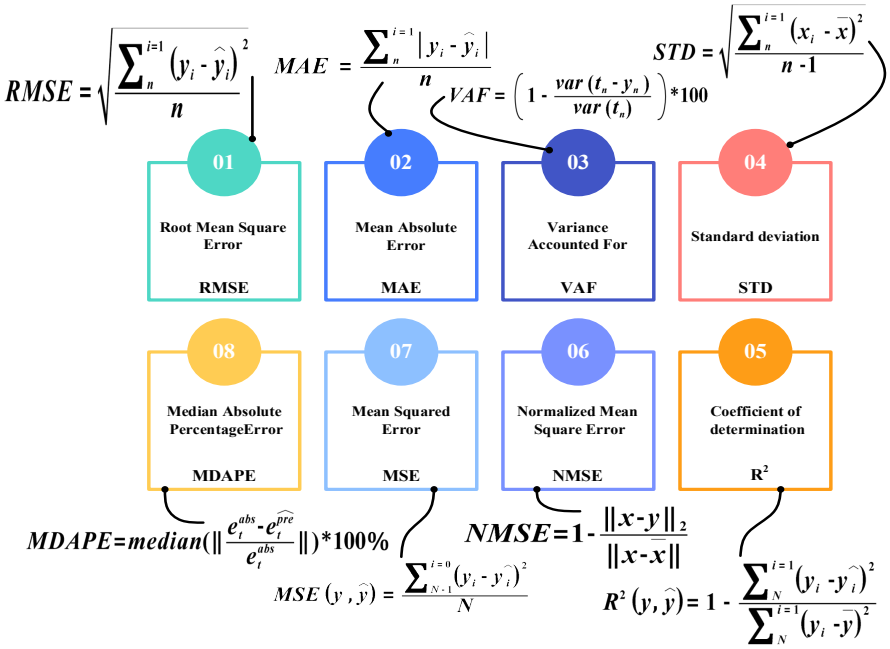


Figure 10: Performance metrics for model evaluation

The Root Mean Square Error (RMSE) is a measure of the magnitude of the prediction errors, heavily penalizing larger errors. MAE, being the mean absolute difference between the values that were anticipated and those that were seen, provides a straightforward measure of the size of the error in the predictions. Variance Accounted For (VAF) measures the proportion of variance in the actual values explained by the model as a percentage. Standard Deviation (STD) reflects the spread or variability of values around the mean. R^2 indicates how much of the variance in the actual data is captured by the model. Normalized Mean Square Error (NMSE) compares the model's error to the data's variance, helping assess relative prediction accuracy. The MSE is the mean of the squared differences between actual and predicted values; it penalizes larger errors. The MDAPE is precisely the median of the absolute percentage errors, being, therefore, less sensitive to outliers.

y_i: The actual (observed) value for the i-th data point in the dataset.

$\hat{y}_i$: The predicted value for the i-th data point is generated by the model.

n: Overall number of observations or data points in the dataset.

$\bar{y}$: The mean of the actual values in the dataset, calculated as $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$.

t_s and y_s: t_s in the VAF formula refers to the actual value, while the t_s is a predicted value at a certain instance or sample.

Var (t_s): It means the variance of the actual values, which is a measure of the dispersion of data around the mean.

x_i: A particular observed value within the dataset (employed in the formula for STD and NMSE) that will relate to a data point.

$\bar{x}$: The mean of the observed values x_i in the dataset, similar to $\bar{y}$ but used in different contexts in certain formulas.

e_{bs}^{th} and e_{bs}^{ref}: In MDAPE, e_bs^th is the theoretical value, and e_bs^th is the reference value, used in most fields where the relative error of measurement applies

N: Another representation of the total number of data points, sometimes used interchangeably with n.

$\|x - y\|_2$: In NMSE, this is used to refer to the Euclidean distance between two sets of values, x, which are actual, and y, which are predicted, and this captures magnitude information on the error.

C, ϵ, and γ: are the most general hyperparameters in SVR, though it was not shown here. These are parameters governing the margin, tolerance bound, and the kernel of the model.

4 Results and discussion

In this part, experimental outcomes are presented regarding a review that was done on the application of various ML models and ensemble techniques in the forecasting of WS using data from the WIND Toolkit, a product **of** NREL Wind Integration National Dataset. The results will highlight the performance evaluation of every different model, using the Mean Squared Error (MSE), RMSE, and R^2 metrics that will grant the ability to analyze the exactness and reliability of each different approach. The following paper compares traditional models, such as SVR and DT, to advanced ensemble methods for the optimal model configuration that robustly represents nonlinear variability in WS data. This, therefore, serves to bring out various weaknesses and strengths of each individual model while at the same time attempting to provide insight into how several models through ensemble methods could possibly improve predictive accuracy, particularly when it involves dealing with intrinsic wind variability. The results from each model are detailed in subsequent sections, along with their discussion in relation to implications for predicting WS in applications

involving renewable energy. The heatmap of Fig. 11 presents the correlations for five of the variables in the dataset: longitude, latitude, capacity, capacity factor, and WS.

The Pearson correlation coefficient's strength and direction are displayed by color intensity, with lighter hues denoting weaker or negative correlations and deeper blues denoting stronger positive correlations. The capacity factor is strongly positively related to WS at 0.8, and it could be said that, in general, this factor increases with increased WS, as higher WSs would normally imply higher energy output from wind turbines. The capacity factor increases moderately with latitude at 0.32 and longitude at 0.28. These may reflect some geographic influence on wind conditions and, hence, energy production. Other relationships are much weaker-for instance, the near-zero correlations among capacity and both longitude 0.014 and latitude -0.059 would suggest that, for this dataset, geographic location may have little effect on the actual placing of a turbine in terms of its capacity. The heatmap, in general, provides insight into how key variables interact by underlining the importance of WS in determining the capacity factor while showing a minimal geographic dependence on the capacity. This, in turn, can inform feature selection during predictive modeling by focusing on the more strongly correlated variables as one aims to develop an accurate model.

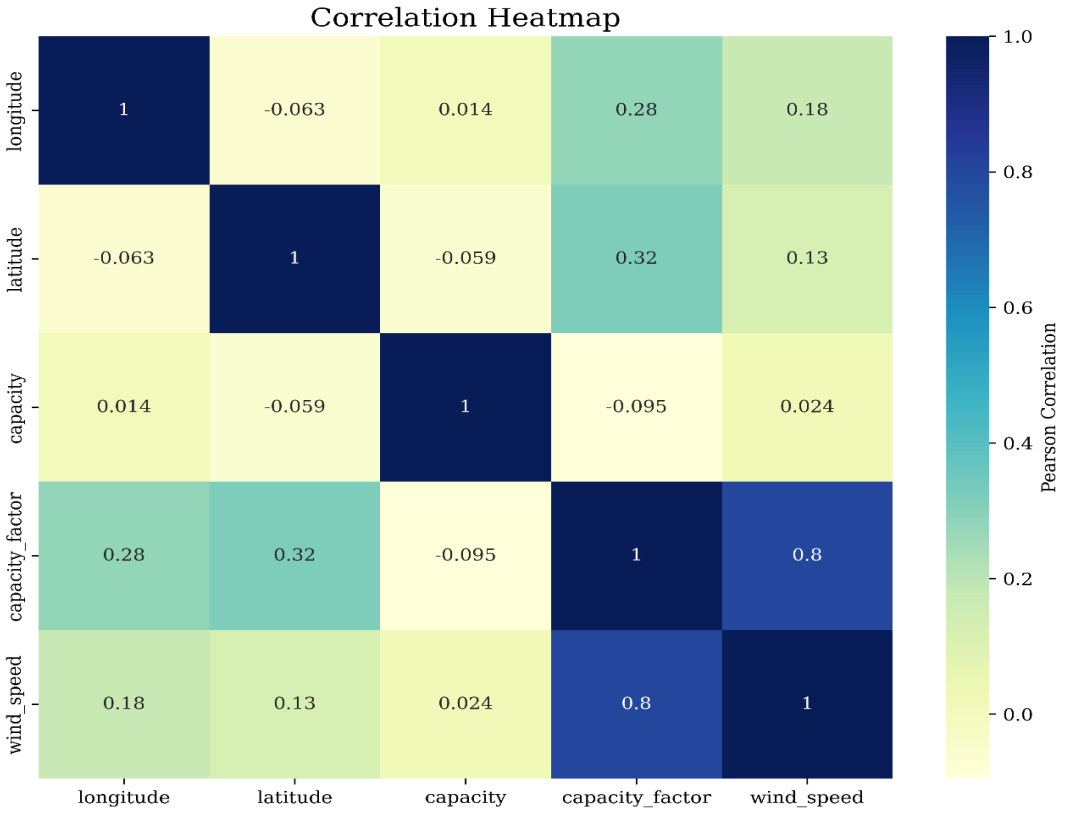


Figure 11: Correlation analysis of key wind energy variables

Fig. 12 shows the forecasted wind velocity values using the various models up to DT, GBM, KNN, MLP, Nu SVR, and SVR plotted against target values across a series of data points, i.e., row numbers. The topmost black line indicates the actual values of the target variable; each model's prediction is plotted as a different color line below it. Another insight that this plot shows is that some of the models tend to be stuck much closer to the target line, like the GBM and DT, and some are very off the target values, such as the SVR and Nu SVR models. This means that the ensemble methods such as GBM, along with tree-based models like DT, turn out to be more accurate; the best result comes in terms of prediction for the WS data. Because performance by SVR and Nu SVR may come out comparably poorer due to non-linear variation, it lacks the capability to capture the WS variations in a non-linear pattern. The above visualization strongly indicates the model performance difference. Hence, GBM and DT are the preferred data sources. Consistency in the predictions from each model on the available data tells us something concerning the resilience and dependability of the models in forecast applications.

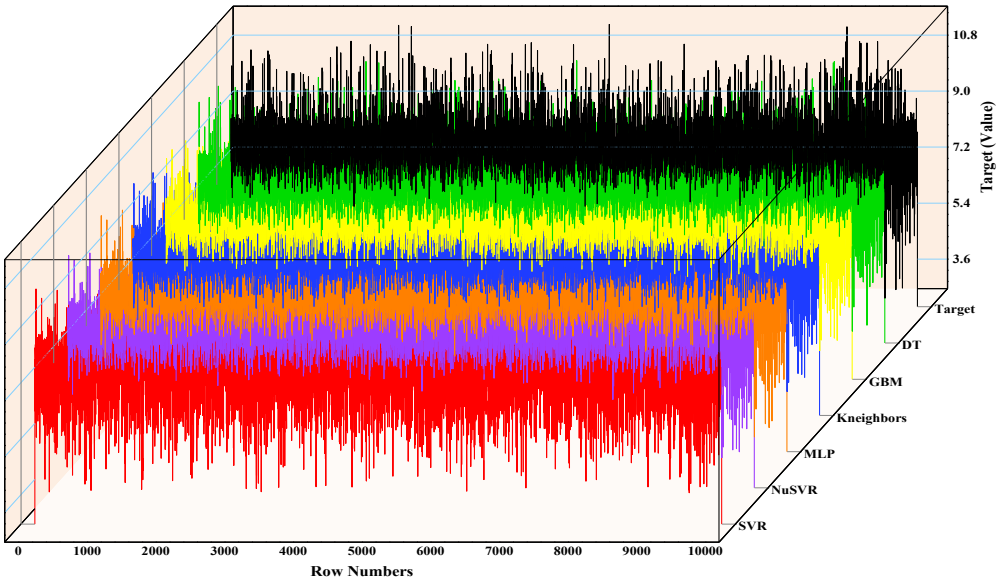


Figure 12: Comparison of model predictions against target WS values

Fig. 13 shows the comparisons between different models: DT, GBM, SVR, KNN, MLP, and Nu SVR. Different plots are drawn to show forecasted WS against actual values for all models. In each plot, the R^2 score of training and testing is also shown to present the strength of the model in explaining the variance in data. From the DT model, even though the R^2 score is very high on training, reaching 0.999, which depicts an almost perfect fit, there is a depression in the R^2-in-test at 0.861, which can be a possible sign of overfitting. The NBA with Theenza GBM, KNN, and MLP all depict performance that is balanced between the training and test sets since the respective R^2 scores are close to each other, therefore generalizing with less overfitting. Among them, KNN presents a test R^2 of 0.895, while GBM follows close at 0.887 and the MLP at 0.881. In contrast, SVR and Nu SVR depict very low R^2 values for both training and testing sets, which reflects the overall relatively poor performances of these two algorithms in modeling the underlying pattern of variation within the data. Overall, KNN, GBM, and MLP emerge as the most reliable models with firm generalization capabilities, while DT may not be that reliable for real-world applications due to overfitting, though with high training accuracy. The above analysis gives an idea that the ensemble and neighborhood-based methods might turn out to be more appropriate for the given task of WS prediction.

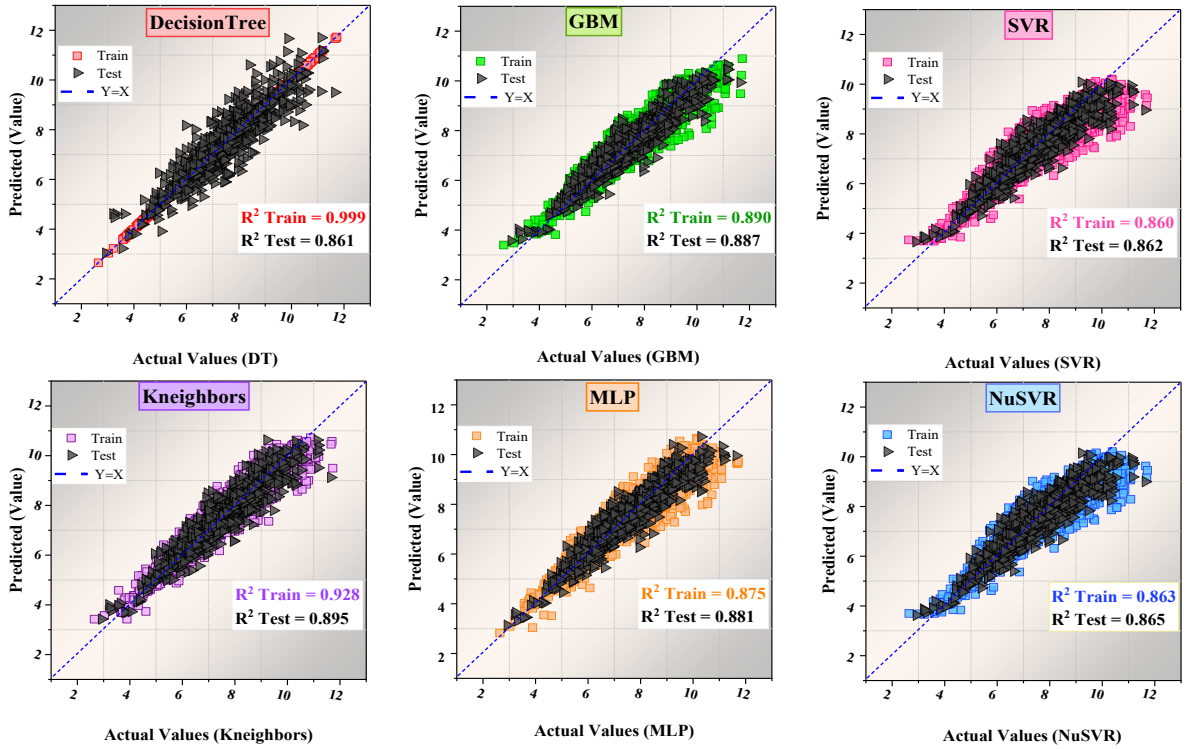


Figure 13: Model comparison based on predicted vs. actual WS values (R^2 analysis)

Fig. 14 is a distribution plot of the prediction error for each model, split into training and testing datasets. Each model error is color-coded as horizontal lines; zero error is colored red and indicates perfect predictions. First, the narrow error spread around zero in the DT and GBM models demonstrates that the model is highly accurate with less deviation in predicting actual values. On the other hand, some of the model results show a wider distribution of errors, such as Nu SVR and SVR models, reflecting larger deviations from actual values more often, especially within the test set. In particular, all the models show smaller error dispersion in the training set, while the test set can widen such dispersions, which is the typical case of loss when the models generalize to unseen data. In particular, the performance is generally maintained by models like DT and GBM but fails to generalize in the case of others, like Nu SVR. In general, the narrow distribution of the errors around zero obtained for both DTs and GBM models denotes their robustness and suitability to the variance in WS prediction tasks represented in this dataset.

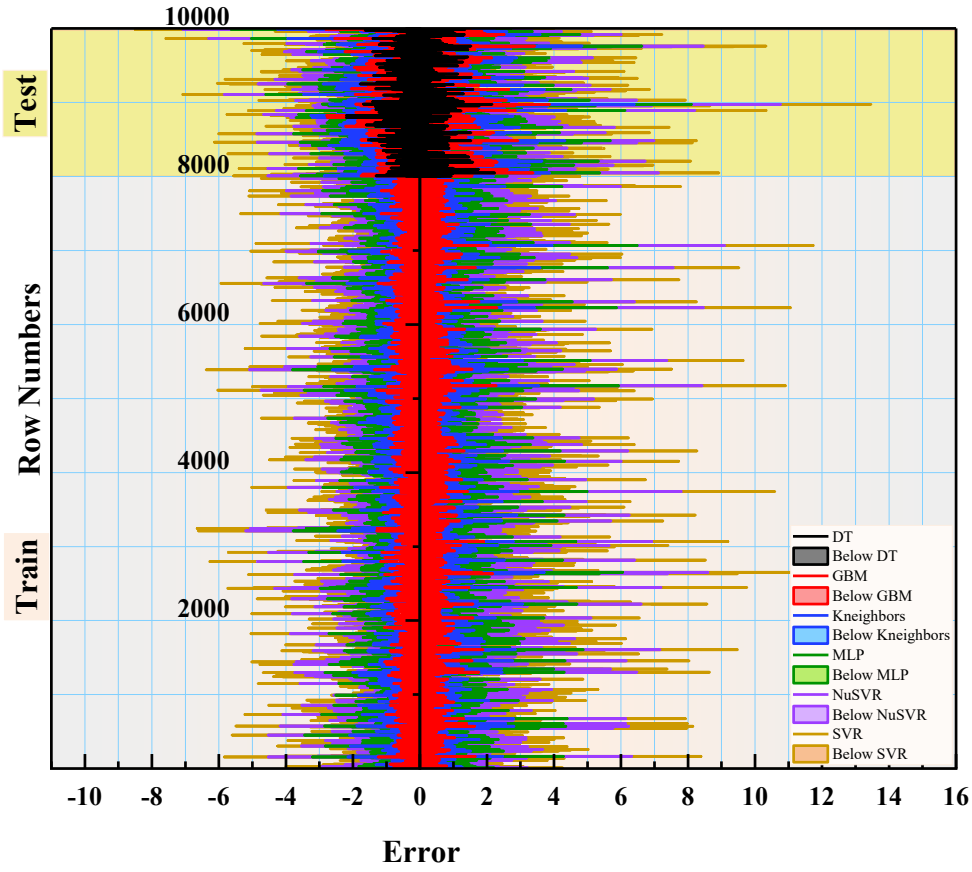


Figure 14: Error distribution of predicted WSs across training and testing sets

In Fig. 15, the boxplot displays the range and distribution of prediction errors for different models divided into training and testing sets. Since the boxplot IQRs center on zero, with their whisker extensions holding the outliers, this suggests that most of each model's error range is centered around zero. Models like DT and GBM have relatively narrow error distributions, especially in the training set, reflecting through the small values of IQRs that most prediction errors are close to zero. In contrast, models such as Nu SVR and SVR had a larger spread in the distribution of error, which had more evident outliers, indicating that their type of models might have generalization issues and be more sensitive to the great deviations of predictions on unseen data. However, KNN, MLP, and GBM presented very similar performance in the training and test sets, with a minimum spread of error, which turned into robust and reliable results. Boxplot analysis shown here suggests that ensemble and neighborhood-based models like GBM and KNN might provide more robust and really accurate predictions, while models like Nu SVR are more prone to variability, especially on new sets.

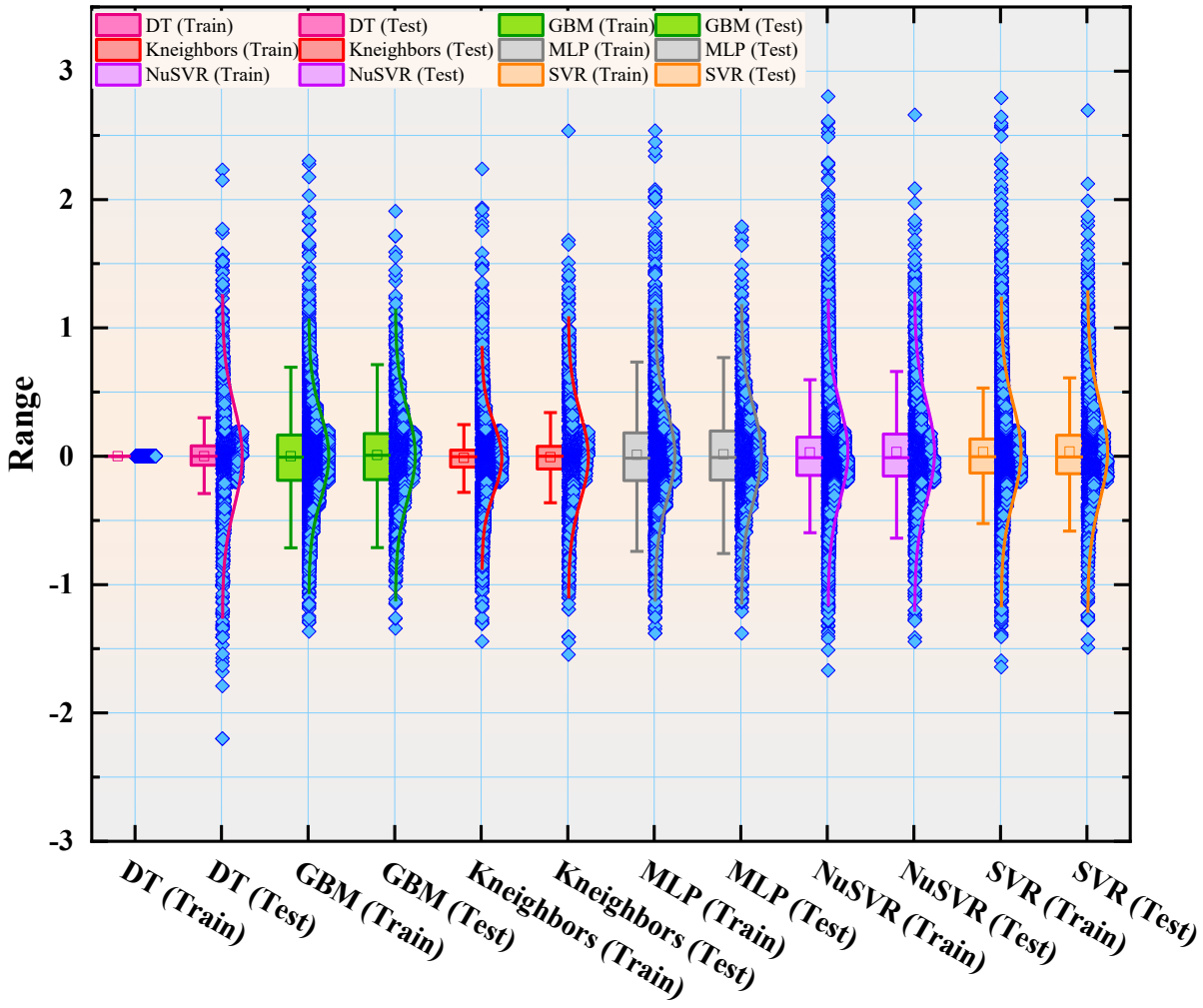


Figure 15: Boxplot comparison of prediction error range for training and testing sets across models

Fig. 16 is a comparative bar chart of several important performance measures: MDAPE, VAF, R^2, and STD for six models based on training and testing sets. DT and MLP models show high R^2 and low MDAPE when training but undergo strong declines on the test set, indicative of overfitting. KNN and GBM plots showed balanced R^2 and VAF between training and testing and very low MDAPE and STD _dev values, hence with more robust generalization and relatively equal precision within sets. Nu SVR and SVR models showed considerably lower R^2 and VAF values, hence capturing the pattern underlying the data effectively. On generalizing and keeping accuracy, overall, the most reliable models are KNN and GBM; DT and MLP train well but unfortunately considerably lose their reliability on unseen data, while SVR-based models are the weakest in this lot.

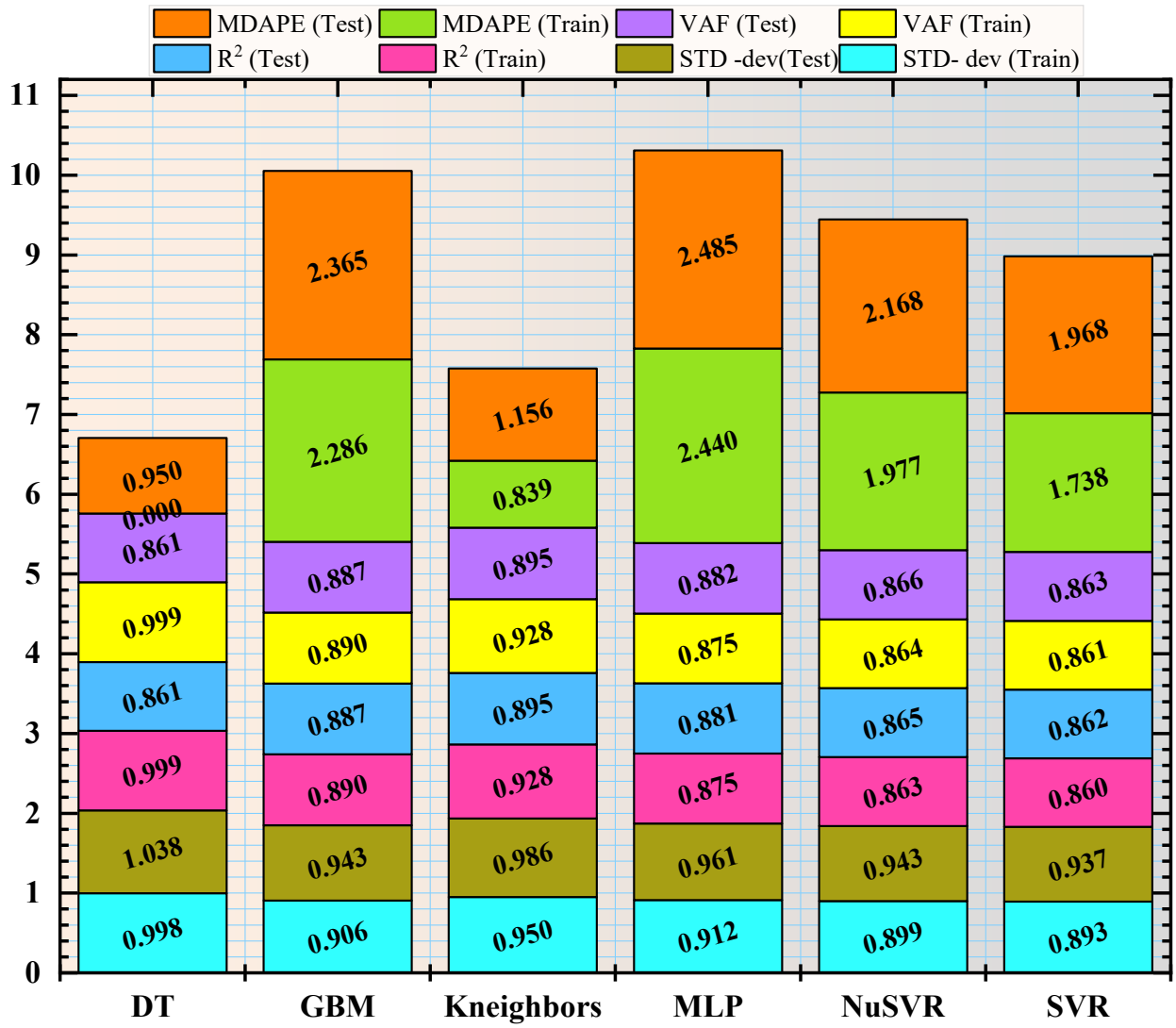


Figure 16: Comparative evaluation of model performance metrics across training and testing sets

Fig. 17 is a performance comparison of six models, namely DT, GBM, KNN, MLP, Nu SVR, and SVR, on four error metrics MAE-Mean Absolute Error, NMSE, MSE, and RMSE for both the testing and training datasets in a stacked bar chart. The DT model gives the minimum errors of both training and testing with very low RMSE and MAE values, indicating high accuracy. The KNN and GBM models also result in relatively balanced training and test set errors, thus indicating good generalization capability. On the other hand, Nu SVR and SVR models rank among those with the highest value of RMSE and MAE, especially for the test set, reflecting weak predictive accuracy with not-so-effective generalization capability. MLP performs moderately, with relatively higher errors within the test set than in the training set, but still outperforms the models based on SVR. Overall, DT, KNN, and GBM are the most accurate and reliable models, while Nu SVR and SVR present relatively high errors, which may thus be less appropriate for this data set.

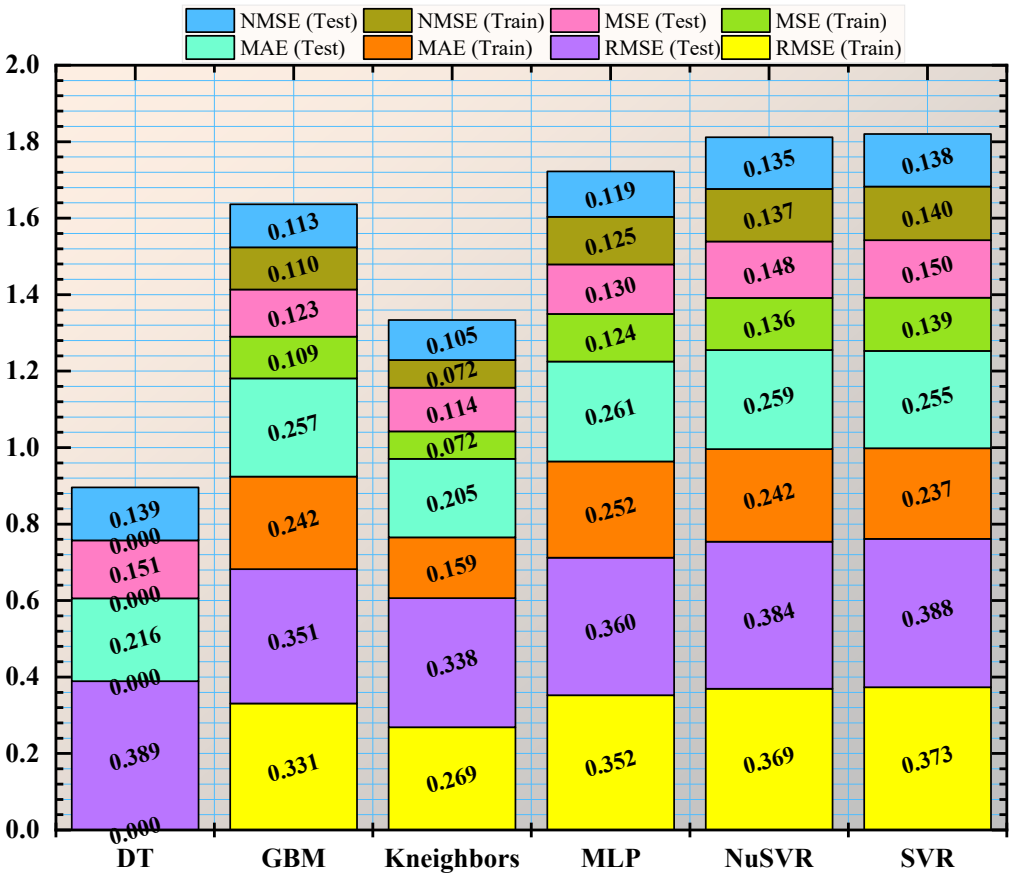


Figure 17: Comparative analysis of error metrics across models for training and testing sets

Fig. 18 is a Taylor diagram, which shows the comparison of the different models' performance visually by plotting their performance regarding the STD, correlation coefficient, and centered RMSE for the WS variable against the observed data point (yellow star). The ideal model should align with the observed data point at the correlation coefficient of 1.0, the STD of 1.0, and the smallest RMSE. The models that are closest to the observed point, such as the Voting and Stacking ensembles represented by triangles, have highly correlated and low STD, meaning they are highly predictive with reliable performance. Other models, such as DT and GBM, have high correlation coefficients but slightly higher STD, hence being relatively accurate yet with more variance. This places the SVR and NuSVR models farther away from this observed point, indicating a weaker correlation and higher RMSE values to show poorer prediction accuracy. In general, what this diagram pinpoints is that the two ensemble models, Voting and Stacking, provide the best balance of correlation and minimization of error as the most robust choices for WS predictions on this dataset.

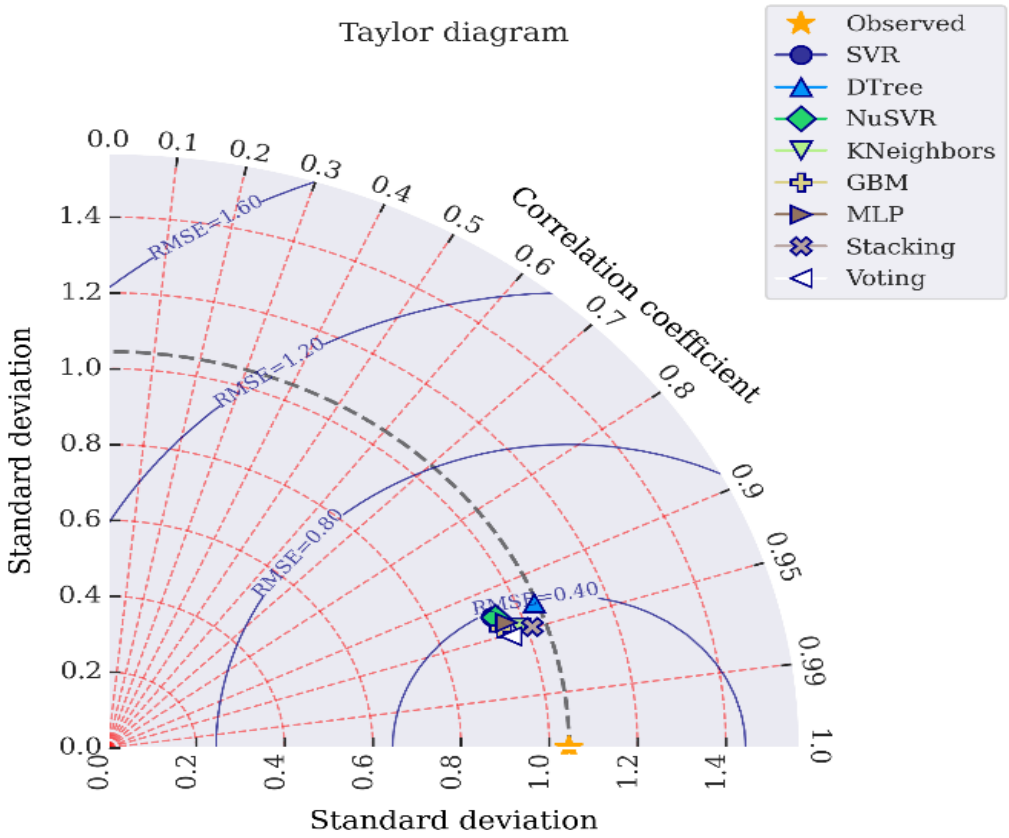


Figure 18: Taylor diagram of model performance in WS prediction

Table 1 presents the performance of two ensemble models, Stacking and Voting, based on a number of performance metrics, both for the testing and training sets. The results of the Stacking model outperform those obtained by the Voting model in the training set: the former has a lower MSE. These lower error values, in addition to the higher value of R^2, which is 0.951 compared with 0.931, suggesting that stacking fitting is more accurate for training data. Furthermore, lower NMSE and MDAPE for Stacking mean that the model is more accurate.

On the test set, both models behave similarly, with the Voting model only marginally outperforming Stacking on some metrics. This is in reaching the minimum MSE of 0.105 against 0.111 and the RMSE of 0.324 against 0.333. Also, the R^2 is marginally higher for the Voting model, 0.904, than for Stacking, 0.898, and so is VAF. This would then indicate that it generalizes slightly better to unseen data. However, both models have a similar consistency, with the STD for the two data sets spreading relatively comparably, as can be seen from the values that the STD has taken. Overall, Stacking generalizes slightly better on the training set data, while Voting creates a slight advantage in terms of performance on the test data and is, therefore, more reliable for generalization. From this, a clear idea of the strength of ensemble techniques in making robust and correct predictions has emerged, with Voting being slightly preferable in this context.

Table 1: Comparative performance analysis of stacking and voting ensemble models

Optimizer	Stacking	Voting
Train		
MSE	0.049	0.069
RMSE	0.221	0.263
MAE	0.130	0.176
R2	0.951	0.931
NMSE	0.049	0.069
MDAPE	0.838	1.478
STD _dev	0.968	0.918
VAF	0.951	0.931
Test		
MSE	0.111	0.105
RMSE	0.333	0.324
MAE	0.206	0.219

R2	0.898	0.904
NMSE	0.102	0.096
MDAPE	1.356	1.778
STD _dev	1.012	0.959
VAF	0.898	0.904

Table 2 compares the error metrics for various ML models, such as DT, GBM, KNN, MLP, Nu SVR, and SVR, in both training and test datasets. The main performance metrics like MSE, RMSE, MAE, and R^2 all suggest that KNN is doing the best on the Test set generally, with the lowest MSE and RMSE along with the highest R^2, which can indicate better generalization on previously unseen data. DT, by contrast, has extremely low errors on the training set, which may indicate overfitting, as its test performance falls comparatively. On the whole, the KNN and GBM models have a very good balance regarding stability and accuracy for both testing and training; hence, these models can be used to make reliable predictions on new data.

Table 2: Error metrics derived from the application of hybrid models

Optimizer	DT	GBM	KNN	MLP	Nu SVR	SVR
	Train					
MSE	1.97E-33	0.109	0.072	0.124	0.136	0.139
RMSE	4.44E-17	0.331	0.269	0.352	0.369	0.373
MAE	2.22E-18	0.242	0.159	0.252	0.242	0.237
R2	0.999	0.890	0.928	0.875	0.863	0.860
NMSE	1.98E-33	0.110	0.072	0.125	0.137	0.140
MDAPE	0	2.286	0.839	2.440	1.977	1.738
STD _dev	0.998	0.906	0.950	0.912	0.899	0.893
VAF	0.999	0.890	0.928	0.875	0.864	0.861
Test						
MSE	0.151	0.123	0.114	0.130	0.148	0.150
RMSE	0.389	0.351	0.338	0.360	0.384	0.388
MAE	0.216	0.257	0.205	0.261	0.259	0.255
R2	0.861	0.887	0.895	0.881	0.865	0.862
NMSE	0.139	0.113	0.105	0.119	0.135	0.138
MDAPE	0.9501	2.365	1.156	2.485	2.168	1.968
STD _dev	1.038	0.943	0.986	0.961	0.943	0.937
VAF	0.861	0.887	0.895	0.882	0.866	0.863

Fig. 19 is Actual vs. predicted scatter plots for Voting and Stacking ensemble model predictions. Markers distinguish between training and testing points. Both the models show a high correspondence with the Y=X line, suggesting that the models have resulted in good predictions. Similarly, the R^2 value of the Voting model is 0.931 for the training data and 0.904 for the test data, indicating very good predictive accuracy and generalization. On the other hand, the Stacking model is relatively a bit higher at 0.951 for the training data, which means that it has fitted better for the training data, whereas the generalization has faced a small drop, as evidenced by its R^2 of 0.898 in the test data. This pattern would hint that while Stacking may capture more detail within the training data, and Voting maintains slightly better consistency across both training and testing datasets. Overall, both ensemble methods yield reliable predictions. Still, voting is slightly more robust and, thus, perhaps better suited for situations where the generalization of new data is paramount.

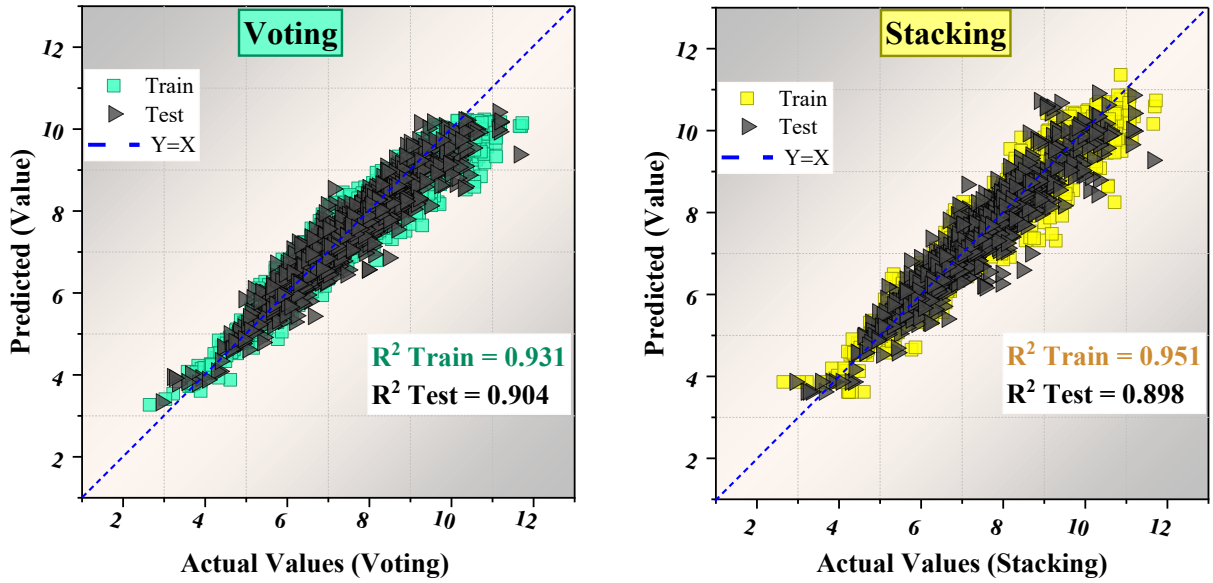


Figure 19: Comparing projected and actual values for voting and stacking ensemble models

The error distribution of the ensemble models for voting and stacking is displayed in Fig. 20. The errors are concentrated around zero, indicating the quite accurate predictions made by both models. The green color bars represent the error distribution for the Voting model, and the yellow color bars represent the error distribution for the Stacking model, which have a similar central peak, reflecting that most of the predictions are quite close to the actual values. This tells us that both models tend to keep their prediction errors low without strong bias, and the spread of both curves is quite similar. The model has overall error distributions being much larger or much smaller in comparison to the competing model. However, the Voting model seems to be somewhat more centered or somewhat narrower in distribution compared to the Stacking model's distribution, though it may indicate that Voting produces more consistent predictions with fewer extreme deviations. This would be a reason for the following conclusion: both ensemble methods yield high accuracy, although Voting has a minor advantage in terms of error consistency.

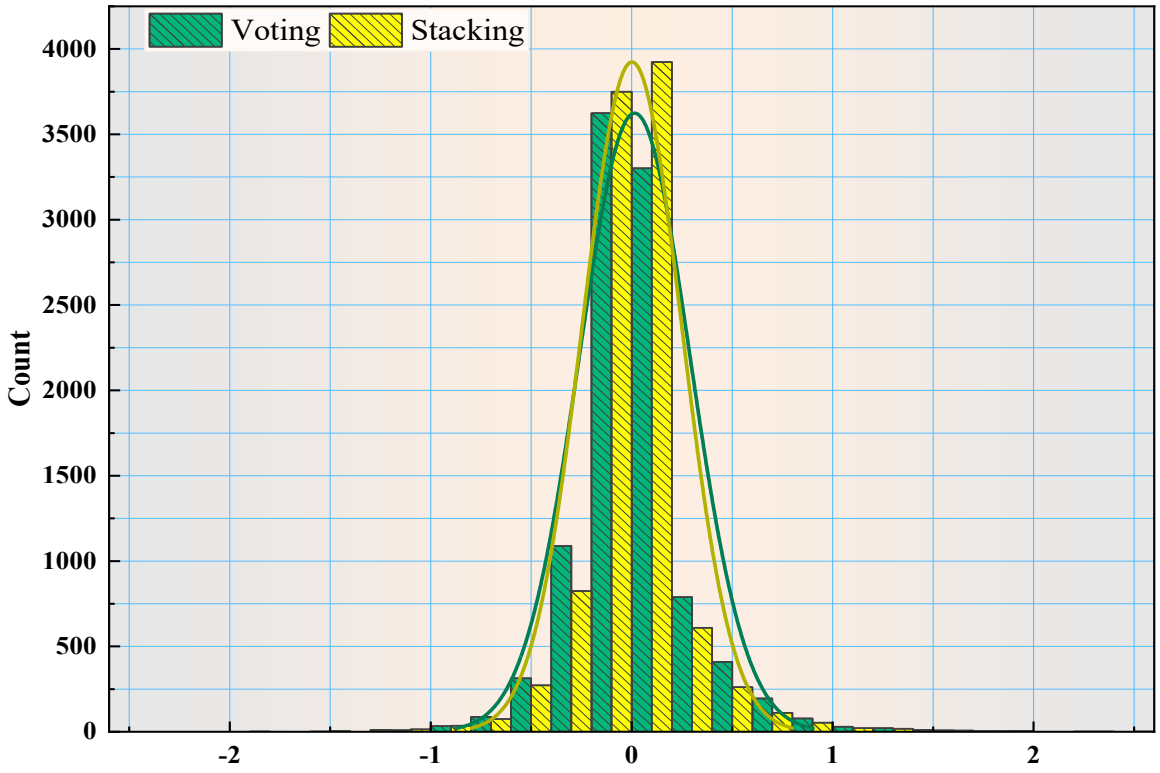


Figure 20: Error distribution for voting and stacking ensemble models

Fig. 21 is a SHAP analysis of the dependence of longitudinal features, latitude, capacity, and capacity factor on WS forecast, with different strengths. The most influencing features are the longitude and capacity factor. Longitude has a nonlinear but huge influence in some ranges of this variable, while the capacity factor has shown quite a clear positive linear relationship with WS predictions-higher predicted WSs at locations with higher capacity factors. Latitude holds variability in its influence from one region to another, whereas capacity is more scattered, which would argue for a less direct relationship with WS. This overall analysis suggests that longitude and capacity factors seem to be strong predictors of wind velocity, whereas latitude and capacity are more profoundly interconnected.

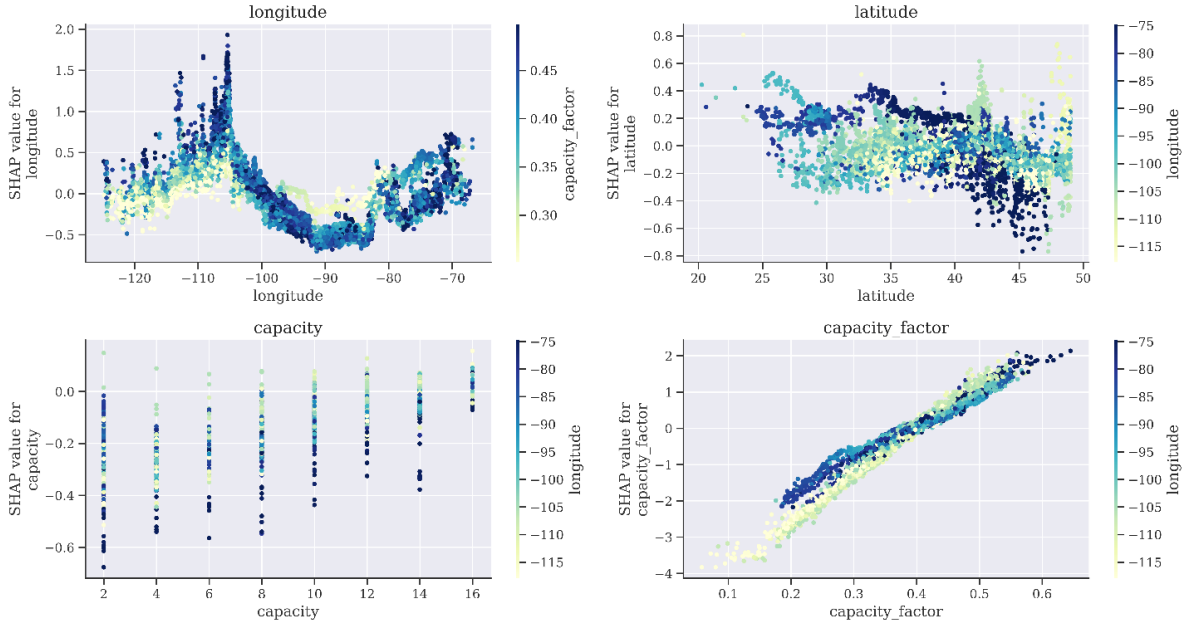


Figure 21: SHAP analysis of feature influence on WS prediction

Fig. 22 represents a summary plot by SHAP, ranking the influence of four features: capacity factor, longitude, latitude, and capacity towards the WS prediction model. The horizontal position of each point depicts the SHAP value, correlating with the impact that each feature had on the model output, where positive values push predictions higher and negative ones push them lower. The most informative feature is the capacity factor, which has a large spread of SHAP values and higher values on the right of the distribution, showing that larger magnitudes of the capacity factor indeed strongly raise the predicted WS. The second most impactful feature value is longitude, which reveals partly positive SHAP values at both the lower and upper bounds of its respective range, suggesting that some special values in longitude positively or negatively influence predictions based on geographical location. Latitude and capacity are relatively smaller in their effects, as indicated by centering their SHAP values around zero, hence their limited influence on the model output. Color gradient represents the feature's actual value: high values appear in dark blue, and low values appear in light yellow. This plot underlines the fact that capacity factor and longitude are the most influential drivers in the prediction of WS, while latitude and capacity are less part of the model's predictions.

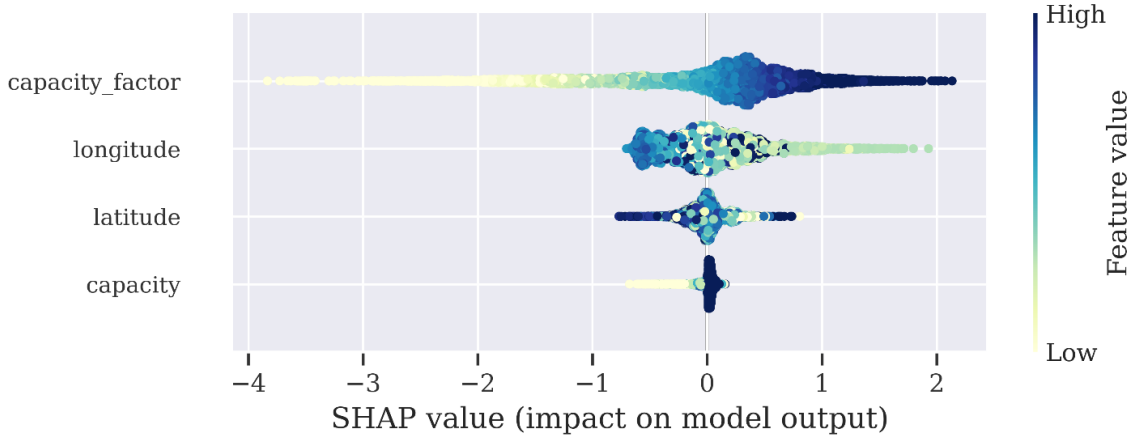


Figure 22: SHAP summary plot of feature impact on WS prediction model

This was a comparison analysis between ML models and ensemble techniques for WS prediction. According to the metrics of R^2, MSE, and RMSE, all showed that the ensembles-the Voting and Stacking methods-are the most robust and accurate predictions. The aforementioned methods work best in capturing the complex patterns in wind data since there is a combination of many models and better generalization on unseen test data. Voting slightly outperformed others for test data, balancing the accuracy with the stability of the results. Besides, SHAP analysis has shown that capacity factor and geographic location longitude are the strongest predictors of WS; therefore, both environmental factors and turbine-specific parameters are critical in determining the output of wind energy. Overall, these results suggest that the application of ensemble techniques with the selected features, such as capacity factor and longitude, will significantly improve the accuracy of the WS forecast, consequently allowing for better and more effective renewable energy prediction. This information will be of paramount importance in further work on wind power integration into the grid for a more sustainable energy infrastructure.

5 Conclusion

It shows the precision of neural network-based and ML models, particularly ensemble methods, in forecasting WS for data from the NREL Wind Integration National Dataset. The research underlines that new, sophisticated ensemble approaches like Voting and Stacking can enable more reliable and precise WS forecasting than basic, classic, stand-alone models. In these advanced ensemble methods, the strengths of multiple algorithms are combined to achieve improved predictive performance. According to the performance metrics of the model-MSE, RMSE, and the R^2, it was observed that the ensemble models capture non-linear relationships in wind data and generalize better to new, unseen data. The SHAP analysis also highlighted the most informative features for the prediction of WS: capacity factor and geographic location (longitude) and, therefore, pointed to the importance of those variables in renewable energy forecasting. This is a very important insight that can be used to optimize the production of wind energy with a view to integrating renewable energy into the grid in such a way as to ensure operational efficiency and grid stability. These findings suggest that a combined approach using both neural networks and ensemble techniques provides one robust solution to handle the intrinsic variability of wind patterns, hence contributing toward sustainable energy management and planning. Future studies will be focused on more sophisticated DL models like RNN and LSTM, which are architecture designs for temporal dependencies in sequences that could yield even better forecasting performance of WS variability across horizons. Besides, the inclusion of more diversified meteorological and atmospheric variables can lead to further refinements of the models to help check problems associated with the dynamic and intermittent properties of wind. This, therefore, forms the basis of continued improvement in renewable energy prediction analytics, improving the development of resilient and sustainable energy systems.

Authorship contribution statement

Tao Zou: Writing-Original draft preparation, Conceptualization, Supervision, Project administration.
Xinwei Xie: Methodology, Software

Data availability

On Request

Author statement

The manuscript has been read and approved by all the authors, the requirements for authorship, as stated earlier in this document, have been met, and each author believes that the manuscript represents honest work.

Ethical approval

All authors have been personally and actively involved in substantial work leading to the paper, and will take public responsibility for its content.

References

[1] M. L. Hossain, S. M. N. Shams, and S. M. Ullah, “Comparative study of machine learning algorithms for wind speed prediction in Dhaka, Bangladesh,” Sustainable Energy Research, 11(1):23, 2024. https://doi.org/10.1186/s40807-024-00109-z

[2] C. Draxl, A. Clifton, B.-M. Hodge, and J. McCaa, “The wind integration national dataset (wind) toolkit,” Appl Energy, 151, 355–366, 2015. https://doi.org/10.1016/j.apenergy.2015.03.121

[3] Y. Ren, P. N. Suganthan, and N. Srikanth, “Ensemble methods for wind and solar power forecasting—A state-of-the-art review,” Renewable and Sustainable Energy Reviews, 50, 82–91, 2015. https://doi.org/10.1016/j.rser.2015.04.081

[4] Y. Hao and C. Tian, “A novel two-stage forecasting model based on error factor and ensemble method for multi-step wind power forecasting,” Appl Energy, 238, 368–383, 2019. https://doi.org/10.1016/j.apenergy.2019.01.063

[5] M. Liu, Z. Cao, J. Zhang, L. Wang, C. Huang, and X. Luo, “Short-term wind speed forecasting based on the Jaya-SVM model,” International Journal of Electrical Power & Energy Systems, 121, 106056, 2020. https://doi.org/10.1016/j.ijepes.2020.106056

[6] G. Maclaurin, C. Draxl, B.-M. Hodge, and M. Rossol, “Wind Integration National Dataset (WIND) Toolkit,” DOE Open Energy Data Initiative (OEDI); National Renewable Energy Laboratory …, 2014. https://doi.org/10.1016/j.apenergy.2015.03.121

[7] T. Blanchard and B. Samanta, “Wind speed forecasting using neural networks,” Wind Engineering44(1): 33–48, 2020. https://doi.org/10.1177/0309524X19849846

[8] J. Nielson, K. Bhaganagar, R. Meka, and A. Alaeddini, “Using atmospheric inputs for Artificial

Neural Networks to improve wind turbine power prediction," Energy, 190, 116273, 2020. https://doi.org/10.1016/j.energy.2019.116273

[9] M. M. Rahman et al., "A comprehensive study and performance analysis of deep neural network-based approaches in wind time-series forecasting," J Reliab Intell Environ, 9(2): 183–200, 2023. https://doi.org/10.1007/s40860-021-00166-x

[10] A. Dolatabadi, H. Abdeltawab, and Y. A.-R. I. Mohamed, "Hybrid deep learning-based model for wind speed forecasting based on DWPT and bidirectional LSTM network," IEEE Access, 8, 229219–229232, 2020.DOI: 10.1109/ACCESS.2020.3047077

[11] M. Lydia, G. Edwin Prem Kumar, and R. Akash, "Wind speed and wind power forecasting models," Energy & Environment, p. 0958305X241228515, 2024. https://doi.org/10.1177/0958305X241228515

[12] A. Routray, Y. S. Reddy, and S. Hur, "Predictive Control of a Wind Turbine Based on Neural Network-Based Wind Speed Estimation," Sustainability, 15(12): 9697, 2023. https://doi.org/10.3390/su15129697

[13] H. Malik, A. K. Yadav, F. P. G. Márquez, and J. M. Pinar-Pérez, "Novel application of Relief Algorithm in cascaded artificial neural network to predict wind speed for wind power resource assessment in India," Energy Strategy Reviews, 41, 100864, 2022. https://doi.org/10.1016/j.esr.2022.100864

[14] K. Wang, O. Gaidai, F. Wang, X. Xu, T. Zhang, and H. Deng, "Artificial neural network-based prediction of the extreme response of floating offshore wind turbines under operating conditions," J Mar Sci Eng, 11(9): 1807, 2023. https://doi.org/10.3390/jmse11091807

[15] I. Ahmad, F. M'zoughi, P. Aboutalebi, I. Garrido, and A. J. Garrido, "Fuzzy logic control of an artificial neural network-based floating offshore wind turbine model integrated with four oscillating water columns," Ocean Engineering, 269, 113578, 2023.https://doi.org/10.1016/j.oceaneng.2022.113578

[16] A. M. Assaf, H. Haron, H. N. Abdull Hamed, F. A. Ghaleb, S. N. Qasem, and A. M. Albarrak, "A review on neural network-based models for short term solar irradiance forecasting," Applied Sciences, 13(14): 8332, 2023. https://doi.org/10.3390/app13148332

[17] M. Jiang, J. Shen, X. Jiang, L. Luo, R. Zhou, and Q. Zhou, "A novel automatic wind power prediction framework based on multi-time scale and temporal attention mechanisms," arXiv preprint arXiv:2302.01222, 2023. https://doi.org/10.48550/arXiv.2302.01222

[18] H. Liu, H. Ma, and T. Hu, "Enhancing short-term wind speed forecasting using graph attention and frequency-enhanced mechanisms," arXiv preprint arXiv:2305.11526, 2023. https://doi.org/10.48550/arXiv.2305.11526

[19] H.-K. Wang, K. Song, and Y. Cheng, "A hybrid forecasting model based on CNN and informer for short-term wind power," Front Energy Res, 9, 788320, 2022. https://doi.org/10.3389/fenrg.2021.788320

[20] S. Huang, C. Yan, and Y. Qu, "Deep learning model-transformer based wind power forecasting approach," Front Energy Res, 10, 1055683, 2023. https://doi.org/10.3389/fenrg.2022.1055683

[21] M. M. Christian, Y. S. Kim, H. Choi, J. Lee, and S. You, "Enhancing Wind Speed and Wind Power Forecasting Using Shape-Wise Feature Engineering: A Novel Approach for Improved Accuracy and Robustness," arXiv preprint arXiv:2401.08233, 2024.https://doi.org/10.17703/IJACT.2023.11.4.393

[22] J. Keisler and E. Le Naour, "WindDragon: Enhancing wind power forecasting with Automated Deep Learning," arXiv preprint arXiv:2402.14385, 2024.https://doi.org/10.48550/arXiv.2402.14385

Audio Steganography for Healthcare Data Using 24x8 Compression and Bit Comparison Substitution Algorithm

K Revathi, and S. Kaja Mohideen*
Department of Electronics and Communication Engineering, B.S.Abdur Rahman Crescent Institute of Science and Technology, Vandalur, Chennai, Tamil Nadu, India
E-mail: revathivigneswaranphd@gmail.com, kajamohideen@crescent.education.
*Corresponding Author

Keywords: Audio steganography, least significant bit algorithm (LSB), run length encoding (RLE) compression, text files, encryption, decryption, waveform audio file format

Received: July 3, 2025

Audio steganography embeds multimedia files within audio files. Traditional audio steganography used the Least Significant Bit algorithm, which required eight samples to embed one byte of a text file. This study reduces the utilization of audio samples by compressing healthcare data from three characters to one using the proposed 24×8 compression algorithm. Three audio files were used to transmit the probability distributions, encoded values, and index values. The encoded values are embedded using the Bit Comparison and Substitution-3 algorithm, with a three-bit difference from audio samples. It is retrieved using the Bit Comparison and Retrieval-3 algorithm and decompressed with the 8×24 decompression algorithm. For enhanced security, healthcare data was encrypted using the Incremental Order Value Algorithm and decrypted with the Decremental Reverse Order Value Algorithm. The least significant bit algorithm embeds the probability distributions and index values with a secret key. Audio files from Mixkit and healthcare data from the COVID Dialogue Dataset were used for evaluation. The proposed algorithms achieved an average throughput of 8592.74 KB/s, surpassing the 3-DES algorithm due to the incremental shift in ASCII values within healthcare data. A compression ratio of 3:1 was achieved by compressing 3 bytes of data to 1, outperforming Huffman and LZW. The embedding algorithm achieved a PSNR of 42.1480 dB and a BER of 7.0165×10^{-5}, demonstrating improved efficiency with reduced audio samples in embedding compared to the traditional LSB algorithm. 15000 bytes of healthcare data were embedded into 5000 audio samples, resulting in 15000-bit differences between the cover and stego audio files.

Povzetek: Študija predlaga zvočno steganografijo za zdravstvene podatke, ki z lastnim 24×8 stiskanjem in (šifriranim) vgrajevanjem prek bitne primerjave/substitucije zmanjša porabo vzorcev in ob ohranjeni kakovosti zvoka omogoči učinkovitejše skrivanje kot klasični LSB.

1 Introduction

The COVID-19 pandemic, combined with advanced technology, has highlighted the significance of telemedicine, enabling patients to consult medical experts wirelessly. Rising healthcare expenses have also driven the need for telemedicine, increasing communication between doctors and between doctors and remote patients. Telemedicine services, such as video conferences and web-based visits, provide remote healthcare. This approach can make healthcare more efficient, centralized, and accessible.

With telemedicine, patients receive services for distant healthcare, critical situations, medication management, and chronic health conditions. It also lowers the risk of hospital infections for patients with weak immune systems. Additionally, several resources, such as self-examinations, step-by-step training programs customized for their conditions, and email-based symptom descriptions, help patients communicate their symptoms to healthcare providers. Monitoring apps and smartphones assist in the care of patients with chronic illnesses. Despite its benefits, telemedicine poses risks to the security of healthcare data when using public networks. Hence, protecting healthcare data is a primary concern. Cryptography and steganography are information-hiding techniques for secure communication of healthcare data. These information-hiding techniques, along with IoT devices and artificial intelligence (AI), enable the successful transmission of healthcare data [1], [2], [3].

This research focused on combining cryptography and steganography to ensure double-layered security for communicating healthcare data.

Cryptography converts plaintext into ciphertext using encryption keys. The plaintext in this research denotes unique healthcare data (HCD) transmitted to the receiver, and the ciphertext represents its unreadable form.

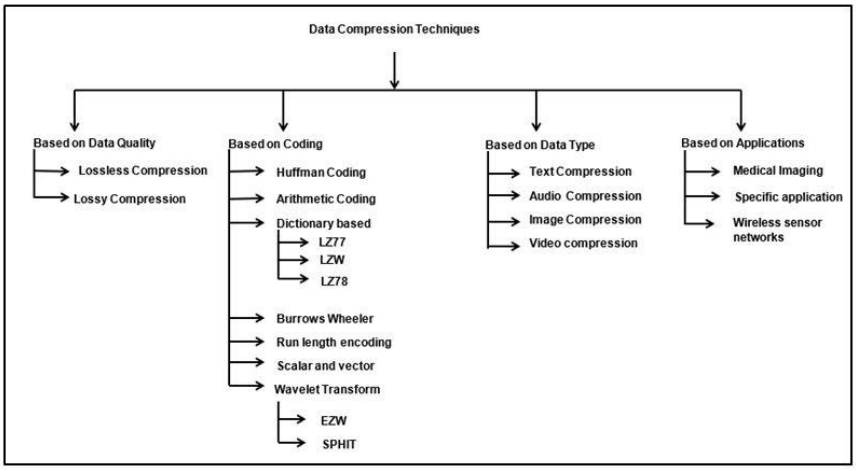


Figure 1: Data compression techniques

Various encryption algorithms and keys convert the medical data into ciphertext. The encryption algorithms include the Advanced Encryption Standard (AES), Data Encryption Standard (DES), Triple DES (3DES), and the International Data Encryption Algorithm (IDEA). At the receiver's end, the decryption algorithm and keys decrypt the ciphertext to plain text. These algorithms are selected based on capacity, security, and reliability [4], [5], [6].

In the proposed work, healthcare data is encrypted into a ciphertext using the Incremental Order Value Algorithm (I-OVA). The Decremental Reverse Order Value Algorithm (D-RVA) decrypts the unreadable form of healthcare data.

Cryptography provides the first layer of security by transforming plain healthcare data into ciphertext.

The second layer of healthcare data security is steganography, which conceals the existence of HCD during communication. Steganography embeds multimedia files within other multimedia files that serve as carriers. These files include audio, video, image, and text formats. Audio carriers are highly efficient due to the human auditory system (HAS) being less sensitive to slight variations in sound than the human visual system (HVS). Audio steganography specifically embeds multimedia files in audio carriers, characterized by high capacity, robustness, and imperceptibility. The maximum amount of data hidden in a cover audio file without affecting its overall quality is known as capacity. Imperceptibility refers to the ability to alter an audio file without compromising its stego audio quality. Robustness is the strength of steganographic communication and data retrieval from the stego audio file. There is a trade-off between these characteristics of audio steganography. Increased embedding capacity often decreases robustness and imperceptibility. The proposed work addresses this trade-off by compressing the healthcare data before embedding it in audio files [7], [8], [9].

Compression is the process of reducing the data size from its original size. The main objectives of multimedia data compression are to reduce transmission time, network bandwidth, and storage volume.

Various compression techniques apply to text, images, videos, and other digital data types. According to the literature, data compression involves encoding, while data decompression is decoding. Decompression restores the compressed data to its original form [10], [11]. Data compression can be classified into two types: lossy and lossless compression. Figure 1 shows the various data compression techniques [11]. Lossless compression is applied when the accuracy of the original data is essential. In lossy compression, data portions are removed by preserving the potential of compression [12]. This research combines double-layered security with lossless compression of encrypted healthcare data using the 24x8 compression algorithm. The Run Length Encoding (RLE) algorithm compresses the probability distribution, a simple form of lossless data compression, and saves data sequences with the same value into a data file.

A comprehensive review of related works details the literature on telemedicine, information-hiding techniques such as cryptography and its algorithms, and audio steganography with its characteristics. The proposed work describes the embedding and extraction algorithms, which are subsequently analyzed and evaluated in the results and discussion section. Theoretical analysis examines these algorithms in depth. Finally, the paper concludes in the conclusion and future scope section.

1.1 Problem statements

The literature survey identifies a significant research gap in securely communicating healthcare data (HCD) using audio steganography in telemedicine.

1. Telemedicine is an emerging area of research with compromised data security and efficiency.
2. Ensuring the security of healthcare data (HCD) in remote consultations is essential.
3. Currently, telemedicine practices image steganographic communication over audio steganography, leaving the latter underexplored despite its potential.
4. Most conventional audio steganography methods, including those based on LSB, do not incorporate compression before embedding, as referred to in Table 1, leading to inefficient use of audio capacity.
5. Existing compression methods result in variable-length encoded data and require transmission of

codebooks or dictionaries for reconstruction. This results in increased overhead and limits their suitability for fixed-size, low-latency audio embedding of compressed data. A fixed-length compression scheme with minimal computational and embedding overhead is required to enhance efficiency in audio steganographic systems.

6. The LSB algorithm has an embedding rate of 12.5%, with 1 byte of a text file embedded into eight audio samples of 1 byte. This introduces a trade-off between embedding capacity and audio sample utilization. Increasing the embedding capacity requires additional audio samples, reducing the available samples for further embedding.
7. In the LSB algorithm, the error bits vary inconsistently with the bit pattern of the audio samples and the embedded text, especially as the embedding capacity increases. This inconsistency limits its effectiveness. There is no consistent relationship between error bits, the embedded text file, and the audio samples.

In response to these challenges, this work proposes effective embedding and extraction algorithms that combine cryptography and steganography for secure communication of compressed healthcare data in telemedicine.

1. I-OVA and D-RVA: The Incremental Order Value algorithm (I-OVA) and Decremental Reverse Order Value algorithm (D-RVA) to encrypt and decrypt healthcare data, ensuring data security.
2. Merging algorithm (MA): Three 8-bit characters are combined into a 24-bit sequence, optimizing data representation.
3. The probability distribution (PD) compression: RLE compresses redundant probability distribution data using value-count pairs, eliminating redundancy, reducing data size, and enabling efficient embedding with minimal retrieval overhead.
4. 24x8 compression algorithm: Cipher Healthcare data (C-HCD) of 24 bits is compressed into an 8-bit encoded value to optimize embedding and is subsequently reconstructed using the 8×24 decompression algorithm.
5. BCS-3 and BCR-3: The Bit Comparison and Substitution-3 (BCS-3) and Bit Comparison and Retrieval-3 (BCR-3) algorithms embed and retrieve the encoded values in Audio File 2 (AF2), ensuring secure embedding with efficient utilization of audio samples.
6. The LSB algorithm embeds and extracts the RLE count and its values for probability distribution in Audio File 1(AF1) and the Index value of Stego Audio Samples (SAF2) in Audio File 3(AF3) with secret key Ks as a reference.

This study investigates the secure communication of compressed healthcare data using double-layer security by integrating cryptographic encryption with efficient steganographic embedding. The proposed design uses a 24×8 compression algorithm, BCS-3 embedding, and cryptographic encryption to reduce healthcare data size and the number of audio samples needed for embedding.

It is hypothesized that this integration ensures consistent error control and accurate data recovery while minimizing retrieval overhead. Simulations show that 15,000 bytes of data were compressed to 5,000 encoded bytes (1 byte each), improving the compression ratio and optimizing audio sample utilization.

The framework of the proposed method is shown in Figure 2, which is more self-explanatory. Table 1 summarizes the related works.

2 Related works

A detailed study on audio steganography discussed techniques such as linear and sequential methods, selective embedding, frequency masking and thresholding, error minimization-based embedding, pattern matching, phase coding, and spread spectrum methods [13]. The merits, demerits, and applications of audio steganography in medicine, transportation, and government domains were discussed. Additionally, they highlighted the need to explore a broader range of steganography methods for transmitting healthcare data. It was also noted that research should use various cover media like audio, video, and text, as 7 out of 9 articles utilized image steganography [14].

The traditional LSB algorithm for audio steganography is enhanced to increase embedding capacity and security. A robust hybrid steganographic system was proposed in [15], where text messages are initially encrypted using bit cycling operations and embedded in randomly selected bits of an audio file.

However, increased embedding capacity also raises the risk of steganalysis. So, advanced symmetric and asymmetric cryptography algorithms like AES, DES, 3DES, Blowfish, IDEA, RSA, ECC, and ElGamal are used to strengthen security [16].

Recent research focuses on providing double-layer security to embedded data by combining cryptography with steganography. The study [17] addresses the critical challenge of securely transmitting sensitive medical data, such as COVID-19-related information, with a novel crypto-steganography-based scheme, incorporating hybrid cryptography for encryption, random block, and pixel selection for enhanced security, and an inversion method for efficient data embedding.

The paper [18] studied the security of the modern steganography method using a lossless compression called arithmetic coding. Integrating compression techniques in audio steganography enhances embedding capacity and security.

Various compression algorithms, such as run-length encoding, Huffman Coding, and arithmetic coding, highlight their efficiency, merits, demerits, and corresponding applications. They also explored different types of data compression, including text, audio, video,

Table 1: Technical overview of methods and results [25]

Reference No.	Methodology	Inferences / Results
[26]	Steganography: Audio Carrier Medium: Audio Concealed Data: Text Steganographic Technique: Two LSB modification Cryptographic Technique: RSA.	Embedding Efficiency (Sample-to-Byte Ratio): 4:1 Payload:107 bytes MSE = 9.08875 Peak Signal to Noise (PSNR) = 34.2775dB Bit Error Rate (BER) =17.995%.
[27]	Steganography: Audio Carrier Medium : Audio Concealed Data: Text Steganographic Technique: The Fourth and Fifth LSBs of audio samples are embedded with encrypted data.	A PSNR of 17.71916498 dB was achieved after embedding the message into a JAZZ audio file. Payload:103 bytes Embedding Efficiency(Sample-to-Byte Ratio): 4:1
[28]	Steganography: Audio Carrier Medium : Audio Concealed Data: Text Steganographic Technique : Duel Encryption-LSB algorithm Cryptographic Technique : Duel Encryption-Pattern Matching Algorithm	A Mean Squared Error of 0.5619 resulted in Pattern Matching method in embedding target text of varying sizes, small, medium, and large in audio file cover1.wav and cover2.wav Payload: 250 bytes Normal LSB method : Sample-to-Byte Ratio: 8:1 Embedding rate =12.5%. MSE = 0.3994
[29]	Steganography: Audio Carrier Medium : Audio Concealed Data: Text Steganographic Technique: DWT for Embedding. Cryptographic Technique : Dynamic encryption algorithm	MSE of 0.8578 resulted in embedding Test5.txt in Two.wav audio file. Payload :9405 bytes.
[30]	Steganography: Audio Carrier Medium : Audio Concealed Data: Text Steganographic Technique :Random Key Indexing method (Trusted Third Party Stego key and Secondary Key generated at encoder) Cryptographic Technique: AES-256.	The right channel of a 16-bit stereo audio file resulted in a BER of 0.0187% after embedding 532 bytes of message data.
[31]	Steganography: Audio Carrier Medium : Audio Concealed Data: Audio Steganographic Technique: ECA-BM ,Fractal coding and chaotic LSB	The Normalized Correlation of the retrieved file is 0.99992. Hiding Capacity:80%
[32]	Steganography: Audio Carrier Medium : Audio Concealed Data: Audio Steganographic Technique : HASFC, Fractal coding and Chaotic least significant bit	Hiding Capacity:100% Histogram error rate (HER) = 0.1278 Difference Ratio (DR) of the fourth first moments: mean: 0.0799, variance: 0.0008, skewness: 0.0018, kurtosis: 0.0028
[33]	Steganography: Audio Carrier Medium : Audio Concealed Data: Image and Audio Steganographic Technique : Integer-to-Integer Lifting Wavelet Transform (Int2Int LWT) and Least Significant Bits (LSBs) substitution Cryptographic Technique: Adaptive steganography key (Stego key) for encryption.	Hiding Capacity:25% HER = 1.2274e-04 Difference Ratio (DR) of the first fourth moments: mean: 0.2304, variance: 0, skewness: 0.0004, kurtosis: 0.0005

and image. Text data is compressed using algorithms such as Burrows-Wheeler Transform and Boolean minimization, logical truth table, and graph-based method [11]. Authors [19] proposed compressive sensing and

audio steganography techniques to secure and compress medical images. Nathaniel Fout et al. highlighted the difficulty of compressing floating-point data compared to integer data. Lossless compression methods for scientific and medical floating-point volume data utilizing prediction-based techniques like the Adaptive Polynomial Encoder (APE) and Adaptive Combined Encoder (ACE) are proposed [20].

In medical image steganography, the medical image is decomposed using an integer wavelet filter, compressed with arithmetic coding, and encrypted with DES. The secret bits are then embedded in the LSB of high-frequency coefficients in the wavelet domain [21].

The pandemic situation has raised the need for telemedicine [22]. The authors [23] proposed a speech-to-speech workflow (STSW), a deep learning-based framework designed for multilingual telemedicine. Information-hiding techniques are an emerging area of research for the secure transmission of medical images for telemedicine [24]. This paper explores the integration of telemedicine with information-hiding techniques.

3 Proposed work

3.1 Embedding algorithm

3.1.1 Phase I: Incremental-order value algorithm (I-OVA)

Input: Healthcare data (HCD).
Output: Cipher healthcare data (C-HCD).

1. The 15000B HCD will be divided into sequential segments of 501 characters each, with the last segment containing the remaining characters.
2. Assign decimal Order Values (OV) ranging from 127 down to 32 to all printable ASCII characters (ASCII 32 to 127), such that OV 127 corresponds to ASCII 32, OV 126 corresponds to ASCII 33, and so on, forming a one-to-one mapping.
3. Assign an index value from 0 to 95 to each printable ASCII character (ASCII 32 to 127), representing its sequential position within the printable ASCII set. These indices serve as the numeric representation of the Cipher Healthcare Data (C-HCD).
4. Read the ASCII character codes for all printable characters in the HCD.
5. For each character in the HCD, treat its ASCII code as an Order Value (OV_input) and retrieve the corresponding OVA character using the OV-to-ASCII mapping (OV 127 → ASCII 32, OV 126 → ASCII 33, …, OV 32 → ASCII 127). This transformation produces the OVA-encrypted representation of the HCD.
6. Determine the ASCII codes of all OVA characters obtained in Step 5. These ASCII values serve as the basis for the subsequent incremental transformation in the I-OVA encryption process.
7. For each segment, increment the ASCII value of each OVA character by its position within the segment (e.g., first character by 1, second character by 2, and so on). If the incremented value exceeds 127, wrap around to 32 to ensure it remains within the printable ASCII range.
8. Convert the incremented ASCII values to their corresponding characters to obtain the I-OVA encrypted characters. Simultaneously, determine the index of each character within the printable ASCII set (Index 0–95) to represent the Cipher Healthcare Data (C-HCD).
9. Repeat Steps 5–8 for all characters in each segment to perform the complete I-OVA encryption of the Healthcare Data (HCD).
10. The indices of the incremented characters within the printable ASCII set (Index 0–95) represent the Cipher Healthcare Data (C-HCD).
11. End.

The metrics throughput and entropy verify the performance of the proposed I-OVA algorithm, as discussed in the Results and Discussion section.

3.1.2 Theoretical discussion of I-OVA

Table 2 presents the I-OVA to the HCD (id=1ht).
The first character of the HCD, ‘i’(ASCII 105), is its order value. The algorithm maps this order value to the character ‘6’(ASCII 54). Since ‘6’ is the first occurrence, its ASCII is incremented to 55, mapped to the character ‘7’ with index 23. Thus, ‘i’ is encrypted to ‘7’, and the same process applies to the other characters.

3.1.3 Phase II: Merging algorithm (MA)

Input: Cipher Healthcare data (C-HCD)
Output: Merged Cipher healthcare data(M-C-HCD) (24 bits)

1. Read the C-HCD in each segment and convert it into 8-bit binary streams.
2. Measure the length (L) of the C-HCD.
3. If L is divisible by three without a remainder, merge the three one-byte characters in the C-HCD to get the character of three bytes (24 bits).
4. If L is not divisible by 3, additional bytes of "00000000" are appended at the end of the C–HCD to ensure the remaining characters form complete sets of three for merging. If one byte remains, two bytes of "00000000" are added; if two bytes remain, one byte of "00000000" is added.
5. Repeat steps 3 or 4 to merge all the characters in each segment of C-HCD.
6. The merged cipher healthcare data (M-C-HCD) for 501-byte segments results in 167 × 24 bits.
7. End.

The simulation results of the merging algorithm are presented and discussed in Figure 11 in the Results and Discussion section

.

3.1.4 Theoretical discussion of MA

The output of the I-OVA algorithm, as shown in Table 2, is used as the input for the merging algorithm. The

decimal values of the I-OVA output are converted to 8-bit binary streams. Since six characters are processed, they are merged into groups of three characters per row, as shown in Table 3.

The 15000B HCD is divided into 30 segments, with the first 29 segments containing 501 characters each. The last segment holds the remaining 471 characters. Each segment is then arranged into three characters (24 bits) per row.

3.1.5 Phase III: 24x8 compression algorithm

Input: M-C-HCD
Output: Probabilities and Encoded value

Step 1: Initialization
Initialize an array named probabilities with a length of 16,777,216, setting all elements to zero.
Initialize index_array ← empty
Initialize encoded_binary_array ← empty
Step 2: Index Construction and Event Assignment
For each segment in M-C-HCD:
Convert the 24-bit binary sequence to its decimal equivalent: index ← bin2dec(segment)
Increment the decimal index by 1: index ← index + 1. In the probability distribution, represent the occurrence at this index as one (1): probabilities[index] ← 1
Append the index to the index_array

Step 3: Normalization
Compute the total sum of all values in the probabilities array: Total ← sum(probabilities)
Step 4: Encoded Value Calculation
For each index in index_array:
Compute the probability value: P ← probabilities[index] / Total
Compute the sum of probabilities up to the element preceding the index:
C ← sum(probabilities[1 to index - 1]) / Total
Calculate the encoded value:
encoded_value ← C + (P / 2)
Scale the encoded value to the 8-bit binary range:
scaled_value ← round(encoded_value × 255)
Convert the rounded value to 8-bit binary:
binary_8bit ← dec2bin(scaled_value, 8)
Store the binary representation in the encoded_binary_array
Step 5: Output Return probabilities, encoded_binary_array

The algorithm compresses 24-bit M-C-HCD to an 8-bit encoded value using the 24x8 compression algorithm. The proposed 24×8 compression algorithm encodes 15000B of HCD into 5000 values, with each row in every segment contributing to an encoded value.

As discussed in the Results and Discussion section, the efficiency of the proposed algorithm is evaluated using compression ratio and space savings.

Table 2: Encryption of HCD using I-OVA algorithm

S:NO	HC Data [step 1]	ASCII Code [step 4]	Order Value [step 5]	OVA Character [step 5]	ASCII Code [step 6]	I-OVA Value [step 7]	I-OVA Character [step 8]	Index: I-OVA Character (C-HCD) [step 3 & 8]
1	i	105	105	6	54	54+1=55	7	23
2	d	100	100	;	59	59+2=61	=	29
3	=	61	61	b	98	98+3=101	e	69
4	1	49	49	n	110	110+4=114	r	82
5	h	104	104	7	55	55+5=60	<	28
6	t	116	116	+	43	43+6=49	1	17

Table 3: Merging algorithm

C-HCD	Binary Value								Merged cipher healthcare (M-C-HCD) data																							
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
23	0	0	0	1	0	1	1	1																								
29	0	0	0	1	1	1	0	1	0	0	0	1	0	1	1	1	0	0	0	1	1	1	0	1	0	1	0	0	0	1	0	1
69	0	1	0	0	0	1	0	1																								
82	0	1	0	1	0	0	1	0																								
28	0	0	0	1	1	1	0	0	0	1	0	1	0	0	1	0	0	0	0	1	1	1	0	0	0	0	0	1	0	0	0	1
17	0	0	0	1	0	0	0	1																								

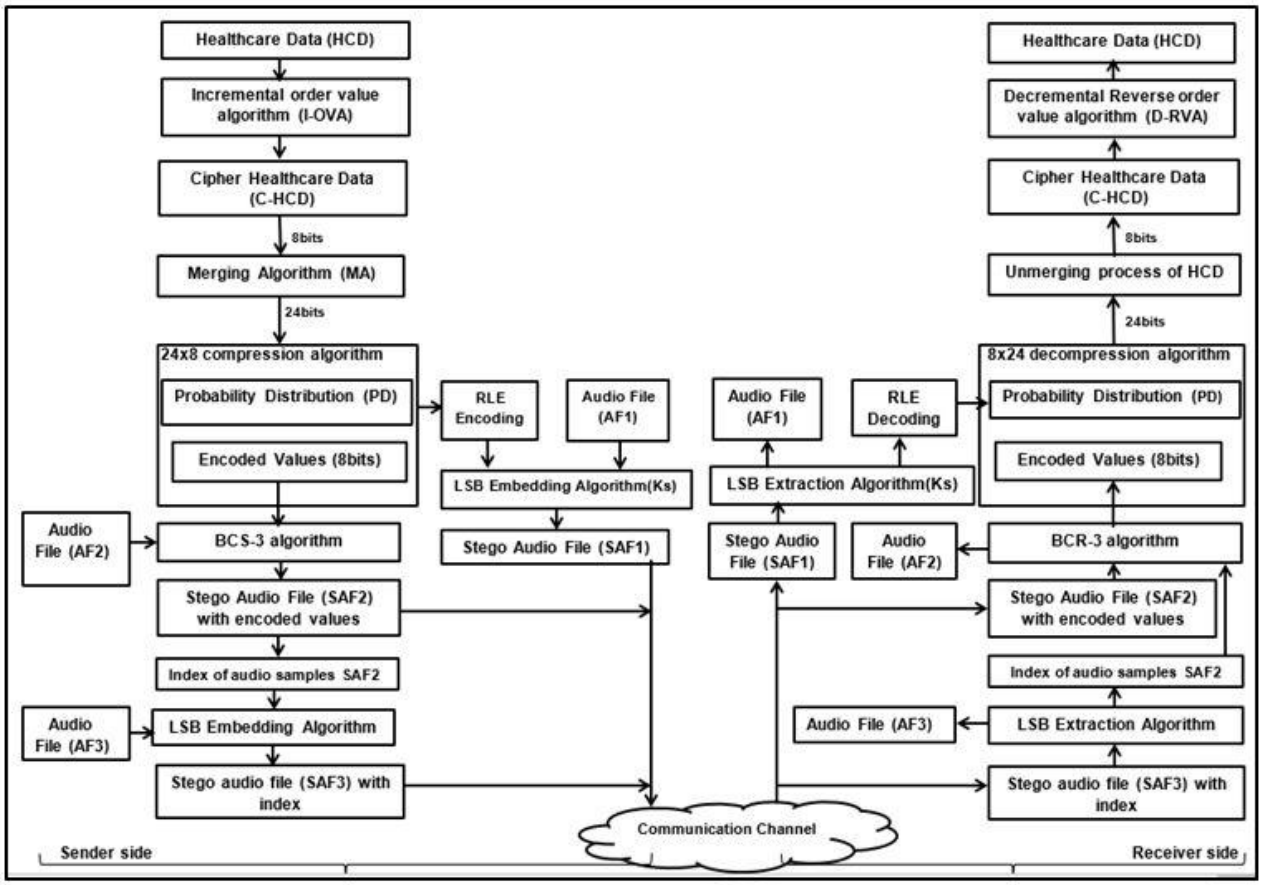


Figure 2: The framework of the proposed algorithm

3.1.6 Theoretical discussion of 24x8 compression algorithm

Figure 3 describes the step-by-step process of compressing M-C-HCD, as shown in Table 3.

1. A probability array of length 16,777,216 was computed, with all elements initialized to zero.
2. The decimal value of the binary "000101110001110101000101" is 1514821. Similarly, the decimal value of "010100100001110000010001"is 5381137.
3. An index array is created by incrementing each decimal value obtained in Step 2 by 1, resulting in indices 1514822 and 5381138.
4. The probabilities are P[1514822]=P[5381138]=1, indicating a single occurrence for each index in the array.
5. The total probability is P[1514822]+P[5381138]=2.
6. Finally, the probabilities are calculated as PD[index]=P[index]/Total probability, resulting in P[1514822] = ½ = 0.5.
   Similarly, P[5381138] = ½ = 0.5.
7. The encoded value is calculated by:
   Encoded_Value[index]=sum(P[index-1]+P[index]/2).
   For index 1514822,
   Encoded_Value[1514822]=0+0.5/2=0.25,
   Similarly, for the index 5381138,
   Encoded_Value[5381138]=0.5+0.5/2=0.75.
8. Scaling the encoded value to 8-bit binary by multiplying the Encoded_Value by 255.
   For index 1514822, Scaled_Encoded_Value [1514822] = 0.25x255=63.75.
9. Similarly, for the index 5381138, Scaled_Encoded_Value [5381138]=0.75x255=191.25.
10. Rounding the Scaled_Encoded_Value to the nearest integer, resulting in 64 for index 1514822 and 191 for index 5381138.
11. Convert the rounded encoded value to binary: 64 = 01000000 and 191 = 10111111.

3.1.7 Phase IV: bit comparison and substitution-3 algorithm (BCS-3)

Input: Audio File (AF2), and encoded_binary_array
Output: Stego Audio File (SAF2)

1. Read the AF2 (.wav) and encoded_binary_array of 8 bits.
2. Convert AF2 to a 24-bit binary stream.
3. Hide an approved code between the sender and receiver to reveal that the SAF2 contains hidden encoded_binary_array data.
4. Compare the AF2 and encoded_binary_array.
5. For i = 1 to N do // Repeat For loop
   until encoded_binary_array
   are hidden.
6. If sum ((AF2) ~= encoded_binary_array) = 3

Merged cipher healthcare (M-C-HCD) data																								Decimal Value	Incremented Decimal Value	Probability	Total Probability	Normalization Probability	Cumulative Probability	Encoded Value	Scaled_Encoded _Value	Rounding_ Scaled_Encoded _Value	Encoded_ binary _array
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24										
0	0	0	1	0	1	1	1	0	0	0	1	1	1	0	1	0	1	0	0	0	1	0	1	1514821	1514822	P[1514822]=1	2	P[1514822]=0.5	0	0.25	0.25*255=63.75.	64	01000000
0	1	0	1	0	0	1	0	0	0	0	1	1	1	0	0	0	0	0	1	0	0	0	1	5381137	5381138	P[5381138]=1		P[5381138]=0.5	0.5	0.75	0.75*255=191.25.	191	10111111

Figure 3: Theoretical discussion of the 24x8 compression algorithm

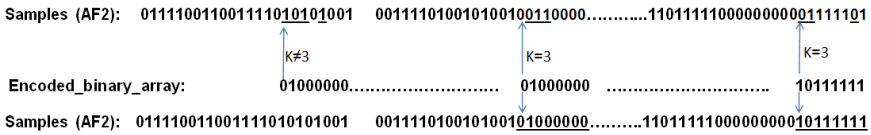


Figure 4: Embedding of encode value 0.25 and 0.75 using the BCS-3 algorithm

7. Embed encoded_binary_array in selected samples of AF2.

   // Embed the Encoded_binary_array in the AF2 selected samples, which differ by exactly 3 bits.
8. End If
9. End For
10. The stego Audio File (SAF2 with hidden encoded_binary_array) is computed.
11. End

The proposed BCS-3 algorithm extends the embedding method [25] through compressed inputs and K. Due to the reduced size, more audio samples are available for comparison and embedding, allowing for improved hiding of encoded values with greater flexibility in sample selection. These modifications in embedding are reflected in the retrieval process by the Bit Comparison and Retrieval-3 (BCR-3) algorithm, as an extension of the retrieval algorithm in [25]. The BCS-3 algorithm compares eight consecutive LSB bits from AF2 with a one-byte Encoded_binary_array. BCS-3 calculates the bit differences (K). If K=3, the algorithm computes and outputs an audio sample from the audio file 2 (AF2). Within this audio sample, eight consecutive bits from the LSBs of AF2 are embedded with a one-byte Encoded_binary_array data.

This process repeats for embedding the Encoded_binary_array data within the audio file 2 (AF2), producing a Stego Audio File 2 (SAF2) with the hidden Encoded_binary_array.

Figure 4 illustrates embedding the encoded_binary_array value obtained from the theoretical discussion of the 24x8 compression algorithm using the BCS-3 algorithm. The first sample is skipped because it has 4-bit differences ($K \neq 3$). The comparison then continues until an AF2 sample with exactly 3-bit differences from the compressed encoded_binary_array is found. In that selected sample, the data 01000000 is embedded. The process then continues with subsequent samples, where the next data value 10111111 is embedded.

The performance of the BCS-3 algorithm is validated using metrics such as Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR), as explained in the Results and Discussion section. Additionally, the algorithm undergoes testing for steganalysis, as claimed by the results of histogram error rate (HER) and the fourth first moments.

3.1.8 Phase V: embedding probabilities in AF1 and indices of SAF2 samples in AF3

Run-length encoding compressed the probabilities with a sequence length of 2^24 for each segment.

It compresses data by representing sequences of repeated values into a single value and a count. The probability sequence for each 501-byte segment typically consists of 16777049 zeros and 167 ones, where the ones correspond to HCD occurrences and are compressed using run-length encoding (RLE). Applying RLE to a 501-byte segment reduces the sequence length from 16777216 to 331 counts, along with their corresponding values, depending on the HCD distribution. The RLE algorithm is applied to all 30 segments within the 15000-byte M-C-HCD, and the resulting RLE counts and values are stored in the probability array.

Decoding the encoded values depends on the probabilities of HCD occurrences at the receiver side. Probability counts and their values are represented in binary using 24 bits for precise retrieval. The traditional least significant bit algorithm embeds probability counts and values using the reference Ks [25] in the least significant bit of audio samples.

An uncompressed digital audio wave file (AF1) with a sample size of three bytes is the input for the LSB algorithm. A private key, known as a secret key (Ks), determines the starting audio sample index for executing the LSB algorithm in the proposed method. Such keys should be shared only with individuals authorized to decode the information. This secret key is selected and mutually agreed upon by both the sender and the receiver. The proposed LSB algorithm utilizes Ks=60 as the private key. For each audio-steganographic communication, the secret key is assigned a new starting embedding sample to enhance security and prevent unauthorized access.

The RLE counts and values with the total in the probability array undergo a conversion from decimal to binary notation of 24 bits. Following this conversion, the resultant binary values of total, counts, and values are sequentially integrated into the AF1 using the LSB algorithm with Ks as reference. Therefore, pinpointing the exact audio sample in the AF1 where the LSB algorithm initiates embedding the probability counts and their values with total poses a challenge.

Figure 3 shows that the probability sequence contains occurrences of M-C-HCD at indices 1514822 and 5381138, while all other indices hold zero values. Applying the RLE algorithm to this input segment

compresses the probability sequence into five counts with their corresponding values, as presented in Table 4.

Each audio sample includes an index value, enabling efficient and direct access. The index of the Stego Audio File 2 (SAF2) containing the embedded Encode_binary_value is documented.

The index value is expressed as a decimal number and converted into a binary representation. The above-indexed values are incorporated into the cover audio file 3 (AF3) through the Least Significant Bit (LSB) algorithm with Ks as a reference. This method extends the approach in [25] by embedding compressed HCD into audio samples to reduce data size, which consequently decreases the number of required index values, with only 5000 indices needed for 15000B of data. It also uses fixed 24-bit index lengths and modified inputs, including Ks and audio samples with encoded values. AF3 is an uncompressed digital audio wave file with a sample size of three bytes. The LSB algorithm embeds the index value in the least significant bit of the audio samples. For embedding 15000B of HCD, the total count of index values is 5000 of 24bits. Pinpointing the exact audio sample in AF3, where the LSB algorithm begins embedding the index value from SAF2, is challenging. Additionally, the size of the index value (24 bits) adds complexity by requiring the handling of larger data segments, making it significantly harder to compromise the LSB algorithm.

The stego audio file SAF1 with RLE counts and probability values, with total and SAF3 with the index values of audio samples SAF2, are transmitted to the receiver.

The metrics PSNR and BER validate the LSB algorithm. The results and discussion section discuss the above metrics.

The private key (Ks) is the initial sample in audio files AF1 and AF3, which initiates the

execution for embedding RLE total, counts, values, and index values. The value of Ks varies for each communication to ensure isolation and security.

The enhanced LSB algorithm uses Ks = 60, where the 60^{th} sample in AF1 and AF3 is used as the private key, mutually agreed upon by the sender and receiver for embedding. This random selection significantly enhances security by expanding the key space, represented as 2^N, where N is the total number of samples, nearly 5300000. Table 6 provides the details of the audio files used in this work. The large number of audio samples and the expanded key space make it computationally difficult for a steganalyst to extract the private key, thereby ensuring robustness in embedding the RLE total, counts, values, and index values.

3.2 Extraction algorithm

3.2.1 Phase I: Extraction of probabilities

Three uncompressed 24-bit stego audio files, SAF1, SAF2, and SAF3, are received. SAF1 contains hidden RLE total, count, and probability values. SAF2 contains encoded values in 8-bit, and SAF3 contains index values of stego audio samples (SAF2).

The initial step in the proposed extraction algorithm involves extracting the RLE total, count, and values of the probability distribution of HCD

from SAF1, with Ks as the reference. An LSB extraction algorithm reads the least significant bit of the stego audio file samples (SAF1) to retrieve the RLE total, counts, and probability values.

The LSB retrieval starts from sample index 60. The first 24 bits represent the total value converted to decimal. This total value indicates the number of 24-bit segments for both the RLE counts and values. The next 24-bit segment represents the first set of counts. The process then repeats, with the next 24 bits representing the next total, determining the number of subsequent 24-bit segments for the next set of counts. This continues until all the count data is retrieved. After all count values are retrieved, the corresponding values are extracted based on the same total values. The extracted counts and values are stored separately and used to reconstruct the probability distribution for further processing. Expanding the RLE count and value generates the probability distribution, which is then input to the 8x24 decompression algorithm.

Table 4: Run Length encoding (RLE) algorithm

Probability zero occurrence	**Probability one occurrence**	**Run Length Encoding Algorithm (RLE)**					
***P[1-1514821]=0** ***P[1514823-5381137]=0** ***P[5381139-16777216]=0**	***P[1514822]=1** ***P[5381138]=1**	**Counts**	**1514821**	**1**	**3866315**	**1**	**11396078**
		Values	**0**	**0.5000**	**0**	**0.5000**	**0**

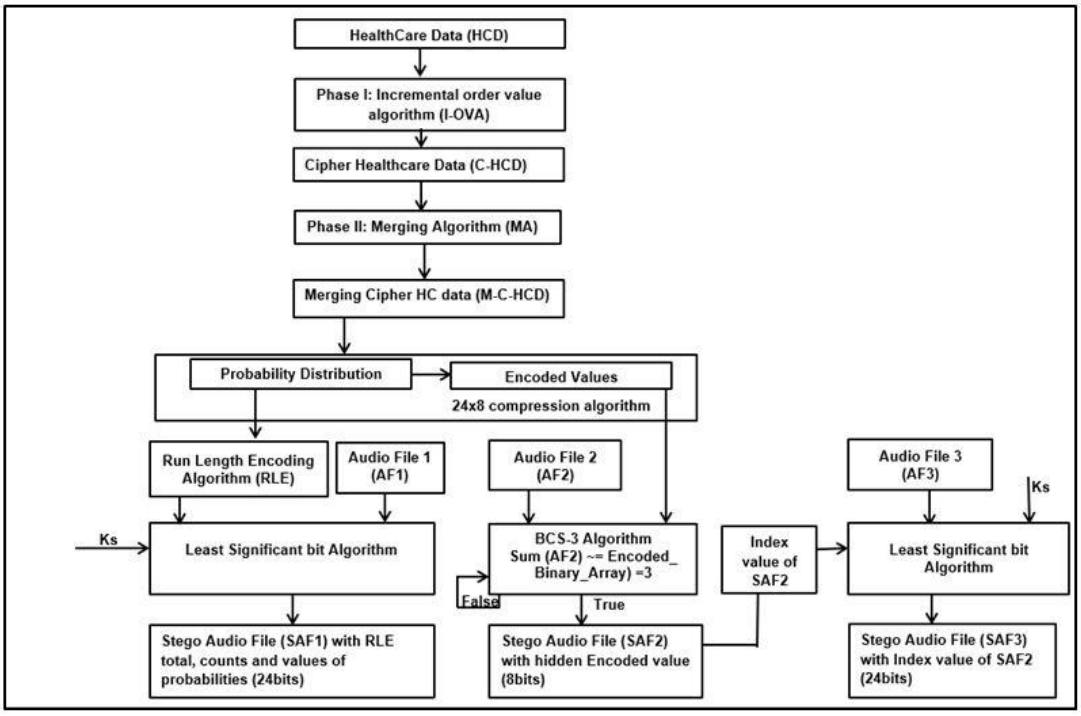


Figure 5: Block diagram for proposed embedding algorithm

3.2.2 Phase II: bit comparison and retrieval-3 algorithm (BCR-3)

The received stego audio file SAF3 was converted into 24-bit binary streams. An LSB extraction algorithm is applied to retrieve the index values, which correspond to the index numbers of audio samples in SAF2 that contain encoded_ binary_values with a 3-bit difference condition. Based on a mutually agreed code between the sender and receiver, the algorithm reads the least significant bit of audio samples in SAF3, starting from sample index 60. Each 24-bit segment corresponds to one index value, converted to equivalent decimal values. This decimal value is the retrieved index value, and the process continues for all retrieved bits. For 15000B of HCD, 5000 index values are retrieved. The retrieved index values are then input to the Bit comparison and retrieval-3 (BCR-3) algorithm. The received stego audio file SAF2 is converted to 24bit binary streams. The encoded_binary_values are extracted using the retrieved index numbers as references. Specifically, the extraction involves retrieving eight consecutive bits from the LSB of the specified index values samples in SAF2. The complete encoded_binary_values are retrieved by repeating the above process. The retrieved encoded values are input into the 8x24 decompression algorithm. The extraction algorithms for retrieving index values are validated using the results of Bit Error Rate (BER) and Normalized Cross-Correlation (NCC), as detailed in the Results and Discussion section.

3.2.3 Phase III: 8x24 decompression algorithm

Input: Probability distribution and Encoded values (8 bits)
Output: Merged Cipher healthcare (M-C-HCD) data (24 bits)

Step 1: Read Inputs
Read the normalized probability distribution: probabilities.
Read the array of encoded values: encoded_binary_array.
Step 2: For each encoded_value in encoded_binary_array:
Convert the 8-bit encoded binary to decimal.
encoded_value ← bin2dec(encoded_binary) / 255
Initialize cumulative_sum ← 0
Step 3: Locate the Index
For J = 1 to N1 // N1 = length of probabilities
cumulative_sum ← cumulative_sum + probabilities[J]
If encoded_value < cumulative_sum: Break
Step 4: Convert Index to Binary
decoded_index ← J - 1
Convert decoded_index to 24-bit binary:
M-C-HCD_segment←dec2bin(decoded_index, 24)
Step 5: Store the Recovered Segment
Append M-C-HCD_segment to M-C-HCD_array
Step 6: Output
Return M-C-HCD_array as the reconstructed M-C-HCD.

3.2.4 Phase IV: decremental reverse order value algorithm (D-RVA)

Input: Merged cipher Healthcare data (M-C-HCD in 24 bits)
Output: Healthcare data (HCD)

1. Read the M-C-HCD of each segment of 15000B.
2. Split the 24 bits of merged cipher healthcare data (M-C-HCD) into 8 bits of cipher HCD.
3. If the length of C-HCD is not divisible by 3, remove the last row or the last two rows if they contain exactly 1 byte of zero binaries added during the merging algorithm.
4. Convert each 8-bit binary cipher HCD segment to its corresponding decimal value. Each decimal value

represents the index of printable characters in the ASCII table.
5. Using the decimal values as indices, fetch the corresponding characters from the ASCII table.
6. Retrieve the ASCII value that corresponds to the fetched cipher HCD.
7. Decrement the ASCII value based on the position of C-HCD.
8. Fetch the cipher character corresponding to the decremented ASCII value.
9. Retrieve the assigned order value for the fetched character.
10. Use the order value as an ASCII value and fetch the corresponding character.
11. The characters fetched in step 10 result in the decryption of the cipher healthcare data.
12. Repeat the above steps for each segment within the 15000B data.
13. Combine each decrypted segment to form the complete decrypted HCD.
14. End

3.3 Computational complexity

To efficiently handle the large probability distribution (PD) of size 2^{24}, the proposed compression–decompression framework processes input in fixed 501-character segments, limiting active PD storage to O(u), where u is the number of unique entries in a segment. During compression, each PD is RLE-compressed, reducing storage from O(2^{24}) to a few hundred entries O(u). Each segment uses a dedicated PD in memory, and memory usage remains low without needing explicit variable clearance. In decompression, RLE-decoded PDs are stored in a cell array, retrieved per segment, used, and cleared, maintaining constant memory usage. The system runs without memory overflow on a 4 GB RAM machine, confirming feasibility. Computational complexity for both compression and decompression is O(1) within the tested input range, as shown in Figure 6. The results were based on 75 simulations supported by very low R^2 values (0.045 for compression, 0.047 for decompression), indicating minimal dependency on input size and ensuring scalability for larger datasets.

The execution time was evaluated through 75 simulations using five audio files and five dataset sizes (3000–15000 bytes), each combination executed three times for consistency. The computational complexity of both the single and triple-audio-file approaches is O(n), with execution time increasing proportionally to the HCD size. The average execution time of the proposed algorithm is 56.1236 s for the three-audio-file approach and 19.65634 s for the single-audio-file implementation. Despite the lower execution time for the single-audio-file approach, the three-audio-file approach enables more efficient and modular management of structured healthcare data. By isolating the embedding of healthcare data (HCD), index values, and RLE counts/values into separate audio streams, the system reduces coupling and supports incremental updates, where only modified components require re-encoding instead of reprocessing all three files. This modular design also allows independent validation of each HCD, minimizing the risk of systemic corruption. Furthermore, the three audio files provide scalability for larger datasets and establish a foundation for future parallelized I/O and processing pipelines. Consequently, despite the additional runtime, the three-audio-file approach provides greater technical robustness and long-term operational efficiency.

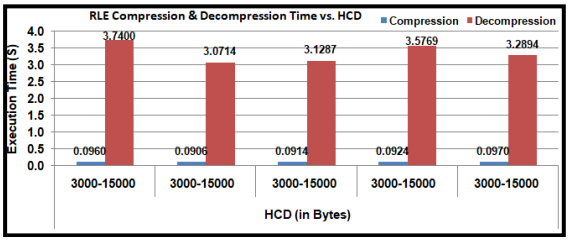


Figure 6: Execution time for RLE Compression and Decompression

4 Results and discussion

The proposed embedding and extraction algorithm was implemented and evaluated using MATLAB software on over fifteen audio files with embedding probability distribution, healthcare data, and index values of audio samples. Uncompressed audio files from the Mixkit dataset [34], with a consistent 44,100 Hz sampling rate, ensure uniform audio quality and minimize variability during embedding or extraction processes.

The selected audio files have a one-minute duration, enabling fast processing and facilitating repeated testing within a reduced time frame.

The dataset's genre diversity offers a variety of audio contexts, allowing a comprehensive assessment of the algorithm's adaptability across different scenarios.

Open availability of the Mixkit dataset ensures the reproducibility of experiments.

Utilizing unprocessed audio files minimizes computational overhead and accurately reflects performance conditions for the steganography algorithms.

Table 6 lists the specifications of the audio files used in the algorithms [35].

The HCD consists of the COVID Dialogue Dataset, which includes conversations between patients and doctors about COVID-19 and other types of pneumonia, sourced from websites such as icliniq.com, healthcaremagic.com, and healthtap.com [36].

4.1 Throughput

Throughput evaluates the performance of encryption and decryption algorithms by measuring the data transfer rate using Equation (1) [37].

$$\text{Throughput} = \frac{\text{Total Size of Healthcare data(B)}}{\text{Total Execution Time}} \quad (1)$$

The data transfer rate for 15000B using the proposed I-OVA and D-RVA algorithms is 8592.74 KB/s, which signifies that 15000B of data was transferred in 0.0017s. The above indicates that the time taken for each step of the

proposed algorithm is minimal, resulting in high throughput.

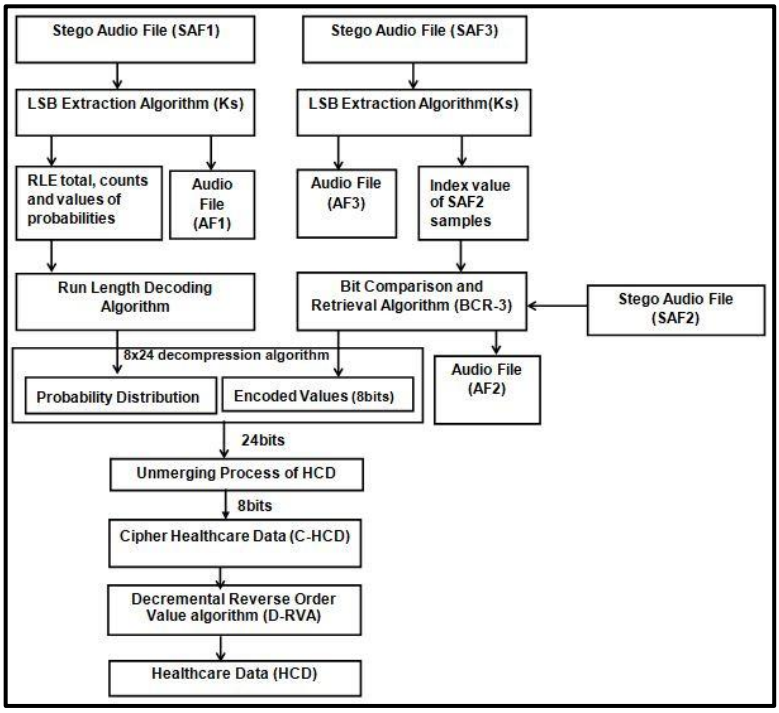


Figure 7: Block diagram for proposed extraction algorithm

Probability	Binary_ encoded _Value	Decimal _ encoded_ Value	Encoded_Value	Cumulative Probability (J)	J Decimal Value	J-1 Decimal Value	Merged cipher healthcare data (M-C-HCD)																									C-HCD							
							1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	1	2	3	4	5	6	7	8
																															0	0	0	1	0	1	1	1
P[1514822]=1	00111111	63	63/255=0.247088	0.5	1514822	1514821	0	0	0	1	0	1	1	1	0	0	0	1	1	1	0	1	0	1	0	0	0	1	0	1	0	0	0	1	1	0	1
																															0	1	0	0	0	1	0	1
																															0	1	0	1	0	0	1	0
P[5381138]=1	10111111	191	191/255=0.74901	1	5381138	5381137	0	1	0	1	0	0	1	0	0	0	0	1	1	1	0	0	0	0	0	1	0	0	0	1	0	0	0	1	1	1	0	0
																															0	0	0	1	0	0	0	1

Figure 8: Theoretical discussion of the proposed extraction algorithm

Table 5: Decremental reverse order value algorithm

Binary Value_Index I-OVA								Index I-OVA	C-HCD	ASCII Code	D-RVA value	ASCII Code	ASCII character	OVA value	ASCII Code	HC data
1	2	3	4	5	6	7	8									
0	0	0	1	0	1	1	1	23	7	55	55-1=54	54	6	105	105	i
0	0	0	1	1	1	0	1	29	=	61	61-2=59	59	;	100	100	d
0	1	0	0	0	1	0	1	69	e	101	101-3=98	98	b	61	61	=
0	1	0	1	0	0	1	0	82	r	114	114-4=110	110	n	49	49	1
0	0	0	1	1	1	0	0	28	<	60	60-5=55	55	7	104	104	h
0	0	0	1	0	0	0	1	17	1	49	49-6=43	43	+	116	116	t

Table 6: Description of audio files [35]

Specifications	
Bit per sample	24
Sample Rate	44100
Channel	2
Audio Type	Music
Duration in Minutes	1

Recent research used 3-DES for securing healthcare data [38], [39].To assess efficiency, the proposed algorithm is compared with 3-DES. Figure 9 compares the throughput values achieved by the proposed algorithms, I-OVA and D-RVA, relative to those of the 3-DES algorithms.

The x-axis denotes the size of healthcare data in bytes, and the y-axis represents throughput in kilobytes per second. The improved throughput is due to the algorithm's simple structure, which avoids complex key scheduling and multiple encryption rounds, reducing overall computation time.

The results prove that the algorithm is scalable, transferring large volumes of HCD without significant degradation. Hence, the proposed algorithm is suited for telemedicine consultations involving healthcare data, ensuring efficient and secure communication.

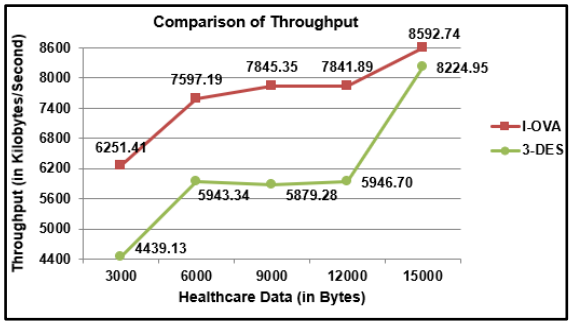


Figure 9: Compares the throughput achieved by the proposed techniques with other encryption algorithms

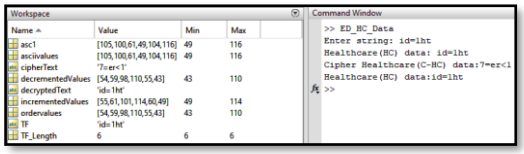
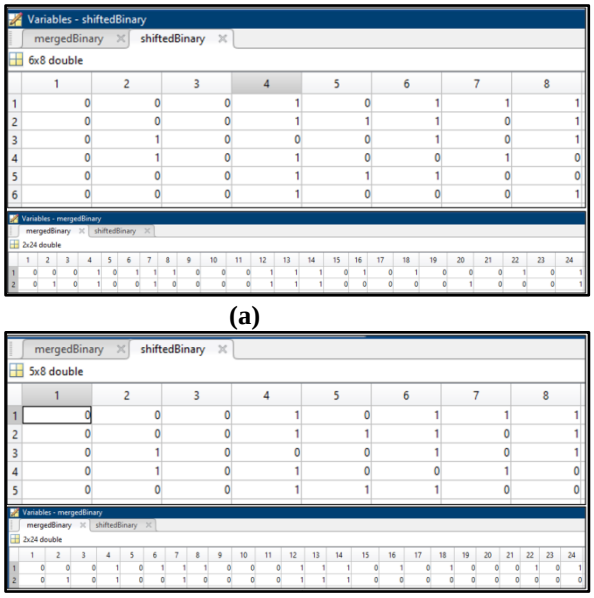


Figure 10: Displays the results of the incremental order value algorithm (I-OVA) and the decremental reverse order value algorithm (D-RVA)



(a)

(b)

Figure 11: Results of the merging algorithm for C-HCD with length (a) divisible by 3 and (b) not divisible by 3

Figure 10 presents the simulation results for the proposed I-OVA and D-RVA algorithms for the HCD listed in Table 2.

Figure 11 displays the outcomes of the merging algorithm. It verifies that when the length of the C-HCD is not divisible by 3, the algorithm appends one or two bytes of zero binary bits at the end of the C-HCD. In Table 2, the C-HCD "23,29,69,82,28,17" has a length of 6, divisible by 3, resulting in the M-C-HCD as "23,29,69" and "82,28,17", which does not require the addition of zero bits. Conversely, when C-HCD is "23, 29, 69, 82, 28" with a length of 5, which is not divisible by 3, the algorithm appends one byte of zero bits. This results in the M-C-HCD as "23, 29, 69" and "82, 28, **00000000**".The decimal values of the C-HCD are first converted to their binary equivalents before merging.

4.2 Entropy

The statistical security of cipher HCD in randomness is measured with entropy given by H(s) in Equation (2) [37].

$$H(s) = \sum_{i=0}^{n-1} p(s_i) \log_2 \frac{1}{p(s_i)} \quad (2)$$

The probability symbol S_i is represented as $P(S_i)$

Figure 12 shows the entropy comparison with the 3-DES algorithm, with the x-axis representing HCD in bytes (B) and the y-axis representing entropy. The entropy of the proposed algorithms is 6.5, which results in strong randomness and resistance to frequency analysis and pattern recognition attacks of the ciphertext. The reason for randomness is the positional increment of the ASCII value in I-OVA. The comparison graph shows that the proposed encryption algorithm is robust, achieving an entropy of 6.5, approximately the same as the standard 3-DES value of 7. This suggests that the algorithm provides strong security with efficiency comparable to 3-DES. The proposed I-OVA ensures the first layer of protection for HCD in audio-steganographic communication.

4.3 Compression ratio (CR)

CR is the ratio of the total number of bits in uncompressed data to the total number of bits in compressed data, expressed in bit per bit (bpb) in Equation (3) [11].

$$CR = \frac{\text{No: of bits in uncompressed data}}{\text{No: of bits in compressed data}} \quad (3)$$

The average CR of the proposed 24x8 compression algorithm is 3:1, as shown in Figure 13, compared to values from existing compression algorithms. The proposed 24×8 compression is more efficient than existing algorithms by eliminating statistical and dictionary dependencies, ensuring faster and consistent compression with minimal complexity. The x-axis denotes the size of healthcare data in bytes (B) (3000B, 6000B, 9000B, 12000B, and 15000B), while the y-axis denotes the compression ratio in bpb. This compression ratio indicates that the healthcare data is compressed to one-third of its original size, reducing 24 bits of M-C-HCD to 8 bits. This results in a significant reduction of the healthcare data from 15000B to 5000B.

Space saving is defined as the reduction in file size relative to the uncompressed size, as given in Equation (4) [11]

$$\text{Space savings} = 1 - \frac{\text{No.of bits in compressed data}}{\text{No.of bits in uncompressed data}} \quad (4)$$

The proposed 24x8 compression algorithm achieves a space savings of 66.67%. The above indicates that the compressed data occupies about one-third of the original storage space (33.33%), which results in significant storage reduction.

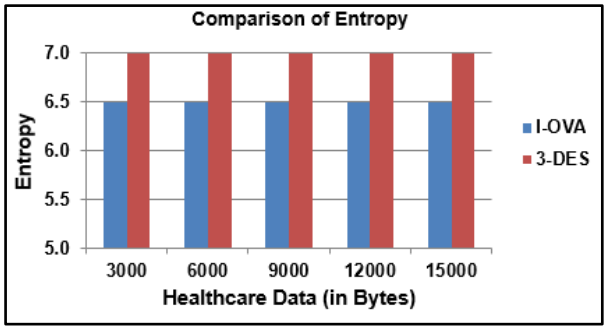


Figure 12: Compares the entropy achieved by the proposed techniques with that of the 3-DES algorithm

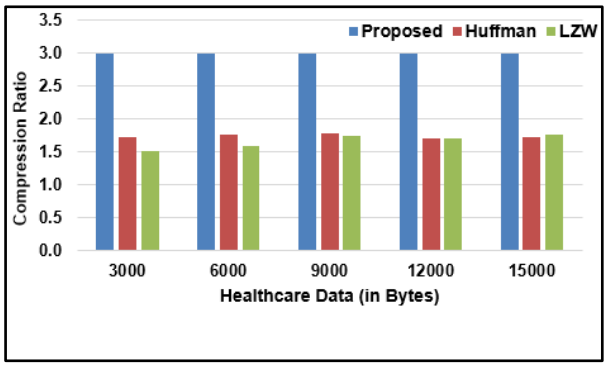


Figure 13: Comparison of different compression schemes in terms of Compression Ratio

Indirectly, efficient compression reduces transmission costs by enabling faster transfer of highly compressed data. Therefore, the 24x8 compression and 8x24 decompression algorithms are efficient. Hence, the simulation results demonstrate that these 24x8 compression and 8x24 decompression algorithms communicate the HCD faster with enhanced capacity during telemedicine consultations.

To handle larger datasets efficiently, the 24-bit HCDs are compressed to 8-bit representations. The normalized values from the compression algorithm are multiplied by 255; the results are rounded and converted to 8-bit binary, as explained in the 24x8 compression algorithm. This reduces the memory required to store each value from 24 bits (3 bytes) to one-third of the original size, 8 bits (1 byte). The 24x8 compression algorithm significantly reduces the storage requirements and memory usage. Hence, HCD datasets larger than 15 KB can be effectively handled without affecting the computational integrity of the algorithm.

4.4 Similarity analysis

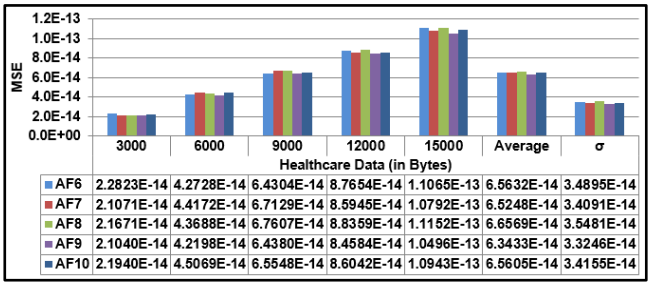
4.4.1 Mean squared error (MSE)

The MSE measures the distortions between the cover and stego audio files. An MSE value of approximately zero indicates minimal distortions, as given by Equation (5) [35].

$$\mathrm{MSE} = \frac{1}{Q}\sum_{i=1}^{Q}(\mathrm{CAF_i} - \mathrm{SAF_i})^2 \qquad (5)$$

$\mathrm{CAF_i}$ and $\mathrm{SAF_i}$ represent samples from the cover audio file and stego audio file, respectively, and Q is the total number of audio samples.

The average MSE value of the proposed BCS-3 algorithm was 6.5297E-14 from 25 simulation runs. Figure 14 illustrates the relationship between different sizes of HCD (3000B, 6000B, 9000B, 12000B, and 15000B) on the x-axis and the MSE of various audio files (AF6, AF7, AF8, AF9, and AF10) on the y-axis. Table 7 compares the average MSE of the proposed and existing algorithms, highlighting significant differences. The proposed BCS-3 algorithm embeds an 8-bit encoded binary value into 24-bit audio samples, resulting in a consistent 3-bit distortion between the CAF and SAF for each 1-byte embedding, thereby outperforming the conventional LSB algorithm, which



	3000	6000	9000	12000	15000	Average	σ
AF6	2.2823E-14	4.2728E-14	6.4304E-14	8.7654E-14	1.1065E-13	6.5632E-14	3.4895E-14
AF7	2.1071E-14	4.4172E-14	6.7129E-14	8.5945E-14	1.0792E-13	6.5248E-14	3.4091E-14
AF8	2.1671E-14	4.3688E-14	6.7607E-14	8.8359E-14	1.1152E-13	6.6569E-14	3.5481E-14
AF9	2.1040E-14	4.2198E-14	6.4380E-14	8.4584E-14	1.0496E-13	6.3433E-14	3.3246E-14
AF10	2.1940E-14	4.5069E-14	6.5548E-14	8.6042E-14	1.0943E-13	6.5605E-14	3.4155E-14

Figure 14: The impact of MSE on different sizes of healthcare data and audio files

may alter between 0 to 8 least significant bits per byte, leading to non-deterministic and potentially higher distortion. This difference progressively increases during the embedding process, reaching 15000 bits out of 40000 bits for an HCD of 15000B. Specifically, 5000 encoded values for 15000B of HCD result in 40000 (5000x8) embedding bits. For each 8-bit embedding, 3 bits will be error bits, reaching 15000 error bits. The above ensures that only 3 bits in a 24-bit audio sample are distorted, resulting in a reduced MSE of 6.5297E-14, close to the benchmark value.

The standard deviation of MSE values ranges from 3.3246E-14 to 3.5481E-14 across all HCD and audio types, as shown in Figure 14. This results from the algorithm's consistent data embedding across all audio files, regardless of their specific properties, resulting in stable performance.

The uniform embedding process introduces minimal distortion, further supported by the consistent sampling rate of the audio files.

Statistical analysis yields a p-value of 0.999931 ($p > 0.05$), accepting the null hypothesis and indicating no significant differences in MSE across audio files.

This result suggests that the MSE does not vary with the bit patterns in the audio samples and HCD. The algorithms introduce a consistent error bit pattern to demonstrate that the audio steganography method effectively controls distortion in HCD. Consequently, the audio quality remains largely unaffected during HCD embedding.

4.4.2 Peak signal to noise ratio (PSNR)

In Equation (6) [35], the PSNR expressed in decibels is a logarithmic measure of the stego signal quality.

$$\text{PSNR} = 10\log_{10}\frac{(2^{q}-1)^2}{\text{MSE}} \qquad (6)$$

Where q is the maximum number of bits in the cover audio sample.

Higher PSNR results in enhanced quality of the stego audio file. Figure 15 illustrates the relationship between different sizes of HCD (3000B, 6000B, 9000B, 12000B, and 15000B) on the x-axis and the PSNR of various audio files (AF6, AF7, AF8, AF9, and AF10) on the y-axis. Table 7 compares the average PSNR of the proposed algorithm with the existing algorithms. The average PSNR of the proposed BCS-3 algorithm is 42.1480 dB. This is due to the average MSE of 6.5297E-14, which is inversely related to PSNR. Hence, the proposed BCS-3 algorithm improves upon existing methods by using 8-bit embedding to achieve higher payload capacity. It embeds 15,000 B of data without significantly altering the stego audio file, resulting in 15,000 error bits that enhance resistance to steganalysis while maintaining high audio quality. This higher PSNR is due to the algorithm's design, where each 8-bit embedded value differs by only 3 bits from the original, minimizing distortion compared to traditional LSB-based methods.

Similarly, PSNR is measured for the stego audio files (SAF1, SAF2, SAF3, SAF4, and SAF5) with hidden RLE total, counts, and values, is 30.1417dB. PSNR for the stego audio files (SAF11, SAF12, SAF13, SAF14, and SAF15) with hidden index values of stego audio samples (SAF6, SAF7, SAF8, SAF9, and SAF10) containing the encoded value is 36.5499 dB. Both PSNRs indicate no degradation in the quality of stego audio files.

The 15000-bit differences resulting from embedding do not impact audio intelligibility, as evidenced by an average PSNR of 42.1480 dB and a low standard deviation of 2.7 across various audio files. A PSNR above 36 dB is typically considered imperceptible to the human ear [40]. The consistent PSNR and MSE values across different HCD sizes confirm the stability of the BCS-3 algorithm.

Figure 16 graphically depicts the imperceptible difference between the cover audio file and the stego audio file for various genres of audio files ranging from AF6 to AF10, each embedded with 15000B of healthcare data. These results confirm that the embedding is both imperceptible and resistant to detection.

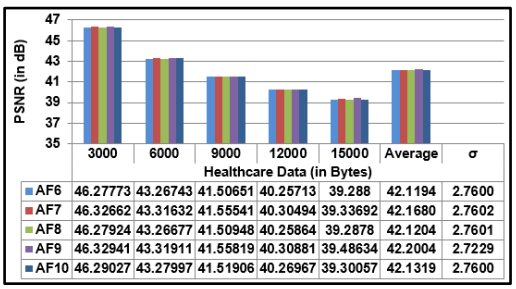


Figure 15: The impact of PSNR on different sizes of healthcare data and audio files

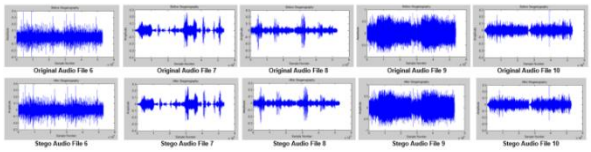


Figure 16: Various cover audios and stego audios with 15000B of healthcare data

4.5 Hiding capacity (Payload)

The hiding capacity represents the maximum size of healthcare data embedded in an audio file, calculated using Equation (7) and expressed as a percentage [35].

The hiding capacity in the LSB algorithm is given by Equation (8) in % [35]

$$\text{Hiding Capacity (HC)} = \left(\frac{\text{Text File size}}{\text{Audio File size}}\right) * 100 \qquad (7)$$

$$\text{Hiding Capacity (HC)} = \left(\frac{\text{number of samples}}{8}\right) * 100 \qquad (8)$$

Table 7: The MSE and PSNR Comparison between the Proposed and Existing Algorithms

Method	Text Files (Bytes)	MSE	PSNR (dB)
[26]	107	9.08875 ▲100%	34.2775 ▲22.96%
[27]	103	-	17.71916498 ▲137.87%
[28] LSB	250	0.3994 ▲100%	-
[28] Pattern Matching	250	0.5619 ▲100%	-
[29]	9405	0.8578 ▲100%	-
Proposed	**15000**	**6.5297E-14**	**42.1480**

The simulated results for embedding 15000B (5000 encoded values) of HCD into an audio file AF7 with 5365030 samples require 5000 samples for the proposed BCS-3 algorithm, as summarized in Table 8. Each encoded value represents 3 bytes of HCD. In comparison, the LSB algorithm requires 120000 samples, indicating a reduction of 115000 samples with the proposed method. For 15000B of HCD, the number of audio samples needed by the BCS-3 algorithm is 5000, based on its 3:1 embedding ratio. Each encoded value requires 1 audio sample. In contrast, the LSB algorithm requires 15000B×8=120000samples, reflecting its 1:8 embedding ratio.

The proposed BCS-3 algorithm ensures that the number of samples used for embedding is proportional to one-third of the length of HCD in bytes, based on its 3:1 embedding ratio.BCS-3 embeds 1 byte per sample, whereas LSB embeds 1 byte per 8 samples. This ensures the BCS-3 algorithm optimizes sample utilization by reducing the number of samples required for embedding based on the embedding ratio. Table 8 compares the number of audio samples used by the LSB method and the proposed BCS-3 algorithm for embedding healthcare data of varying sizes.

4.5.1 Embedding rate

The embedding rate is the number of bits embedded in a one-byte audio file sample. The LSB algorithm embeds 1 byte of the HCD in 8 audio samples of 1 byte. This results in an embedding rate of 12.5%. However, the BCS-3 algorithm embeds 1 byte of encoded value into 1 audio sample (24 bits) with a 33.33% embedding rate. For the tested audio AF8 with samples 5306818, the proposed algorithm achieves a 20.83% increase in embedding rate compared to the LSB algorithm. The proposed algorithm requires only 5000 audio samples to embed 5000 bytes of encoded values. Since each audio sample is 3 bytes, the 5000 samples correspond to 15,000 bytes of audio, resulting in 33.33% embedding efficiency. Hence, the BCS-3 algorithm has significantly improved its embedding rate compared to the traditional Least Significant Bit algorithm.

Table 8: Audio samples utilized by the proposed BCS-3 algorithm for AF7 compared with the traditional LSB algorithm

Healthcare data (in Bytes)	Samples Utilized (in samples)		Difference Bit (DB) (in Bits)	
	LSB	Proposed Method	LSB	Proposed Method
3000	24000	1000	4033	3000
6000	48000	2000	7968	6000
9000	72000	3000	12160	9000
12000	96000	4000	16011	12000
15000	120000	5000	19876	15000

4.6 Strength analysis

4.6.1 Bit error rate (BER)

BER measures the correctness of embedding with the percentage of the embedded HCD bits that were retrieved incorrectly, as expressed by Equation 9.

$$\mathrm{BER} = \frac{\mathrm{HCD_{error}}}{\mathrm{HCD_{bits}}} \tag{9}$$

Where $\mathrm{HCD_{error}}$ is the retrieved error bits and $\mathrm{HCD_{bits}}$ is the total number of bits embedded. In the absence of attacks, the embedded HCD is retrieved 100% successfully using the proposed BCR-3 retrieval algorithm for all tested audios with varying HCD sizes [35].

It is also defined as the ratio of the difference in bits between cover and stego audio files with hidden healthcare data to the total number of bits in the audio file.

$$\mathrm{BER} = \frac{1}{Q}\sum_{i=1}^{Q} \begin{matrix} 1, \mathrm{CAF(i)} \neq \mathrm{SAF(i)} \\ 0, \mathrm{CAF(i)} = \mathrm{SAF(i)} \end{matrix} \tag{10}$$

Where CAF is the original audio, SAF is the stego audio, and Q is the total samples in the audio files. The lower BER indicates the quality stego audio file

The average BER of the proposed BCS-3 algorithm across the tested files (AF6–AF10) is 7.0165E-05, confirming the preservation of audio quality. Since the embedding process modifies the audio in a controlled manner, it ensures the reliable extraction of the embedded Healthcare data.

The above is evident in Figure 17, plotted between the length of HCD (Bytes) on the horizontal axis and BER on the vertical axis. The computed BER ranges from 2.3284E-05 for audio file AF9 with an HCD of 3000B to 1.1782E-04 for audio file AF8 with an HCD of 15000B, as compared to the BER reported by existing researchers [26, 30] in Table 9. The proposed BCS-3 algorithm outperforms the results of previous researchers, in which only one-eighth (3/24) of the original audio file is distorted during the embedding process. This accomplishment was achieved by embedding the healthcare data into the stego audio files (SAF6-SAF10), with alterations occurring in 3 out of 24 bits corresponding to the cover audio files (CAF6-CAF10). Specifically, the BCS-3 algorithm embeds three characters as one byte encoded value in 24-bit

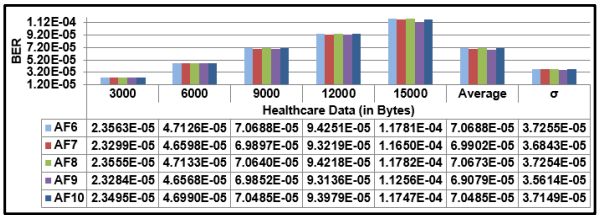


Figure 17: Influence of BER on various HCD sizes and audio files

audio samples with a bit difference of 3, reducing error bits during the embedding process. The BER resulted from altering 3 out of 24 bits while embedding 1 byte of HCD in 24-bit audio samples. Embedding 15000B of HCD generated 15000 error bits. The error bits produced during the embedding process are proportional to the length of the HCD in bytes at a 1:3 ratio, while for audio samples, the ratio is 3:1. This predictable bit-alteration pattern minimizes bit errors and ensures consistent BER performance. The BCS-3 algorithm optimizes the embedding process through effective error bit control and efficient audio sample utilization, ensuring predictable BER outcomes. In contrast, the LSB algorithm shows no predictable pattern of error bits, as its error bits vary depending on the bit pattern of the HCD and audio samples. The above leads to unpredictable bit alterations, higher BER variability, increased error rates, and reduced performance due to ineffective bit management.

The average BER values indicate that the algorithm's performance remains consistent regardless of the specific audio sample used. The standard deviation values, ranging from 3.561E-05 to 3.725E-05, reflect this consistency and are further supported by the p-value of 0.9999934($p>0.05$) in statistical analysis. Figure 17 presents these low standard deviations, indicating that the proposed algorithm performs consistently across different inputs.

The average BER of the stego audio files SAF1-SAF6 for retrieving the run-length encoding total count, individual count values, and corresponding data values is 0.001118, indicating successful decoding of the RLE values. The stego audio files SAF11-SAF15 successfully decoded the index values, achieving a BER of 0.000258.

4.6.2 Normalized cross-correlation(NCC)

NCC measures the degree of correlation between embedded and extracted Healthcare data and between cover audio and stego audio files. NCC given by Equation (11).

$$\mathrm{NCC}(\mathrm{HCD},\mathrm{HCD}') = \frac{\sum_{K=1}^{P}\mathrm{HCD}(K)\,\mathrm{HCD}'(K)}{\sqrt{\sum_{K=1}^{P}\mathrm{HCD}\,(K)^2}\sqrt{\sum_{K=1}^{P}\mathrm{HCD}'(K)^2}} \quad (11)$$

Where HCD and HCD' are embedded and extracted Healthcare data. P is the number of samples in the audio file [35]. The correlation values of NCC range from -1 to 1, where 1 indicates similarity, -1 indicates dissimilarity, and 0 indicates no correlation.

The NCC value between the embedded and extracted healthcare data is 1, indicating perfect correlation. This confirms the accurate and complete retrieval of the secret message in simulations of the proposed BCR-3 algorithm in the absence of attacks [35].

The NCC value of the proposed BCS-3 algorithm is 0.999999, indicating a high degree of correlation between the cover audio file and the stego audio file. Table 9 compares the NCC value obtained from the proposed BCS-3 algorithm with existing algorithms [31]. The NCC value of 0.999999 results from the predictable 3-bit differences introduced when embedding 15000 B of healthcare data into audio files (AF6-AF10), amounting to 40,000 bits (5,000 bytes × 8) after compression. Each encoded value (1 byte or 8 bits) in HCD is incorporated in the cover audio file with a 3-bit difference between the cover audio file sample and its corresponding 24-bit stego audio file sample on comparison. The bit difference generated for the 15000B embedded data is 15000 bits. This predictable bit-alteration pattern ensures a consistent error pattern in the cover audio file with minimal distortion after the embedded process, as proved in Figure 16.

4.7 Steganalysis

4.7.1 Histogram attack

The Histogram Error Rate (HER) measures the deviations between the cover and stego audio files while embedding healthcare data. The frequency distribution of cover and stego audio file samples is compared based on HER values ranging from 0 to 1. An HER of 0 indicates perfect similarity between the cover and stego audios.

In this study, 25 simulations were conducted using five different audio files labeled AF6 to AF10. These involved embedding healthcare data of various lengths, including 3000B, 6000B, 9000B, 12000B, and 15000B. The Histogram Error Rate (HER) of the proposed BCS-3 algorithm was assessed for each simulation using Equation (12), with the findings presented in Figure 18 [35].

$$\mathrm{HER} = \frac{\sum_{i=1}^{Q}(\mathrm{His_a} - \mathrm{His_s})^2}{\sum_{i=1}^{Q}(\mathrm{His_a}^2)} \quad (12)$$

Where $\mathrm{His_a}$ and $\mathrm{His_s}$ are the histograms of audio and stego audio files.

Table 9: The BER and NCC comparison between the proposed and existing algorithms

Method	BER	NCC
[26]	17.995% ▲99.96%	-
[30]	0.0187% ▲62.47%	-
[31]	-	0.99992 ▲0.008%
Proposed	**7.0165E-05**	0.999999

Table 11 compares the HER values between the proposed BCS-3 algorithm and existing algorithms. The average HER obtained for the proposed algorithm is 0, indicating a high resistance to histogram attacks.

Additionally, Figure 18 displays the histograms of five tested cover audio files (AF6 to AF10) with embedded 15000B healthcare data. The histograms before and after steganography are similar, indicating that the magnitude distribution of the stego audio file closely resembles that of the cover audio file.

4.7.2 Fourth first moments

The statistical distribution of the amplitude of cover and stego audios with embedded healthcare data is measured by the fourth first moments: average (μ), variance (σ^2), skewness (sk), and kurtosis (k). The Equations (13),(14),(15),(16), and (17) find the fourth first moments and the Difference Ratio (DR), respectively [32].

$$\mu = \frac{\sum_{i=1}^{Q} S_i}{Q} \quad (13)$$

$$\sigma^2 = \frac{\sum_{i=1}^{Q}(S_i - \mu)^2}{(Q-1)} \quad (14)$$

$$\mathrm{sk} = \frac{\sum_{i=1}^{Q}(S_i - \mu)^3}{(Q-1)\sigma^3} \quad (15)$$

$$\mathrm{k} = \frac{\sum_{i=1}^{Q}(S_i - \mu)^4}{(Q-1)\sigma^4} \quad (16)$$

$$\text{Difference Ratio(DR)} = \left|\frac{\mathrm{fm_a} - \mathrm{fm_s}}{\mathrm{fm_a}}\right| \times 100 \quad (17)$$

Where fma, and fms are any fourth first moments of an audio file and a stego audio file, respectively. S_i denotes the Samples in the cover or stego audio files, and Q represents the total number of audio samples in the cover or stego audio files.

The DR ranges from 0 to 1, with 0 indicating similar statistical distributions between cover and stego audio and 1 indicating significant differences. Table 11 presents the simulation results of the DR for the fourth first moments values in comparison with existing algorithms. These results were obtained by testing five different audio files (AF6-AF10) with 15000bytes of healthcare data embedded into them. The simulation results of DR are approximately zero, indicating resistance to statistical attack.

The average skewness is 6.7953E-07, showing no significant outliers in the amplitude of stego audio files with hidden healthcare data. Similarly, the average kurtosis of 3.5282E-09 indicates a stable amplitude distribution of audio samples with light tails, even after embedding 15000B of healthcare data. The average

variance value of 2.5759E-09 confirms that the trade-off between the embedding capacity of 15000B healthcare data and the integrity of cover audio files is maintained. Lastly, the mean of 1.8331E-04 indicates minimal perceptible change in the listening experience of stego audio files with 15000B hidden healthcare data during audio steganographic communication.

4.7.3 Re-sampling and AWGN attack:

Re-sampling involves down-sampling the stego audio from 44.1 kHz to 22.58 KHz and subsequently reconstructing it back to 44.1 KHz before transmission. This process alters sample values and can distort the embedded HCD. To evaluate resampling attacks, 25 simulation tests were conducted on five audio files. The stego audios were subjected to the resampling process, and performance was analyzed. The results demonstrate that the BCS-3 algorithm preserved audio quality under this attack. Specifically, the stego audios achieved a PSNR of 54.5 dB, a very low MSE of 6.0×10^{-6}, and a high NCC value of 0.987 between the original and resampled audio files. These values confirm that the distortion introduced by re-sampling was negligible.

White Gaussian noise was added to the stego audios at SNR levels of 0.001dB, 5dB, and 25dB. The algorithm was tested with 25 simulations on 5 audio files with HCD data, 3000 to 15000bytes. The attack resulted in HER of 2.06×10^{-1}, 7.34×10^{-2}, and 1.14×10^{-4} for 0.001 dB, 5 dB, and 25 dB, respectively. Correspondingly, the NCC values were 0.707, 0.872, and 0.999. Similarly, the stego audio with hidden index value has average HER of 0.394944, 0.235005, and 0.024533, with corresponding NCC values as 0.541275, 0.752979, and 0.970612. The stego audio with hidden RLE count and values has average HER and NCC values of 0.3937, 0.213304, 0.006237, 0.554481, 0.996419, and 0.763709, respectively.

These results demonstrate that the proposed BCS-3 algorithm and the enhanced LSB algorithm preserve the quality of stego audio files, with audio quality decreasing slightly as noise increases. The stego audio files are transmitted sequentially, and both the sender and receiver know the order and hidden content of the files. Under these controlled conditions, decoding proceeds without alignment issues, and the overhead of managing multiple files is minimal.

While the proposed BCS-3 algorithm preserves the quality of the stego audio under adversarial conditions, the robustness of the recovered HCD remains challenging [35]. The key reason for preserving perceptual quality is that only 3 bits out of 24 bits in each audio sample are affected by the embedding process. However, while decoding works perfectly under controlled conditions where the sender and receiver know the order of the files, in real-world noisy environments, synchronization among multiple stego audio files may be challenging, potentially causing misalignment during decoding and affecting data recovery. Future work will focus on improving overall robustness to address both hidden data recovery and multi-file synchronization challenges.

4.7.4 Subjective auditory evaluations

The Mean Opinion Score (MOS) was employed as the subjective evaluation metric for the BCS-3 algorithm. MOS provides a numerical measure of audio quality on a scale from 1 (very Annoying) to 5 (Imperceptible), as shown in Table 10 [41]. The final score is obtained as the arithmetic mean of all listener ratings.

In this study, a total of 25 pairs of audio files (cover and stego) were evaluated. Listening tests involved 30 participants from diverse backgrounds, including audio engineers, industry professionals, academicians, researchers, and students. Each participant was asked to carefully evaluate paired audio signals (cover and stego) and assign quality ratings.

The results, presented in Table 10, indicate consistently high MOS values across all test cases, confirming the strong perceptual transparency of the proposed algorithm. The high MOS values are attributed to the fact that three characters (24 bits) are compressed into 8 bits using the proposed 24×8 compression algorithm. These 8 bits are then embedded into 24-bit audio samples, resulting in only a 3-bit modification per sample.

Figure 19 provides a comprehensive comparison chart outlining performance metrics between the existing and proposed algorithms. The chart employs the y-axis to represent performance metric values for both the proposed and existing algorithms, and the x-axis denotes references and names for the performance metrics. The proposed 24x8 compression algorithm aims to reduce three characters into a single character, which enhances the embedding capacity for audio steganographic communication.

The proposed BCS-3 algorithm aims to ensure that the length of hidden data in bytes and the differing bits between the cover and stego audio files are equal. This approach enables predictable bit errors during the embedding process, resulting in 15000 error bits for 15000 bytes of healthcare data. This helps to predict bit errors in advance and establishes a consistent standard. Additionally, the algorithm maintains a standard by embedding three characters (compressed into one character) with a three-bit difference in the corresponding audio samples.

This results in a well-maintained trade-off between embedding capacity and the integrity of the audio file. Furthermore, the total number of audio samples required for embedding should be one-third the length of the embedded data, as demonstrated in Table 8 and Figure 20. The BCS-3 algorithm embeds data in audio samples with a 3-bit difference, reducing the iterations needed to select audio samples for data embedding.

The proposed method introduces innovative algorithms like I-OVA and D-RVA for secure encryption and decryption, the Merging Algorithm (MA) for optimizing data representation, the 24x8 Compression Algorithm for compressing three characters to one character, BCS-3 and BCR-3 for systematic embedding and retrieval, and an enhanced LSB algorithm for embedding probability distribution and indexing audio samples securely. Together, these algorithms ensure double-layer security

with efficient utilization of audio samples by embedding three characters of 24 bits as 8-bit encoded values in an audio sample of 24 bits. This approach results in an error bit of 3, with each character contributing one error bit. This results in a relationship where the HCD length in bytes corresponds to error bits, resulting in a 3:1 utilization ratio of audio samples.

Addressed novelty and research gap

The proposed work addresses a research gap in telemedicine by overcoming the limitations of existing audio steganography techniques for secure healthcare data communication. Traditional LSB algorithms exhibit inefficiencies in embedding capacity, inconsistent error patterns, and a lack of integrated security mechanisms. Image steganography is commonly used in telemedicine, while audio steganography is less explored for secure healthcare data communication.

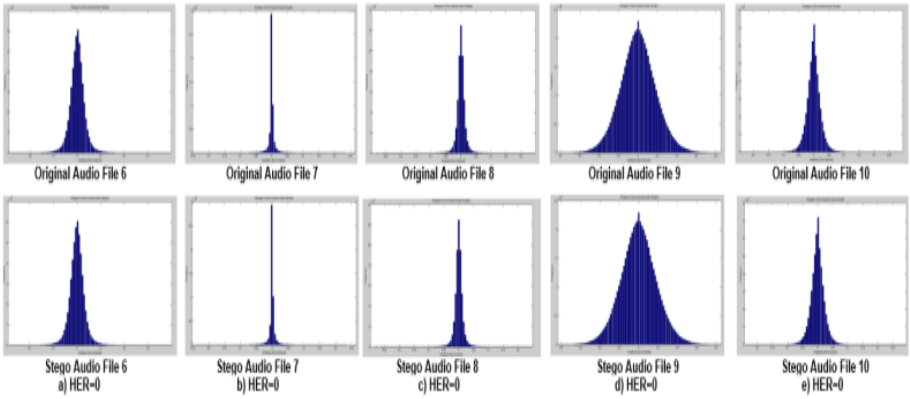


Figure 18: Histograms for various cover audios and stego audios using 15000B Healthcare data

Table 10: MOS grading scale and scores for HCD payloads (3000–15000 bytes) across various audio files

MOS Grade	1	2	3	4	5
Remarks	Very Annoying	Annoying	Slightly Annoying	Perceptible Not Annoying	Imperceptible
Source audio	A6	A7	A8	A9	A10
MOS Grade	5	5	5	5	5

Table 11: Comparison of DR for the Fourth First Moments and HER for various cover audios using 15000B of concealed HCD

Audio File	Difference Ratio (DR)				HER
	(μ)	(σ^2)	(sk)	(k)	
[32]	0.0799 ▲99.77%	0.0008 ▲99.99%	0.0018 ▲99.96%	0.0028 ▲99.99%	0.1278 ▲100%
[33]	0.2304 ▲99.92%	0 ≈	0.0004 ▲99.83%	0.0005 ▲99.99%	1.2274E-04 ▲100%
Proposed	**1.8331E-04**	**2.5759E-09**	**6.7953E-07**	**3.5282E-09**	**0**

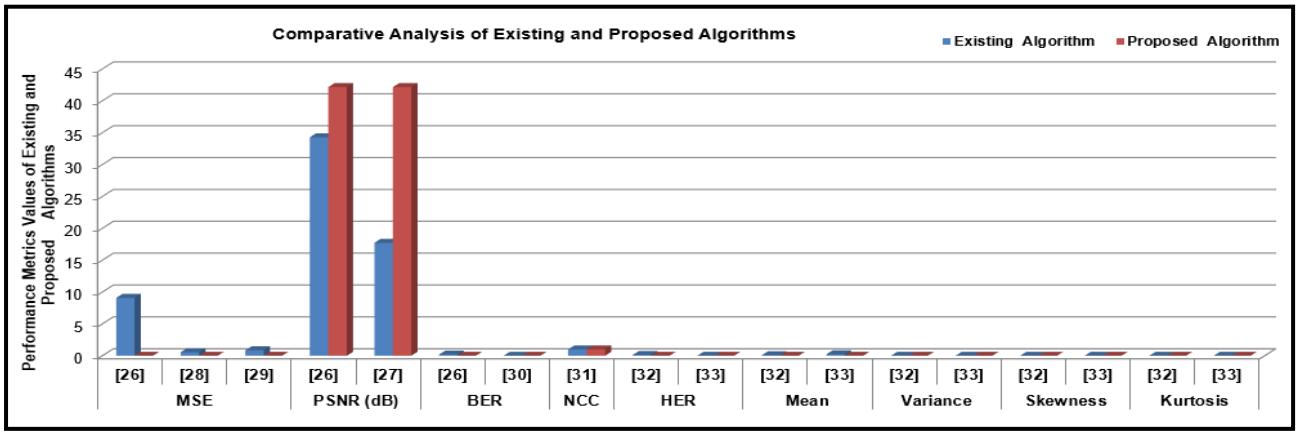


Figure 19: Comparative analysis of proposed and existing algorithms

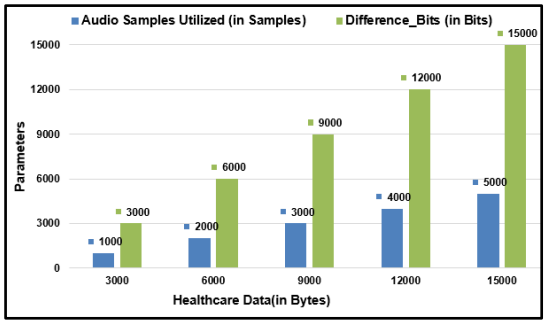


Figure 20: Influence of parameters of audio files versus hiding capacity with various text and audio files

This work addresses these challenges by introducing a robust and efficient framework designed for telemedicine applications. The framework proposes a novel method for communicating compressed healthcare data in telemedicine by integrating cryptography and steganography.

B The significance of the proposed work

The integration of cryptography and steganography provides double-layer security for healthcare data.

The proposed method compresses 15000B of healthcare data into 5000B of encoded values, achieving a threefold reduction and improving communication efficiency.

Efficient utilization of audio samples enhances the embedding capacity while maintaining audio quality.

The approach ensures consistent and predictable error bit patterns, enhancing the reliability of the embedding process.

The framework communicates compressed HCD, making it adaptable to various telemedicine scenarios in diverse network environments.

The overall work establishes a relationship between the HCD size, the audio samples utilized, and the error bits introduced in the embedding of HCD. This insight allows for systematic planning of the entire healthcare communication process, minimizing the potential for steganalysis and enhancing the security of the transmitted data.

5 Conclusion and future work

This paper proposes a model for secure healthcare data transmission using audio steganography for telemedicine. The 24x8 compression algorithm optimally uses audio samples to increase embedding capacity by compressing three characters into one. The model utilizes three audio files. The first audio file transmits the run-length encoding total count, individual count values, and corresponding data values using the least significant bit (LSB) algorithm with Ks as reference values. The metrics supporting the algorithm are the PSNR: 30.1417 dB and BER: 0.001118. The second audio file transmits 8-bit healthcare data (HCD) as encoded values using the BCS-3 algorithm. The compressed 1 byte (8 bits) of healthcare data from 3 bytes (24 bits) is further embedded within three bytes of the audio file. The average MSE and PSNR for the BCS-3 algorithm are 6.5297E-14 and 42.1480 dB, respectively. The BCR-3 algorithm retrieves the hidden HCD, with its efficiency indicated by a BER of 7.0165E-5 and an NCC value of 0.999999. The third audio file transmits the index values of stego audio samples containing HCD as encoded values, again using the LSB algorithm with Ks as the reference. The efficiency of algorithms is proved with the metrics PSNR: 36.5499 dB and BER: 0.000258.

Healthcare data (HCD) is encrypted and decrypted using the I-OVA and D-RVA algorithms to ensure data security. When transmitting 15000B of HCD, the algorithms achieve a throughput of 8592.74 KB/s. The 24x8 compression algorithm reduces the audio samples required for embedding healthcare data (HCD), resulting in 5000 samples for 15000B of data compared to the least significant bit algorithm.

The prevalence of 3 differing bits in the BCS-3 algorithm reduces the complexity of the process and the computational iterations in computing audio samples for embedding healthcare data (HCD).

The comparative methodology of the proposed BCS-3 algorithm resulted in identical values. With an HCD comprising 15000B (15000 characters), the audio samples utilized for embedding are 5000 samples. The difference in bits between the cover and stego audio files containing concealed data of 15000B is 15000 bits. Future research could explore real-time transmission of healthcare data using an audio steganographic system integrated with IoT for telemedicine applications.

Acknowledgement

The authors thank the anonymous reviewers for their insightful feedback. Their suggestions significantly improve the quality of this research article.

Data availability statement

Audio Files: https://mixkit.co/free-sound-effects/
HCD:https://www.kaggle.com/datasets/xuehaihe/covid-dialogue-dataset?select=COVID-Dialogue-Dataset-English.txt

References

[1] Rumi Chunara, Yuan Zhao, Ji Chen, Katharine Lawrence, Paul A Testa, Oded Nov, and Devin M Mann.Telemedicine and healthcare disparities: a cohort study in a large healthcare system in New York City during COVID-19.Journal of the American Medical Informatics Association. 28(1):33-41,2021. https://doi.org/10.1093/jamia/ocaa217

[2] Abid Haleem, Mohd Javaid, Ravi Pratap Singh, and Rajiv Suman. Telemedicine for healthcare: Capabilities, features, barriers, and applications. Sensors International,2(100117):1-12,2021. https://doi.org/10.1016/j.sintl.2021.100117

[3] A. S. Albahri, Jwan K. Alwan, Zahraa K. Taha, Sura F. Ismail, Rula A.Hamid, A. A. Zaidan, O. S. Albahri , B. B. Zaidan, A. H. Alamoodi, and M. A. Alsalem .IoT-based telemedicine for disease prevention and health promotion: State-of-the-Art.Journal of

Network and Computer Applications.173(102873),2020. https://doi.org/10.1016/j.jnca.2020.102873

[4] Malak Alkhudaydi, and Adnan Gutub. Securing Data via Cryptography and Arabic Text Steganography.SN Computer Science .2(46):1-18, 2021. https://doi.org/10.1007/s42979-020-00438-y

[5] Hamouda B.E.H.H. Comparative Study of Different Cryptographic Algorithms. Journal of Information Security.11(3):138-148,2020. https://doi.org/10.4236/jis.2020.113009

[6] Abdalbasit Mohammed Qadir, and Nurhayat Varol. A Review Paper on Cryptography .2019 7th International Symposium on Digital Forensics and Security (ISDFS).IEEE, 2019. https://doi.org/10.1109/isdfs.2019.8757514

[7] Sazeen T. Abdulrazzaq, Mohammed M. Siddeq, and Marcos A. Rodrigues. A Novel Steganography Approach for Audio Files.SN Computer Science. 1(97):1-13,2020. https://doi.org/10.1007/s42979-020-0080-2

[8] Hrishikesh Dutta, Rohan Kumar Das, Sukumar Nandi, and S. R. Mahadeva Prasanna. An Overview of Digital Audio Steganography. IETE Technical Review.37(6):632-650,2019. https://doi.org/10.1080/02564602.2019.1699454

[9] Dingwei Tan, Yuliang Lu, Xuehu Yan and Xiaoping Wang. A Simple Review of Audio Steganography. 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC).1409-1413,2019. https://doi.org/10.1109/ITNEC.2019.8729476

[10] Ujala Razaq, Xu Lizhong, Changli Li, and Muhammad Usman.Evolution and Advancement of Arithmetic Coding Over Four Decades. Open Journal of Science and Technology. 3(3):194-236, 2020. https://www.researchgate.net/publication/346441614_EVOLUTION_AND_ADVANCEMENT_OF_ARITHMETIC_CODING_OVER_FOUR_DECADES

[11] Uthayakumar Jayasankar, Vengattaraman Thirumal, and Dhavachelvan Ponnurangam.A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications. Journal of King Saud University-Computer and Information Sciences.33(2):119-140,2021. https://doi.org/10.1016/j.jksuci.2018.05.006

[12] Tonny Hidayat, Mohd Hafiz Zakaria, and Ahmad Naim Che Pee .Comparison of Lossless Compression Schemes for WAV Audio Data 16-Bit between Huffman and Coding Arithmetic.International Journal of Simulation: Systems, Science & Technology .19(6):36.1-36.7,2019. https://doi.org/10.5013/ijssst.a.19.06.36

[13] A. A. Alsabhany, A. H. Ali, F. Ridzuan, A. H. Azni, and M. R. Mokhtar.Digital audio steganography: Systematic review, classification, and analysis of the current state of the art.Computer Science Review.(38).100316:1-27,2020. https://doi.org/10.1016/j.cosrev.2020.100316

[14] Indy Haverkamp, and Dipti K. Sarmah, Evaluating the merits and constraints of cryptography-steganography fusion: a systematic analysis.International Journal of Information Security.23:2607-2635,2024; https://doi.org/10.1007/s10207-024-00853-9

[15] Enas Wahab Abood, Abdulhssein M. Abdullah , Mustafa A. Al Sibahee , Zaid Ameen Abduljabbar, Vincent Omollo Nyangaresi , Saad Ahmad Ali Kalafy, and Mudhafar Jalil Jassim Ghrabta. Audio steganography with enhanced LSB method for securing encrypted text with bit cycling.Bulletin of Electrical Engineering and Informatics.11(1):185-194,2022. https://doi.org/10.11591/eei.v11i1.3279

[16] Dilip Kumar Sharma, Ningthoujam Chidananda Singh, Daneshwari A Noola, Amala Nirmal Doss , and Janaki Sivakumar. A review on various cryptographic techniques & algorithms. Materials Today:Proceedings.51(1):104-109,2022. https://doi.org/10.1016/j.matpr.2021.04.583

[17] Mohanad Sameer Jabbar, and Samer Saeed Issa. A crypto-steganography healthcare management: towards a secure communication channel for data COVID-19 updating. Indonesian Journal of Electrical Engineering and Computer Science.29(2):1102-1112,2023. https://doi.org/10.11591/ijeecs.v29.i2.pp1102-1112

[18] Reihane Saniei, and Karim Faez. The Security of Arithmetic Compression Based Text Steganography Method. International Journal of Electrical and Computer Engineering (IJECE).3(6):797-804,2013. https://ijece.iaescore.com/index.php/IJECE/article/view/5425/4932

[19] Wafaa Al-Chaab, Zaid Ameen Abduljabbar, Enas Wahab Abood, Vincent Omollo Nyangaresi, Hussein M. Mohammed, and Junchao Ma. Secure and Low-Complexity Medical Image Exchange Based on Compressive Sensing and LSB Audio Steganography. Informatica, 47(6):65–74, 2023. https://doi.org/10.31449/inf.v47i6.4628

[20] Nathaniel Fout, and Kwan-Liu Ma. An Adaptive Prediction-Based Approach to Lossless Compression of Floating-Point Volume Data. IEEE Transactions on Visualization and Computer Graphics,18(12):2295-2304,2012. https://doi.org/10.1109/TVCG.2012.194

[21] Mostafa A. Ahmad, Mourad Elloumi , Ahmed H. Samak , Ali M. Al-Sharafi, Ali Alqazzaz , Monir Abdullah Kaid ,and Costas Iliopoulos. Hiding patients' medical reports using an enhanced wavelet steganography algorithm in DICOM images. Alexandria Engineering Journal.61(12): 10577–10592,2022. https://doi.org/10.1016/j.aej.2022.03.056

[22] Anne Y. Ning, Claudia I. Cabrera, and Brian D Anza. Telemedicine in Otolaryngology: A Systematic Review of Image Quality, Diagnostic Concordance, and Patient and Provider Satisfaction. Annals of Otology, Rhinology & Laryngology.130(2):195–204,2020. https://doi.org/10.1177/0003489420939590

[23] Medapati Venkata Manga Naga Sravan, and K Venkata Rao.5G-Optimized Deep Learning Framework for Real-Time Multilingual Speech-to-Speech Translation in Telemedicine Systems. Informatica, 49 (2):279–298, 2025. https://doi.org/10.31449/inf.v49i2.7826

[24] Mahmoud Magdy, Khalid M. Hosny, Neveen I. Ghali, and Said Ghoniemy. Security of medical images for telemedicine: a systematic review. Multimedia Tools and Applications.81:25101–25145, 2022. https://doi.org/10.1007/s11042-022-11956-7

[25] K Revathi, S. Kaja Mohideen, and Latha Tamilselvan. Efficient Audio Steganography Using Bit Comparison and Substitution Algorithms for Improved Embedding Capacity. Informatica, 49 (21)**:** 179–198, 2025. https://doi.org/10.31449/inf.v49i21.6957

[26] Bambang Harjito, Beni Sulistyarso, and Esti Suryani. Audio steganography using two LSB modification and RSA for security data transmission. Journal of Telecommunication, Electronic and Computer Engineering (JTEC),10(2–4):107–111,2018. https://jtec.utem.edu.my/jtec/article/view/4326/3173

[27] M. Parthasarathi, and T. Shreekala.Secured Data Hiding in Audio Files Using Audio Steganography Algorithm.International Journal of Pure and Applied Mathematics, 116(21):619–628, 2017. https://acadpubl.eu/jsi/2017-116-13-22/articles/21/79.pdf

[28] Ratul Chowdhury, Debnath Bhattacharyya, Samir Kumar Bandyopadhyay, and Tai-hoon Kim .A View on LSB Based Audio Steganography.International Journal of Security and Its Applications ,10(2),51-62, 2016. https://doi.org/10.14257/ijsia.2016.10.2.05

[29] Hemalatha S, U Dinesh Acharya, Renuka A, Deepthi S, and Jyothi Upadhya K.Audio steganography in discrete wavelet transform domain. International Journal of Applied Engineering Research. 10(16):36639–36644,2015. https://www.ripublication.com/ijaer10/ijaerv10n16_85.pdf

[30] Vipul Sharma, and Ravinder Thakur.LSB Modification based Audio Steganography using Trusted Third Party Key Indexing Method.IEEE 2015 Third International Conference on Image Information Processing(ICIIP), 403–406, 2015. https://doi.org/10.1109/ICIIP.2015.7414805

[31] Ahmed Hussain Ali, Mohd Rosmadi Mokhtar, and Loay Edwar George .Enhancing the hiding capacity of audio steganography based on block mapping. Journal of Theoretical and Applied Information Technology.95(7):1441–1448,2017. https://www.jatit.org/volumes/Vol95No7/13Vol95No7.pdf

[32] Ahmed Hussain Ali , Loay Edwar George, A. A. Zaidan, and Mohd Rosmadi Mokhtar.High capacity, transparent and secure audio steganography model based on fractal coding and chaotic map in temporal domain.Multimedia Tools and Applications, 77(23):31487-31516,2018. https://doi.org/10.1007/s11042-018-6213-0

[33] Haider Ismael Shahadi, Razali Jidin, and Wong Hung Way.Lossless audio steganography based on lifting wavelet transform and dynamic stego key.Indian Journal of Science and Technology,7(3):323–334, 2014. https://doi.org/10.17485/ijst/2014/v7i3.14

[34] Xubao Zhang. Designs, experiments, and applications of multichannel structures for hearing aids. EJECE, European Journal of Electrical Engineering and Computer Science. 5(4): 46–55,2021. https://doi.org/10.24018/ejece.2021.5.4.347

[35] Mahmoud M. Mahmoud, and Huwaida T. Elshoush. Enhancing LSB Using Binary Message Size Encoding for High Capacity, Transparent and Secure Audio Steganography-An Innovative Approach.IEEE Access.10,29954-29971,2022. https://doi.org/10.1109/access.2022.3155146

[36] Chakravorty, Subrato and He, Xuehai and Yang, Xingyi and Xie, Pengtao. 2020. Chakravorty2020COVIDDialogueDatasetEnglish.COVID-Dialogue-Dataset-English: an English medical dialogue dataset about COVID-19 and other types of, pneumonia. https://github.com/UCSD-AI4H/COVID-Dialogue.2020. https://www.kaggle.com/datasets/xuehaihe/covid-dialogue-dataset?select=COVID-Dialogue-Dataset-English.txt

[37] Thoyazan Sultan Algaradi, and B. Rama. A New Encryption Scheme For Performance Improvement In Big Data Environment Using Mapreduce. Journal of Engineering Science and Technology.16(5):3772 – 3791,2021.https://jestec.taylors.edu.my/Vol%2016%20Issue%205%20October%202021/16_5_11.pdf

[38] R. Ramya Devi, and V. Vijaya Chamundeeswari. Triple DES: Privacy Preserving in Big Data Healthcare. International Journal of Parallel Programming.48:515-533, 2018. https://doi.org/10.1007/s10766-018-0592-8

[39] Muhammad Yunus, Intan Sulistyaningrum Sakkinah, Ulfa Emi Rahmawati, Atma Deharja , and Maya Weka Santi. File Security Design in Electronic Health Record (EHR) System with Triple DES Algorithm (3DES) at Jember Family Health Home Clinic.1(1):1-8,2023. https://doi.org/10.47134/ijhis.v1i1.2

[40] Ahmed A. Alsabhany, Farida Ridzuan, and A. H. Azni, The Progressive Multilevel Embedding Method for Audio Steganography, Journal of Physics: Conference Series, 1551 012011,2020. https://doi.org/10.1088/1742-6596/1551/1/012011

[41] Dipankar Pal, and Nabin Ghoshal. A Robust Audio Steganographic Scheme in Time Domain (RASSTD). International Journal of Computer Applications (0975 8887), 80(15):1-8, 2013. https://doi.org/10.5120/13934-1803

.