International Journal of Advanced AI Applications

Volume2,Issue5,May.2026

Online ISSN 3104-9338

Print ISSN 3104-932X

Hong Kong Dawn Clarity Press Limited

http://www.dawnclarity.press/index.php/ijaaa

TABLE OF CONTENTS

Online ISSN: 3104-9338
Print ISSN: 3104-932X

Research on the Double-Edged Sword Effects of Artificial Intelligence on Enterprise Human Resource Management

Dahao Li*

Doctoral candidate in Business Administration, Al-Farabi Kazakh National University International Business School Almaty, Kazakhstan

Received: March 26, 2026

Revised: March 27, 2026

Accepted: March 28, 2026

Published online: April 11, 2026

To appear in: *International Journal of Advanced AI Applications,* Vol. 2, No. 5 (May 2026)

* Corresponding Author: Dahao Li (li_dakhao2@live.kaznu.kz)

Abstract. Artificial intelligence has been increasingly embedded into the full-cycle practices of enterprise Human Resource Management (HRM), forming a typical technological empowerment scenario with coexisting opportunities and predicaments. Based on the Technology Acceptance Model, Sociotechnical Systems Theory, and Organizational Justice Theory, this paper constructs an integrated theoretical framework to systematically dissect the double-edged sword effects of AI application in HRM. The positive dimensions are refined into four core paths: efficiency improvement in recruitment and selection, objectivity enhancement in performance management, personalized empowerment of employee development, and data-driven upgrading of strategic decision-making. The negative dimensions are summarized as four prominent risks: algorithmic bias and discriminatory infringement, employee privacy leakage and supervision anxiety, dehumanization of employment relations, and multi-level organizational resistance. Furthermore, this study explores the differential impact mechanisms of AI outcomes from four dimensions: AI system design, implementation procedures, organizational contextual factors, and individual employee differences. On this basis, targeted coping strategies are proposed from ethical design, transparent governance, employee participation, and human-AI hybrid collaboration, aiming to provide theoretical references and practical paths for enterprises to maximize the positive effects of AI and control potential risks in HRM practices.

Keywords: *Artificial Intelligence; Human Resource Management; Double-edged Sword Effects; Algorithmic Management; Ethical AI; Human-AI Collaboration*

1. Introduction

Driven by digital transformation and technological iteration, artificial intelligence has broken through the technical boundary of enterprise management and penetrated into the whole process of human resource management, covering recruitment screening, performance evaluation, training and development, turnover prediction, labor relations coordination and other core modules [5] [10]. Global technology giants such as IBM, Oracle, and SAP have launched integrated AI-HR platforms, which have realized the intelligent transformation from traditional manual operation to data-driven decision-making, and gradually changed the management logic and operation mode of organizational human resources. According to industry statistics, more than 68% of large multinational enterprises have adopted AI-based HR tools, and the market size of global AI HR applications is expected to exceed 35 billion US dollars by 2030, showing a rapid growth trend.

However, the integration of AI and HRM presents a complex and contradictory picture in practical application. Optimists believe that AI can liberate HR professionals from heavy administrative work, improve management efficiency and decision-making accuracy, and realize the strategic transformation of human resource management [2]. Pessimists point out that the widespread use of algorithmic decision-making has triggered a series of ethical and management crises, such as algorithmic discrimination, excessive supervision, humanistic indifference, and employee resistance, which seriously impact the fairness and sustainability of employment relations [9]. This dual effect determines that AI in the field of human resource management is not a simple technological tool, but a double-edged sword that can both empower management and induce risks.

The double-edged sword effect refers to the phenomenon that a single factor can produce both beneficial and harmful results, and its final performance depends on the combination of contextual factors and operation methods. In the context of digital HRM, clarifying the positive value, negative risks, influencing mechanisms and governance strategies of AI application is not only a theoretical frontier issue, but also a practical problem that enterprises must solve in digital transformation. At present, academic research has carried out exploratory discussions on AI applications in single HR modules such as recruitment [11] [13], performance management [8], and talent retention, and has also paid attention to the ethical risks and organizational responses of algorithmic decision-making [2] [5]. However, the existing literature lacks a systematic theoretical framework that integrates positive effects, negative effects, influencing mechanisms and governance countermeasures, and cannot fully explain the formation logic and

regulation path of the double-edged sword effect of AI in HRM.

To fill this research gap, this study takes the double-edged sword effect of AI in enterprise HRM as the core research theme, and constructs a comprehensive analysis framework based on three classical theories. The research objectives of this paper are as follows: first, to clarify the specific manifestations and formation paths of the positive effects of AI on HRM; second, to identify the types and potential hazards of negative effects; third, to reveal the key factors and action mechanisms that determine the differential outcomes of AI application; fourth, to put forward targeted governance strategies to amplify advantages and avoid disadvantages. The theoretical contributions of this study are reflected in three aspects: integrating multi-theoretical perspectives to enrich the research system of digital HRM; systematically deconstructing the double-edged sword effect to clarify the dual logic of AI empowerment and risk; proposing a multi-dimensional governance framework to provide theoretical guidance for practical application. In practice, this study can help enterprises rationally view the role of AI, optimize the design and implementation of AI-HR systems, balance technological efficiency and humanistic care, and promote the sustainable development of human resource management in the digital era.

2. Theoretical Foundations

2.1. Technology Acceptance Model

The Technology Acceptance Model (TAM) was proposed by Davis (1989) to explain and predict users' behavioral intention to adopt new information technologies. The core logic of the model is that users' acceptance behavior is determined by two key perceptions: perceived usefulness and perceived ease of use. Perceived usefulness refers to the degree to which individuals believe that using a certain technology can improve their work performance; perceived ease of use refers to the degree to which individuals believe that using a certain technology does not require excessive effort. On this basis, Venkatesh et al. (2003) expanded the model into a unified theory of acceptance and use of technology, incorporating social influence and facilitating conditions into the analysis framework, which enhanced the explanatory power of the model in organizational scenarios.

In the scenario of AI-enabled HRM, the Technology Acceptance Model provides a micro-level analytical perspective for understanding employees' and managers' responses to AI systems. When HR practitioners and employees perceive that AI systems can effectively improve work efficiency, reduce operational burden, and enhance decision-making quality

(high perceived usefulness), and the system operation is simple and easy to understand (high perceived ease of use), they will take a positive attitude towards AI application and actively participate in system use [14]. Conversely, if users believe that AI systems threaten their work autonomy, replace their core responsibilities, or have complex operation and opaque logic (low perceived usefulness and ease of use), they will produce resistance and rejection, which will lead to the failure of AI implementation. This theoretical perspective emphasizes that the effectiveness of AI application does not solely depend on the technical level of the system, but is closely related to users' subjective perception and psychological acceptance, which lays a foundation for analyzing the micro-foundation of the double-edged sword effect.

2.2. Sociotechnical Systems Theory

Sociotechnical Systems Theory was founded by Trist and Bamforth (1951) in the study of coal mining production systems. The core view of the theory is that any organization is a complex system composed of interactive social subsystems and technical subsystems; the optimal operation of the organization requires the coordinated adaptation of the two subsystems, and the introduction of technical systems must fully consider the matching of social factors such as organizational structure, employee relations, and work design. Bostrom and Heinen (1977) further applied this theory to the field of information system management, pointing out that the failure of most management information systems is not due to technical defects, but due to the neglect of social subsystem factors, resulting in the disconnection between technology and people.

In the context of AI integration into HRM, Sociotechnical Systems Theory provides a holistic analytical framework for balancing technological efficiency and humanistic care. AI, as a technical subsystem, has the advantages of high efficiency, objectivity, and data processing, but it must be matched with the social subsystem including organizational culture, employee attitudes, power relations, and work design [1]. When AI systems are designed to assist human work rather than replace human judgment, and fully consider the social attributes and emotional needs of employees, the technical and social subsystems can form a synergistic effect, and the positive effects of AI will be highlighted. If enterprises only pursue technological efficiency and ignore the social impact of AI implementation, such as damaging interpersonal trust, destroying employment relations, and weakening organizational cohesion, it will lead to system imbalance and trigger a series of negative effects. This theory reveals the systemic logic of AI's double-edged sword effect and provides a theoretical basis for constructing a human-AI hybrid management model.

2.3. Organizational Justice Theory

Organizational Justice Theory focuses on employees' perception of fairness in organizational decision-making and its impact on individual attitudes and behaviors. Greenberg (1987) divided organizational justice into distributive justice and procedural justice; distributive justice refers to the fairness of the distribution of organizational resources and outcomes, and procedural justice refers to the fairness of the decision-making process. Colquitt (2001) further integrated the theoretical framework and added interactional justice, which refers to the fairness of interpersonal treatment received by individuals in the process of organizational management. The three dimensions of justice jointly affect employees' trust, satisfaction and organizational commitment.

In the application of AI in HRM, Organizational Justice Theory is a key theoretical tool for explaining employees' psychological perception and behavioral response to algorithmic decision-making. AI systems participate in key HR decisions such as recruitment, promotion, performance appraisal and termination, and employees' perception of the fairness of these algorithmic decisions directly determines their acceptance of AI [2]. If employees believe that AI decisions are based on fair data and transparent processes (procedural justice), the results are reasonable and unbiased (distributive justice), and the system fully respects individual dignity (interactional justice), they will recognize the application of AI and form positive organizational behavior. On the contrary, if the algorithm is a "black box", the decision results are discriminatory, and the management process lacks humanistic care, employees will perceive injustice, resulting in resistance, distrust and even departure, which amplifies the negative effects of AI. This theory clarifies the psychological mechanism of employees' response to AI and provides a theoretical basis for the ethical design and transparent governance of AI-HR systems.

2.4. Theoretical Integration Framework

Based on the above three theories, this study constructs an integrated theoretical framework for the double-edged sword effect of AI on HRM. The Technology Acceptance Model explains the micro-psychological mechanism of users' acceptance of AI, which is the micro-foundation of AI effect realization; Sociotechnical Systems Theory emphasizes the synergistic matching of technical and social subsystems, which is the systemic guarantee of AI effect; Organizational Justice Theory reveals the fairness perception mechanism of employees to algorithmic decisions, which is the psychological premise of AI effect. The three theories complement each other: TAM focuses on individual perception, Sociotechnical Systems Theory focuses on

organizational system matching, and Organizational Justice Theory focuses on fairness perception. Together, they constitute a multi-level, multi-dimensional theoretical system to explain why AI produces both positive and negative effects in HRM, and provide a logical basis for analyzing influencing factors and proposing governance strategies.

3. Positive Effects of AI on HRM

3.1. Enhanced Efficiency in Recruitment and Selection

AI has realized the intelligent transformation of recruitment and selection, and greatly improved the operational efficiency of the recruitment link. Traditional recruitment relies on manual resume screening, which has problems such as long time-consuming, high labor cost and large subjective error. AI-powered automated resume screening tools can use natural language processing and machine learning technologies to quickly identify keywords, match job requirements, and screen qualified candidates from massive resumes. Relevant data shows that AI resume screening can shorten the average recruitment cycle by 55%-60%, and process thousands of resumes in a few minutes, which is dozens of times more efficient than manual screening.

Intelligent interview assistants and chatbots further optimize the recruitment process. AI chatbots can conduct preliminary communication with candidates 24 hours a day, answer frequently asked questions about positions, enterprises and welfare, schedule interviews automatically, and feed back information to HR in real time [11]. Video interview AI systems can analyze candidates' expressions, language logic and behavioral characteristics through computer vision and speech recognition technology, providing auxiliary evaluation references for recruiters. These applications reduce the repetitive work of HR professionals, enabling them to focus on high-value activities such as candidate relationship maintenance and cultural fit assessment.

In terms of cost control, AI recruitment tools have significant advantages. Compared with traditional recruitment methods, AI-enabled recruitment can reduce recruitment costs by 20%-30%, mainly reflected in the reduction of intermediary agency fees, administrative labor costs and time costs [13]. For enterprises, shortened recruitment cycles mean that vacant positions are filled faster, reducing the production and operational losses caused by talent gaps; for candidates, AI recruitment provides faster response speed and more standardized communication, improving the candidate experience. The efficiency improvement of recruitment and selection is the most direct positive effect of AI on HRM, laying a foundation

for enterprises to quickly obtain high-quality talents.

3.2. Improved Objectivity in Performance Management

AI applications effectively alleviate the subjective bias in traditional performance management and improve the objectivity and fairness of evaluation. Traditional performance appraisal is highly dependent on managers' subjective judgment, which is prone to recency bias, halo effect, central tendency and interpersonal favoritism, leading to the deviation of evaluation results from actual performance [8]. AI performance management systems collect objective data from multiple sources, including project completion rate, work quality indicators, customer satisfaction, team collaboration efficiency and other behavioral and result data, and conduct comprehensive evaluation through algorithm analysis, reducing the interference of human subjective factors.

The real-time feedback function of AI optimizes the dynamic management of performance. Traditional performance appraisal mostly adopts annual or semi-annual periodic evaluation, with delayed feedback and poor timeliness. AI systems can monitor employees' work data in real time, provide instant feedback on work defects and improvement directions, help employees adjust work strategies in a timely manner and achieve continuous performance improvement [5]. For example, sales employees can obtain real-time data on customer conversion and performance completion through AI systems, and adjust sales strategies according to feedback; R&D personnel can track project progress and task completion through AI tools to ensure the smooth progress of projects.

In addition, AI performance analytics supports fair and transparent evaluation results. The system generates quantitative evaluation reports based on objective data, and managers can refer to algorithmic suggestions to make evaluation decisions, reducing the impact of personal preferences. Some enterprises have applied AI performance systems to achieve multi-dimensional and full-cycle evaluation, making performance appraisal more standardized and credible, enhancing employees' recognition of performance management, and promoting the formation of a fair competition organizational atmosphere.

3.3. Personalized Employee Development

AI empowers employee training and career development to achieve personalized and precise management at scale. Traditional training modes adopt a one-size-fits-all approach, which cannot match the individual differences in employees' skill gaps, learning abilities and career plans, resulting in low training efficiency and poor effectiveness. AI-powered learning

platforms can analyze employees' skill attributes, work performance, learning habits and career aspirations through big data technology, and recommend personalized training courses, learning paths and development plans [8]. For example, for new employees, the system can provide targeted induction training; for senior employees, it can recommend advanced management or technical courses to meet their career promotion needs.

AI career pathing tools provide personalized development guidance for employees. Based on the matching analysis of employees' personal abilities and organizational talent needs, the system can predict employees' career development potential, recommend suitable positions and promotion paths, and formulate skill improvement plans [10]. This personalized development service makes employees feel the organization's attention and investment in their growth, enhances employees' sense of belonging and work engagement, and reduces the turnover risk of high-potential talents. For enterprises, personalized employee development helps to accurately fill the skill gap, optimize the talent structure, and realize the coordinated development of individual growth and organizational goals.

AI skill gap analysis provides a basis for proactive talent development. The system can compare the current skill inventory of the workforce with the skill requirements required by the enterprise's future development strategy, identify the key skill gaps in the organization, and help the human resource department carry out targeted training and talent introduction in advance [5]. This predictive management mode changes the passive response of traditional HRM, realizes proactive talent layout, and provides talent guarantee for enterprises' long-term development.

3.4. Data-Driven Strategic Decision-Making

AI promotes the transformation of human resource management from administrative execution to strategic decision-making, and realizes data-driven scientific management. Traditional HRM relies on experience and qualitative judgment, lacking quantitative support for strategic decisions such as workforce planning, talent retention and organizational structure adjustment. AI-enabled people analytics platforms can mine and analyze massive workforce data, identify potential laws and trends, and provide quantitative decision-making support for strategic human resource management [8].

Predictive analytics is a core application of AI in strategic HR decision-making. For example, AI turnover prediction models can analyze the key factors affecting employees' departure intention by integrating demographic characteristics, work performance, salary level,

organizational atmosphere and other data, issue early warning signals for high-risk turnover employees, and help managers take intervention measures in advance [10]. Workforce planning AI tools can simulate the impact of different recruitment, training and mobility strategies on the future workforce scale, structure and cost, and optimize the workforce allocation plan to match the enterprise's business strategy.

AI improves the strategic value and organizational status of the human resource department. By providing accurate data insights and decision support, HR can participate in the enterprise's top-level strategy formulation, and transform from a functional support department into a strategic partner. For enterprise executives, AI-HR analytics provides a visual presentation of human capital value and risk, helping them accurately grasp the status of organizational talents and make scientific business decisions. The strategic upgrading of HRM driven by AI is an important positive effect, which enhances the core competitiveness of enterprises in the digital era.

4. Negative Effects of AI on HRM

4.1. Algorithmic Bias and Discrimination Risks

Algorithmic bias is the most prominent negative effect of AI in HRM, which may lead to discriminatory violations and damage the fairness of employment. Algorithmic bias in HRM scenarios essentially stems from two distinct and independent sources: data bias and model design bias, which can separately or jointly trigger discriminatory decision-making and form systemic unfairness in organizational HR practices [2].

Data bias derives from inherent flaws in the training data for AI models, the most common source of algorithmic bias in HRM practice. AI systems rely on historical organizational and industry data for iterative learning and decision-making; if the historical data contains implicit or explicit biases related to gender, age, race, region, or the dataset is unrepresentative, unbalanced, or incomplete for specific groups, the algorithm will amplify and solidify these pre-existing unfairness in the training process. For example, in the recruitment of technical positions, if the historical training data is dominated by male employees due to the traditional gender structure of the industry, the AI model may incorrectly associate gender with job competence, resulting in discriminatory screening against female candidates [13]. This type of bias is a passive reflection of the unfairness in the data source, with the AI model only reproducing and strengthening the original bias without independent judgment.

Model design bias originates from artificial technical defects in the development and design

stage of AI models, caused by inappropriate human intervention in the model construction process. Flawed algorithmic logic, unreasonable selection and weight setting of evaluation features, structural defects in model architecture, or the lack of effective debiasing algorithms in the design stage can all lead to model design bias. For instance, in the design of performance evaluation AI models, if developers set excessive weight on quantitative work output indicators and ignore qualitative indicators such as team collaboration and innovation contribution, the model will form a systemic bias against employees engaged in creative and collaborative work, leading to unfair and one-sided evaluation results.

Scholars have verified the existence of algorithmic discrimination in practical cases caused by the two types of biases above. Some AI recruitment systems have been found to downgrade candidates with female-typical names or experience in women's organizations (a typical result of data bias); some performance evaluation algorithms have shown racial bias by penalizing employees whose language and behavior characteristics are associated with specific ethnic groups (often a combination of data imbalance and unreasonable feature selection in model design) [2]. This algorithmic discrimination is concealed and procedural, making it difficult for individuals to defend their rights, and it is easier to form systemic unfairness in the organization.

Algorithmic bias brings serious legal and reputational risks to enterprises. In recent years, lawsuits and regulatory penalties caused by AI discriminatory decisions have increased in Europe and the United States; regulatory authorities have strengthened the review of algorithmic fairness in employment scenarios. For enterprises, algorithmic discrimination not only faces economic losses such as fines and compensation, but also damages the employer brand and social image, leading to the loss of talents and customers.

4.2. Employee Privacy Concerns and Surveillance Anxiety

The wide application of AI in HRM relies on the collection and analysis of a large amount of employee personal data, which triggers prominent privacy and security risks. AI monitoring systems can collect employees' work behavior data in an all-round way, including email content, chat records, attendance status, keystroke frequency, office track and even physiological characteristics [9]. Although these data are used for performance analysis and management decision-making, excessive collection and unauthorized use seriously infringe on employees' personal privacy.

Transparent lack of data usage exacerbates employees' supervision anxiety. Many enterprises do not fully inform employees of the scope, purpose and usage rules of data collection when

applying AI monitoring systems, forming a "transparent employee" management model. Employees feel that their work behaviors are monitored and analyzed all the time, resulting in psychological pressure, anxiety and resistance [5]. Some employees even take perfunctory work, performative behaviors and other ways to "cope" with algorithmic supervision, which reduces work efficiency and creativity.

Data security risks further aggravate privacy threats. The centralized storage of massive sensitive employee data increases the risk of data leakage, theft and abuse. Once a data breach occurs, it will cause serious harm to employees' personal information and bring regulatory penalties and trust crises to enterprises [2]. Privacy and supervision issues have become important obstacles affecting employees' acceptance of AI and the sustainable operation of AI-HR systems.

4.3. Dehumanization of Employment Relationships

AI application leads to the weakening of human interaction and the dehumanization of employment relations, damaging the emotional connection between individuals and organizations. Traditional HRM relies on face-to-face communication between HR professionals, managers and employees, which conveys organizational care and humanistic warmth. When recruitment consultation, performance feedback, training guidance and other links are replaced by AI systems, employees mainly interact with machines and algorithms, and feel treated as data points rather than independent individuals with emotions and needs [9].

The dehumanization effect is more obvious in key employment decisions. When employees encounter dismissal, demotion, rejection of promotion and other results generated by algorithms, they often feel that the decision lacks human judgment, empathy and respect, even if the result is reasonable [2]. This sense of dehumanization reduces employees' trust in the organization, weakens organizational commitment and work engagement, and may even lead to negative behaviors such as absenteeism and resignation.

In addition, the increasing automation of HR processes significantly reduces the opportunities for meaningful interpersonal communication within organizations. As HR professionals become increasingly occupied with managing and operating AI systems, they often find themselves lacking direct communication with employees, which can hinder relationship-building. Moreover, managers tend to rely heavily on algorithmic reports generated by these systems and, as a result, may neglect crucial face-to-face interactions with their subordinates. This weakening of interpersonal interaction can have detrimental effects on the organizational culture and team cohesion, ultimately leading to the alienation of

employment relations. Such alienation not only impacts employee morale but also poses risks to the long-term sustainability and growth of the organization itself.

4.4. Multi-Level Organizational Resistance to AI Adoption

The application of AI in HRM encounters multi-level resistance from organizational stakeholders, affecting the implementation effect and sustainable operation of the system. First, HR professionals have resistance. They worry that AI will replace their core work tasks, leading to job redundancy and loss of professional autonomy, so they take negative attitudes such as perfunctory implementation and passive rejection [5]. Many HR professionals lack the digital skills to operate AI systems, and the fear of technical unemployment further intensifies resistance.

Second, line managers have resistance. AI systems participate in performance appraisal, talent evaluation and other decisions that originally belong to managerial authority, which makes managers feel that their decision-making power is weakened and their status is threatened [10]. Some managers believe that algorithms cannot grasp the complex context of work, and blindly relying on AI will lead to decision-making deviation, so they choose to ignore or bypass AI suggestions, resulting in low utilization of AI systems.

Third, ordinary employees have resistance. Employees resist AI systems that are perceived as unfair, intrusive and dehumanized; some employees take hidden resistance measures such as reducing work input, avoiding data submission and using alternative communication channels [2]. The multi-level organizational resistance increases the implementation cost of AI, reduces the operational efficiency of the system, and even leads to the complete failure of AI-HR projects.

5. Mechanisms Shaping AI Outcomes

5.1. System Design Factors

The design characteristics of AI systems directly determine whether the double-edged sword effect tends to be positive or negative, and are the core technical factors affecting AI outcomes. Transparency and explainability are the primary design principles. "Black box" algorithms with opaque decision-making logic make users unable to understand the basis of AI recommendations, prone to doubt and resistance; AI systems with explainable functions can clearly output decision-making factors, calculation processes and result basis, helping users understand and trust the system [2]. Enterprises should integrate explainable AI technology in

the design stage to improve the transparency of the system.

Fairness-embedded design is the key to avoiding algorithmic bias. AI systems should use diversified and representative training data to avoid data imbalance leading to bias; pre-deployment bias testing and post-operation real-time monitoring should be carried out to correct discriminatory tendencies in a timely manner [10]. The fairness design should cover all protected groups such as gender, race, age and disability to ensure that the algorithm does not produce differential infringement.

Human-in-the-loop design is an important guarantee for positive outcomes. The system should position AI as an auxiliary decision-making tool rather than an independent decision-maker, retaining human review and final decision-making power [5]. This design allows managers to consider complex contextual factors that cannot be captured by algorithms, ensuring that decisions are both efficient and humanistic, and balancing technical rationality and human judgment.

5.2. Implementation Processes

The implementation process of AI systems is equally as important as their technical design; therefore, a standardized and human-centered approach to implementation can significantly mitigate potential negative effects. Effective communication with all stakeholders is a vital prerequisite for ensuring a smooth implementation process. Prior to launching AI systems, enterprises should clearly articulate the purpose, functionalities, usage guidelines, and rights protection measures associated with these systems to HR professionals, managers, and employees. Addressing common concerns related to job security, privacy protection, and fairness is crucial in order to alleviate fears and uncertainties, thereby reducing resistance to change and fostering a more positive reception of the technology among all parties involved [2].

Phased and pilot implementation is a practical strategy. Instead of large-scale full deployment at one time, enterprises should carry out small-scale pilot tests in a single department or a single module, collect user feedback, optimize system design and operation processes, and then gradually promote them to the whole organization [5]. The phased implementation helps to accumulate experience, resolve risks and improve user acceptance.

Comprehensive user training and support are essential. Enterprises should provide targeted training for different groups: for HR professionals, focus on AI system operation, data analysis and ethical governance skills; for managers, focus on how to combine AI suggestions with

practical management; for ordinary employees, focus on system use methods and rights protection channels [10]. Adequate training can improve users' technical literacy and operational confidence, promoting the effective use of AI systems.

5.3. Organizational Context

Organizational contextual factors regulate the relationship between AI application and HRM effects, and determine the adaptation degree of AI in the organization. Organizational culture is a key contextual factor. Enterprises with an innovative, open and trustworthy culture are more tolerant of new technologies, and employees are more willing to accept and try AI systems; enterprises with a conservative, rigid and distrustful culture are more prone to resistance and conflict, amplifying the negative effects of AI [5].

Leadership commitment and support determine the implementation intensity. Senior leaders' positive attitude towards AI, investment of resources and demonstration of use can form a guiding effect within the organization and promote the smooth promotion of AI systems [10]. If leaders lack attention, insufficient investment or inconsistent signals, the implementation of AI will lack institutional guarantee and resource support, leading to poor results.

Resource matching is a basic guarantee. The successful application of AI requires matching technical infrastructure, data management capabilities, professional teams and organizational systems [8]. Enterprises with sufficient digital resources can optimize system operation and user experience; enterprises with insufficient resource investment will face problems such as system lag, data errors and insufficient support, leading to negative outcomes.

5.4. Individual Differences

Individual differences of employees lead to heterogeneous perceptions and responses to AI systems, affecting the final effect of AI application. Technical self-efficacy is a key individual factor. Employees with high technical self-efficacy have strong confidence in using new technologies, can quickly adapt to AI systems, and perceive more positive effects; employees with low technical self-efficacy are afraid of operating difficulties and technical substitution, and are prone to negative perceptions and resistance [14].

Trust in organization and management regulates individual responses. Employees who trust the enterprise and managers believe that the organization will use AI fairly and protect their rights and interests, and have higher acceptance of AI systems; employees with low trust are skeptical of AI application, worried about being treated unfairly and invaded privacy, and amplify negative perceptions [2].

Demographic and occupational characteristics also have an impact. Studies show that young employees, highly educated employees and employees in technical positions have higher acceptance of AI; older employees, less educated employees and employees in traditional positions are more cautious about AI [5]. Enterprises should formulate differentiated communication and training strategies based on individual differences to improve the overall acceptance of AI systems.

6. Countermeasures for Enterprises

6.1. Ethical AI System Design: Foundation of Risk Prevention

Enterprises should take ethical design as the core of AI-HR system construction to prevent risks from the source. First, establish a clear AI ethical governance framework, formulate ethical principles covering fairness, transparency, accountability, privacy protection and human-centeredness, and run through the whole life cycle of AI system development, deployment and operation [2]. The ethical principles should be combined with labor laws and regulations to ensure compliance.

Second, implement full-cycle algorithmic bias governance. Conduct bias audit on training data before system deployment to eliminate discriminatory information; test the algorithm for disparate impact on different groups to ensure fair decision-making; establish a real-time monitoring mechanism after deployment, track the decision results of the system, and correct bias in a timely manner [10]. Record the whole process of bias testing and governance to form a compliant evidence chain.

Third, strengthen the explainability and human oversight of the system. Adopt explainable AI technology to enable users to clearly understand the decision basis of the system; set up a multi-level human review mechanism, and AI can only provide recommendations for key decisions such as recruitment, promotion and dismissal, and the final decision shall be made by managers to ensure that humanistic care and contextual judgment are not missing [5].

6.2. Transparent Implementation Practices: Path to Trust Building

Transparent and standardized implementation practices can enhance stakeholder trust and reduce organizational resistance. First, carry out full-process transparent communication. Disclose the objectives, functions, data usage scope, supervision mechanisms and appeal channels of AI systems to all employees through meetings, manuals, online platforms and other forms; regularly publish the operation status and effect of AI systems to ensure employees' right

to know [2].

Second, promote employee participation in governance. Invite employee representatives to participate in AI system pilot testing, demand design and effect evaluation; collect employees' opinions and suggestions on AI application, and optimize the system and management methods based on feedback [5]. Participatory governance makes employees feel respected and enhances their sense of ownership.

Third, formulate complete AI management policies. Clarify the norms of data collection, storage, use and destruction, the scope of application of algorithmic decisions, the rights and obligations of employees, and the appeal process for unfair decisions; publicize the policies to all employees and strictly implement them to provide institutional guarantee for standardized AI application [10].

6.3. Employee Involvement Mechanisms: Guarantee of Rights Protection

Improving employee participation and rights protection mechanisms can alleviate employees' concerns and enhance their positive perception of AI. First, establish a smooth feedback and appeal channel. Set up a special mailbox, hotline or online platform for employees to report problems such as algorithmic bias, privacy infringement and unfair decisions; respond to and handle feedback in a timely manner to ensure that employees' voices are heard [5].

Second, improve the human review appeal mechanism. Employees who disagree with AI-based decisions can apply for human review; the organization should form an independent review team to re-evaluate the decision, fully consider the employee's situation and contextual factors, and give a reasonable explanation and result [10]. The appeal mechanism ensures that employees have a way to protect their rights and enhances the fairness of management.

Third, carry out employee AI literacy training. Popularize AI basic knowledge, ethical norms and rights protection knowledge to employees, improve employees' ability to understand and use AI systems, and enhance their awareness of self-protection and participation in governance [2]. Literacy training helps eliminate employees' fear of AI and promote positive interaction between people and AI.

6.4. Hybrid Human-AI Work Models: Path to Synergistic Empowerment

Constructing a hybrid human-AI work model is the core strategy to give full play to the advantages of both sides and achieve synergistic empowerment. Figure 1 visually presents the hierarchical division of labor, interactive mechanism and application scenarios of the human-AI hybrid collaboration model in the full cycle of enterprise HRM, which intuitively reflects

the complementary operation logic of AI and humans in different HR core modules.

First, clarify the division of labor between humans and AI. AI is responsible for repetitive, standardized, data-intensive tasks such as resume screening, data statistics and real-time monitoring; humans are responsible for creative, emotional, complex judgment tasks such as cultural fit evaluation, interpersonal communication, conflict resolution and key decision-making [5].

Second, clarify the division of labor between humans and AI. AI is responsible for repetitive, standardized, data-intensive tasks such as resume screening, data statistics and real-time monitoring; humans are responsible for creative, emotional, complex judgment tasks such as cultural fit evaluation, interpersonal communication, conflict resolution and key decision-making [5].

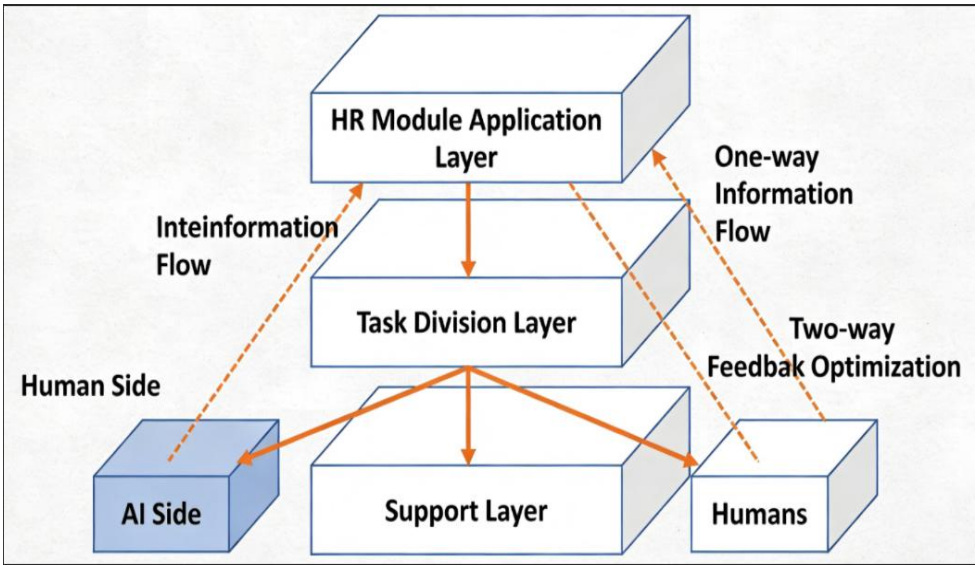

Figure 1.

Third, optimize the organizational process and team structure matching the hybrid model. Adjust the workflow of HRM to adapt to the collaborative operation of humans and AI; train cross-disciplinary teams with HR expertise and AI technology to promote the deep integration of technology and management [10]. The hybrid model realizes the complementary advantages of efficiency and humanization, and maximizes the positive effects of AI while controlling risks.

7. Conclusion and Future Directions

7.1. Summary of Theoretical Contributions

This study systematically explores the double-edged sword effect of artificial intelligence on enterprise human resource management based on the Technology Acceptance Model,

Sociotechnical Systems Theory and Organizational Justice Theory, and forms a complete theoretical and practical framework. Theoretically, this study first integrates multi-dimensional theories to construct a comprehensive analytical framework for AI-HRM effects, clarifying the micro-psychological, organizational-systemic and fairness-perceptual mechanisms of AI's dual effects, and enriching the theoretical system of digital human resource management. Second, this study systematically deconstructs the positive effects of AI on HRM into four dimensions: efficiency, objectivity, personalization and strategicization, and the negative effects into four risks: algorithmic bias, privacy anxiety, dehumanization and organizational resistance, clarifying the dual logic and specific manifestations of AI empowerment and risk. Third, this study reveals the differential impact mechanisms from system design, implementation process, organizational context and individual differences, and proposes a four-in-one governance strategy of ethical design, transparent implementation, employee participation and hybrid collaboration, providing a theoretical basis for enterprises to regulate AI application.

7.2. Implications for Practice

The conclusions of this study have important practical guiding significance for enterprise managers, HR practitioners and technology providers. For enterprise executives, AI application in HRM cannot blindly pursue technological efficiency, but should balance technological advantages and ethical risks, attach importance to system design, organizational matching and employee acceptance, and take risk prevention and control as a necessary part of digital transformation. For HR leaders, they should take the initiative to master AI ethical governance and hybrid management capabilities, transform from traditional administrative executors to digital strategic managers, promote the coordinated development of AI technology and humanistic management, and enhance the strategic value of human resources. For AI technology providers, they should focus on the fairness, transparency and explainability of products in the development of HR-oriented AI systems, embed human oversight and ethical governance functions, and develop human-centered AI tools to meet the actual needs of enterprises.

7.3. Limitations and Future Research Directions

This study has certain limitations. As a theoretical research, this paper constructs a framework based on existing literature and practical cases, but lacks empirical testing such as questionnaire survey and empirical analysis to verify the causal relationship and impact effect of various variables. In addition, this study does not distinguish the differential effects of AI in

different industries, enterprise scales and institutional environments, and the universality of the conclusions needs to be further verified.

Future research can be carried out from four aspects. First, conduct empirical research to collect data through questionnaires, interviews and case studies, test the theoretical framework of this paper, verify the impact of various factors on the double-edged sword effect of AI, and clarify the boundary conditions of AI positive and negative effects. Second, explore the long-term dynamic impact of AI on employment relations, track the changes in employees' trust, engagement and organizational behavior after long-term use of AI systems, and reveal the evolution law of AI effects over time. Third, compare the effectiveness of different governance strategies, analyze the governance effects of ethical design, transparent implementation and other paths, and find the optimal combination strategy suitable for Chinese enterprises. Fourth, carry out cross-cultural and cross-institutional comparative research, explore the differences in AI-HRM effects and governance paths under different regulatory systems and cultural backgrounds, and provide a reference for global digital HRM practices.

References

[1] Bostrom, R. P., & Heinen, J. S. (1977). MIS problems and failures: A socio-technical perspective. *MIS Quarterly*, 1(3), 17-32.

[2] Budhwar, P., Chowdhury, S., Wood, G., Aguinis, H., Bamber, G. J., Beltran, J. R., & Varma, A. (2023). Human resource management in the age of generative artificial intelligence: Perspectives and research directions on ChatGPT. *Human Resource Management Journal*, 33(3), 606-659.

[3] Colquitt, J. A. (2001). On the dimensionality of organizational justice: A construct validation of a measure. *Journal of Applied Psychology*, 86(3), 386-400.

[4] Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3), 319-340.

[5] Fernández-Vidal, J., Peral-Peral, B., & Gascó, J. L. (2025). Technology-driven change in human resource management: Reshaping talent management and organizational design. *Administrative Sciences*, 15(11), 452.

[6] Malyshev, V., Lipskyi, Y., Kovalenko, V., Gab, A., Shakhnin, D., & Orel, O. (2024). Assessment of the global artificial intelligence market in healthcare. *Technology audit and production reserves*, 6(4/80), 62-70.

[7] Greenberg, J. (1987). A taxonomy of organizational justice theories. *Academy of Management Review*, 12(1), 9-22.

[8] Ioniță, A., & Ștefan, S. C. (2025). Strategic human resource management in the digital era: Technology, transformation, and sustainable advantage. *Merits*, 5(4), 23.

[9] Kellogg, K. C., Valentine, M. A., & Christin, A. (2020). Algorithms at work: The new contested terrain of control. *Academy of Management Annals*, 14(1), 366-410.

[10] Kim, S., Khoreva, V., & Vaiman, V. (2025). Strategic human resource management in the era of algorithmic technologies: Key insights and future research agenda. *Human Resource Management*, 64(2), 447-464.

[11] Marliyas, A., Ummah, M. A. C. S., & Gunapalan, S. (2026). The transformation of talent

acquisition through artificial intelligence in the context of Industry 4.0: A systematic literature review. *Journal of Business Research and Innovation*, 11(2).

[12] Trist, E. L., & Bamforth, K. W. (1951). Some social and psychological consequences of the longwall method of coal-getting. *Human Relations*, 4(1), 3-38.

[13] Vaishnavi, D., Hada, V., Sharma, R., Singh, G., Singh, R. K., & Bhingardive, S. (2026). Integrating AI into human resource management: Implications for recruitment and retention. *Journal of Asia Entrepreneurship and Sustainability*, 22(1).

[14] Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. *MIS Quarterly*, 27(3), 425-478.

Online ISSN: 3104-9338
Print ISSN: 3104-932X

A Binary Classification Detection Method for Smart Contract Honeypots Based on LSTM-Attention-CNN

Chenran Xi[1*], Handong She[2]

[1] Taiyuan Normal University

[2] Taiyuan University of Science and Technology

Received: April 13, 2026

Revised: April 14, 2026

Accepted: April 15, 2026

Published online: April 25, 2026

To appear in: *International Journal of Advanced AI Applications,* Vol. 2, No. 5 (May 2026)

* Corresponding Author: Chenran Xi (1215819301@qq.com)

Abstract. To address the technical bottlenecks of existing Smart Contract Honeypot detection methods, such as reliance on expert rules, low detection efficiency, and difficulty in identifying new honeypots, this paper proposes a binary classification detection method for Smart Contract Honeypots based on the fusion of Long Short-Term Memory (LSTM) network, Attention mechanism, and Convolutional Neural Network (CNN). Taking smart contract source code as the research object, this method converts code into trainable sequence data through data preprocessing, and designs a hybrid LSTM-Attention-CNN model to jointly capture temporal dependency features and local key patterns of code, so as to accurately distinguish Smart Contract Honeypots from normal contracts. The cross-entropy loss function and Adam optimization algorithm are introduced, combined with the early stopping strategy to avoid model overfitting and improve detection stability. Experiments are carried out based on the Ethereum smart contract dataset. The results show that the accuracy, recall rate and F1-score of the proposed model on the test set reach 98.97%, 98.82% and 98.89% respectively. Compared with single LSTM, CNN and BLSTM-ATT models, the detection performance is significantly improved, and the detection efficiency meets the requirements of large-scale batch contract detection, providing an efficient and reliable technical means for smart contract security audit.

Keywords: *Smart Contract; Smart Contract Honeypot; LSTM; Attention Mechanism; TextCNN*

1. Introduction

Since the release of the Bitcoin whitepaper in 2008 [1], blockchain technology has gradually

penetrated from the digital currency field to multiple fields such as Decentralized Finance (DeFi), supply chain management, and Non-Fungible Tokens (NFT), becoming an important support for the development of the digital economy. As the core component of blockchain technology, smart contracts have greatly expanded the application boundaries of blockchain with the characteristics of automatic execution, immutability, transparency and traceability [2]. According to DeFiPulse statistics, by the end of 2024, the total value of locked assets in various DeFi protocols around the world has exceeded 50 billion US dollars, highlighting the core position of smart contracts in the digital economy.

However, once deployed, smart contract code cannot be modified and directly manages a large number of digital assets, so its security vulnerabilities can easily lead to serious economic losses. The DAO attack in 2016 caused the theft of about 60 million US dollars worth of Ether, directly leading to a hard fork of the Ethereum blockchain; the Ronin Network cross-chain bridge attack in 2022 resulted in losses of more than 600 million US dollars [3]. Such security incidents not only damage users' property security, but also seriously undermine the credibility of the blockchain ecosystem. Among many smart contract security threats, Smart Contract Honeypots, as a new type of fraud, have become an important hidden danger endangering the blockchain ecosystem due to their strong concealment and high deception [4].

Smart Contract Honeypots lure attackers to invest funds to exploit "vulnerabilities" by deliberately embedding fake vulnerabilities, and finally steal attackers' funds through hidden logic [5]. Different from traditional vulnerability attacks, the deception mechanism of Smart Contract Honeypots relies on the characteristics of Solidity language and the underlying mechanism of Ethereum Virtual Machine (EVM), and constantly evolves new variants, bringing great challenges to detection work. Existing Smart Contract Honeypot detection methods are mainly divided into three categories: first, rule-based methods relying on experts, which require manual definition of detection patterns, lag behind new honeypots in rule update, and have a high missed detection rate [6]; second, symbolic execution-based methods, such as the HONEYBADGER tool [5], which have high detection accuracy but suffer from path explosion, low detection efficiency, and cannot adapt to large-scale contract detection scenarios; third, methods based on a single deep learning model, such as BLSTM-ATT vulnerability detection [12], the former loses code semantic information, and the latter is not optimized for Smart Contract Honeypots, resulting in limited detection performance.

Deep learning technology has unique advantages in sequence data processing and feature extraction. Among them, LSTM is good at capturing temporal dependencies of sequence data

and suitable for processing context associations of smart contract source code; Attention mechanism can adaptively focus on key code segments and improve the pertinence of feature extraction; TextCNN can effectively capture key patterns of local features of code [7]. Based on this, this paper fuses the three to design a hybrid LSTM-Attention-CNN model specially used for binary classification detection of Smart Contract Honeypots and normal contracts, solving the problems of low efficiency and insufficient accuracy of existing methods. Experimental verification shows that this method outperforms existing mainstream methods in both detection performance and efficiency, providing a new technical path for smart contract security detection.

2. Related Technologies

2.1. Core Characteristics of Smart Contract Honeypots

A Smart Contract Honeypot is a type of malicious smart contract disguised as having exploitable vulnerabilities for fraud purposes, and its core characteristics are reflected in two aspects: "deceptiveness" and "concealment". According to the classification by Torres et al. [5], Smart Contract Honeypots are mainly divided into 8 types, among which hidden state update type (accounting for 46.6%), balance disorder type, and inheritance disorder type are the most common. The common feature of such contracts is that there are obvious "exploitable vulnerabilities" on the surface (such as abnormal balance check logic and function call ambiguity), but in fact, hidden code (such as inline assembly and concealed transfer logic) is used to tamper with contract state or steal funds. Their source code contains specific keywords and syntax patterns, providing a feature basis for deep learning-based detection.

2.2. Principle of LSTM Network

Long Short-Term Memory (LSTM) network is an improved model to solve the problems of gradient disappearance and gradient explosion of traditional Recurrent Neural Network (RNN). It realizes effective memory and screening of sequence historical information by introducing three control gates: input gate, forget gate and output gate [8]. The mathematical expressions of its core structure are as follows:

$$f_t=\sigma\left(W_f\cdot[h_{t-1},x_t]+b_f\right)$$

$$i_t=\sigma(W_i\cdot[\mathrm{h_{t-1}},\mathrm{x_t}]+b_i)$$

$$\tilde{C}_t=tanh(W_C\cdot[\mathrm{h_{t-1}},\mathrm{x_t}]+b_C)$$

$$C_t=f_t\odot C_{t-1}+i_t\odot\tilde{C}_t$$

$$o_t=\sigma(W_o\cdot[h_{t-1},x_t]+b_o)$$

$$h_t=o_t\odot tanh(C_t)$$

Where f_t, i_t and o_t are the outputs of forget gate, input gate and output gate respectively; C_t and $\tilde{C}_t$ are the current cell state and candidate cell state respectively; h_t and h_{t-1} are the current and previous hidden states respectively; x_t is the input at the current moment; W_f, W_i, W_C and W_o are weight matrices; b_f, b_i, b_C and b_o are bias terms; σ is the sigmoid activation function; tanh is the hyperbolic tangent activation function; $\odot$ is element-wise multiplication.

2.3. Attention Mechanism

Attention mechanism calculates the weight of each element in the input sequence to focus on features more important to the task and suppress interference from irrelevant information [9]. This paper adopts the additive attention mechanism. Its core is to map hidden states to the same dimension through linear transformation, and then calculate attention weights through the softmax function. The mathematical expressions are as follows:

$$e_{it}=v_a^T tanh(W_a h_i+U_a h_t)$$

$$\alpha_{it}=\frac{exp(e_{it})}{\sum_{k=1}^{T} exp(e_{ik})}$$

$$h_{att}=\sum_{i=1}^{T}\alpha_{it}h_i$$

Where e_{it} is the correlation score between the i-th hidden state and the t-th moment; v_a, W_a and U_a are trainable parameters; α_{it} is the attention weight; h_{att} is the fused feature output by the attention mechanism; T is the sequence length.

2.4. TextCNN Model

TextCNN captures local features of sequence data through convolution operations, and is suitable for extracting local key patterns in smart contract source code (such as hidden transfer and assembly instructions) [10]. Its core process is: input the embedded sequence data into convolution kernels of different sizes, extract local features through convolution and pooling operations, and finally splice them into a global local feature vector. The mathematical expression of convolution operation is as follows:

$$c_j=tanh(W_k\cdot x_{i:i+k-1}+b_k)$$

Where c_j is the output of the j-th convolution kernel; W_k is the weight of the convolution

kernel with size k; $x_{i:i+k-1}$ is a local segment of length k in the input sequence; b_k is the bias term of the convolution kernel.

3. Binary Classification Detection Method for Smart Contract Honeypots Based on LSTM-Attention-CNN

3.1. Overall Framework of the Method

The overall Smart Contract Honeypot binary classification detection method proposed in this paper is divided into four stages: data preprocessing, feature extraction, model training and classification detection. The framework flow chart is shown in Figure 1.

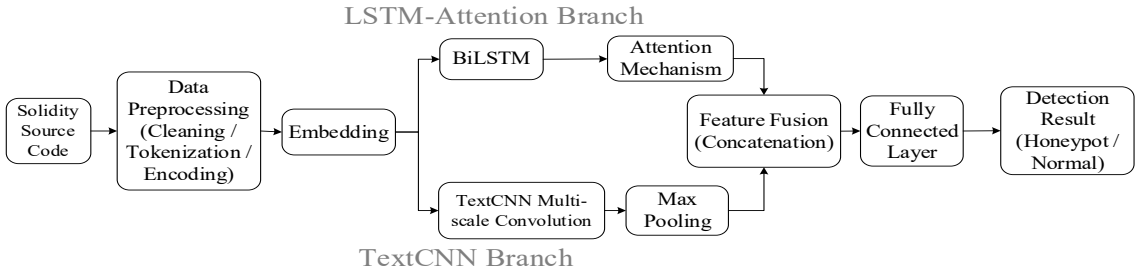

Figure 1. Overall framework flow chart of the method.

The data preprocessing stage converts the original smart contract source code into sequence data trainable by the model; the feature extraction stage captures the temporal dependency features of the code through the LSTM-Attention branch, captures the local key features through the TextCNN branch, and then performs feature fusion; the model training stage improves the model performance through optimization algorithms and regularization strategies; the classification and detection stage performs binary classification judgment on the input contract and outputs the detection results.

3.2. Data Preprocessing

Smart contract source code contains irrelevant information such as comments, blank lines and redundant characters, which needs to be converted into standardized sequence data through preprocessing. The specific steps are as follows:

(1) Source code cleaning: Remove comments, blank lines and redundant spaces, retain core code logic; unify the format of variable names and function names (such as converting camel case naming to underscore naming) to eliminate the impact of format differences.

(2) Lexical analysis and tokenization: Use Solidity lexical analysis tools to decompose the cleaned source code into tokens, including keywords (such as contract, function), identifiers, operators, constants, etc.; filter meaningless tokens (such as semicolons, brackets) to obtain

pure token sequences.

(3) Vocabulary construction and sequence encoding: Count the occurrence frequency of all tokens, retain tokens with occurrence times ≥ 3 to construct a special vocabulary; convert token sequences into numerical sequences using One-Hot encoding, and use a unified "unknown token" encoding for tokens not in the vocabulary; unify all sequences to a fixed length L (L=200 in this paper) through padding or truncation to obtain standardized input sequence X=$[x_1, x_2, \ldots, x_L]$, where x_i is the encoding value of the i-th token.

3.3. Hybrid Model Architecture Design

The hybrid LSTM-Attention-CNN model proposed in this paper is composed of several essential components, including an embedding layer, an LSTM-Attention branch, a TextCNN branch, a feature fusion layer, and a classification layer. This meticulously designed architecture facilitates the effective integration of sequential data processing and attention mechanisms with convolutional neural networks, allowing for improved contextual understanding and feature extraction. The overall structure of the model is clearly illustrated in Figure 2, which provides a comprehensive visual representation of the various layers and their interactions, effectively demonstrating how these components work together to enhance performance on the specified tasks. This layered approach aims to leverage the strengths of each individual component, ultimately leading to superior results in the targeted applications.

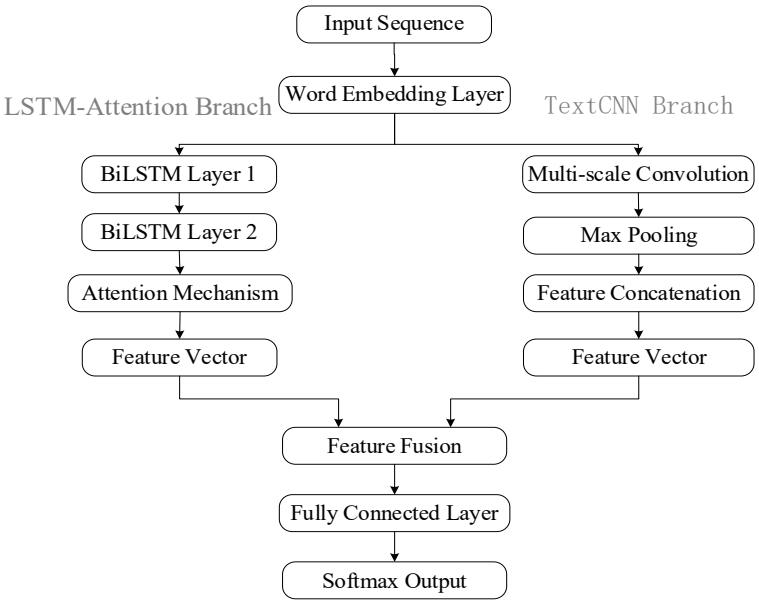

Figure 2. Structure diagram of hybrid LSTM-Attention-CNN model.

3.3.1. Embedding Layer

The embedding layer maps standardized numerical sequences into low-dimensional dense

vectors to reduce dimensionality disaster and retain semantic associations between tokens. Let the vocabulary size be V and the embedding dimension be d. The embedding layer converts the input sequence X into an embedded vector sequence $X_{emb} = [x_{emb1}, x_{emb2}, \dots, x_{embL}]$ through the trainable matrix $W_{emb} \in \mathbb{R}^{V \times d}$, where $x_{embi} \in \mathbb{R}^d$ is the embedding vector of the i-th token.

3.3.2. LSTM-Attention Branch

This branch is used to extract temporal dependency features of smart contract source code and capture associations between code contexts (such as function call order and state variable modification logic). Input the embedded vector sequence X_{emb} into the bidirectional LSTM layer to obtain the bidirectional hidden state sequence $H_{LSTM} = [h_1, h_2, \dots, h_L]$, where $h_i = h_i^{forward}, h_i^{backward}$, which are the hidden states of forward and backward LSTM respectively.

Input H_{LSTM} into the attention mechanism layer, calculate attention weights through formulas, and obtain the vector $h_{att} \in \mathbb{R}^{2n}$ fused with key temporal features (n is the dimension of LSTM hidden layer).

3.3.3. TextCNN Branch

This branch is used to extract local key features in the source code (such as assembly, sstore and other honeypot feature token combinations). Three convolution kernels of different sizes (3, 4, 5) are adopted, with 32 convolution kernels for each size, to perform convolution operations on the embedded vector sequence X_{emb} to obtain local features of different dimensions; perform global max pooling on each convolution output to retain the most representative local features; splice all pooled features to obtain the local feature vector $h_{cnn} \in \mathbb{R}^{3 \times 32}$.

3.3.4. Feature Fusion and Classification

Splice the temporal feature vector h_{att} and the local feature vector h_{cnn} to obtain the fused feature vector $h_{fusion} = [h_{att}, h_{cnn}]$; input the fused feature into the fully connected layer, perform feature mapping through the ReLU activation function, and introduce a Dropout layer (dropout rate=0.5) to prevent overfitting; finally output the binary classification probability through the sigmoid activation function. When the probability ≥ 0.5, it is judged as a Smart Contract Honeypot, otherwise it is a normal contract.

3.4. Model Training Strategy

3.4.1. Loss Function

The cross-entropy loss function is used to measure the difference between the model predicted value and the real label, which is suitable for binary classification tasks. The mathematical expression is as follows:

$$L = -\frac{1}{N}[y_i log(\hat{y}_i) + (1-y_i) log(1-\hat{y}_i)]$$

Where N is the number of samples; y_i is the real label of the i-th sample (0=Normal Contract, 1=Smart Contract Honeypot); $\hat{y}_i$ is the probability that the model predicts the i-th sample is a Smart Contract Honeypot.

3.4.2. Optimization Algorithm and Early Stopping Strategy

The Adam optimization algorithm is adopted to minimize the loss function. This algorithm combines momentum gradient descent and adaptive learning rate adjustment, with fast convergence speed and good stability [11]. The learning rate is set to 0.001, the attenuation coefficients $\beta_1 = 0.9$, $\beta_2 = 0.999$, and the regularization parameter $\epsilon = 1e-8$.

The early stopping strategy is introduced to avoid model overfitting: divide the dataset into training set and validation set (ratio 8:2), monitor the validation set loss during training, stop training when the validation set loss does not decrease for 5 consecutive epochs, and save the current optimal model parameters.

3.4.3. Hyperparameter Selection

The optimal hyperparameters of the model are determined through grid search. The core hyperparameters are set as follows: embedding dimension d=128, LSTM hidden layer dimension n=64, TextCNN convolution kernel sizes 3/4/5, number of convolution kernels for each size 32, number of neurons in fully connected layer 128, dropout rate=0.5, batch size=32, maximum training epochs=50.

4. Experiments and Result Analysis

4.1. Experimental Environment and Dataset

4.1.1. Experimental Environment

Experimental hardware environment: CPU is Intel Core i7-12700H, memory 32GB, GPU is NVIDIA RTX 3060 (6GB); software environment: operating system is Ubuntu 20.04, programming language is Python 3.8, deep learning framework is PyTorch 1.12.0, lexical

analysis tool is SolidityParser, data processing tools are Pandas and Numpy.

4.1.2. Dataset Construction

The experimental dataset is derived from public contracts on the Ethereum mainnet and existing research datasets [5,12]. A total of 10,000 smart contract source codes are collected, including 3,000 Smart Contract Honeypots (covering 8 main types) and 7,000 normal contracts (including DeFi protocols, NFT contracts, ordinary transfer contracts, etc.). All contracts are written in Solidity language, with versions covering 0.4.x-0.8.x, ensuring the diversity and representativeness of the dataset.

The dataset is divided into training set (8,000), validation set (1,000) and test set (1,000) according to the ratio of 8:1:1. Random shuffling and deduplication strategies are adopted to avoid model deviation caused by uneven data distribution.

4.1.3. Evaluation Metrics

Accuracy, Recall, Precision and F1-score are used as model evaluation metrics. The calculation formulas are as follows:

$$Accuracy=\frac{TP+TN}{TP+TN+FP+FN}$$

$$Precision=\frac{TP}{TP+FP}$$

$$Recall=\frac{TP}{TP+FN}$$

$$F1=2\times\frac{Precision\times Recall}{Precision+Recall}$$

Where TP (True Positive) is the number of contracts correctly judged as honeypots; TN (True Negative) is the number of contracts correctly judged as normal; FP (False Positive) is the number of normal contracts misjudged as honeypots; FN (False Negative) is the number of Smart Contract Honeypots misjudged as normal.

4.2. Experimental Results and Analysis

4.2.1. Model Performance Comparison

To verify the superiority of the proposed hybrid LSTM-Attention-CNN model, it is compared with the single LSTM model, single TextCNN model and BLSTM-ATT model (for reentrancy vulnerability detection [12]) in comparative experiments. The experimental results are shown in Table 1.

It can be seen from Table 1 that the LSTM-Attention-CNN model proposed in this paper

outperforms the comparison models in all evaluation metrics: the accuracy is increased by 6.62 percentage points compared with the single LSTM and 3.09 percentage points compared with BLSTM-ATT; the F1-score reaches 98.89%, indicating that the model has excellent binary classification performance. Although the single contract detection time is slightly longer than that of single models, the detection speed of 23.1 ms can still meet the demand of large-scale batch contract detection (about 43 contracts can be detected per second).

Table 1. Performance comparison of different models on the test set.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Single contract detection time (ms)
Single LSTM	92.35	91.82	92.17	92.00	18.6
Single TextCNN	93.72	93.15	93.58	93.36	12.3
BLSTM-ATT	95.88	95.42	95.67	95.54	21.5
Proposed LSTM-Attention-CNN	98.97	98.75	98.82	98.89	23.1

Analysis reasons: the single LSTM model can only capture temporal features and cannot effectively identify local key patterns; the single TextCNN model is good at local feature extraction but lacks the capture of context dependencies; the BLSTM-ATT model introduces the attention mechanism but is not optimized for Smart Contract Honeypot features and does not fuse local features; while the proposed model combines temporal features and local features, and focuses on key honeypot code segments through the attention mechanism, significantly improving detection accuracy.

4.2.2. Ablation Experiment

To verify the effectiveness of each component of the model, ablation experiments are carried out by removing the Attention mechanism and TextCNN branch respectively, and the model performance changes are compared. The results are shown in Table 2.

Table 2. Ablation experiment results.

Model Variant	Accuracy (%)	Recall (%)	F1-score (%)
Complete model (LSTM-Attention-CNN)	98.97	98.82	98.89
Remove Attention mechanism (LSTM-CNN)	96.53	96.38	96.45
Remove TextCNN branch (LSTM-Attention)	95.92	95.77	95.84

It can be seen from Table 2 that after removing the Attention mechanism, the model accuracy decreases by 2.44 percentage points and the F1-score decreases by 2.44 percentage points, indicating that the attention mechanism can effectively focus on key honeypot features and improve model discrimination ability; after removing the TextCNN branch, the model accuracy decreases by 3.05 percentage points and the F1-score decreases by 3.05 percentage points, indicating that the local features extracted by TextCNN play an important role in honeypot

detection. The fusion of the two components can achieve complementary advantages and significantly improve model performance.

4.2.3. Comparison with Existing Methods

The proposed method is compared with existing mainstream honeypot detection methods (HONEYBADGER [5], Camino et al.'s method [13]). The results are shown in Table 3.

It can be seen from Table 3 that the proposed method is significantly superior to existing methods in accuracy and detection efficiency: the accuracy is increased by 13.97 percentage points compared with HONEYBADGER; the detection efficiency reaches 43.3 contracts/second, much higher than HONEYBADGER. It does not rely on transaction history data and is suitable for real-time detection of newly deployed contracts, with stronger practical application value.

Table 3. Performance comparison with existing methods.

Method	Core Technology	Accuracy (%)	Detection Efficiency (contracts/second)	Application Scenario
HONEYBADGER [5]	Symbolic execution + heuristic rules	85.00	0.8	Small-scale accurate detection
Camino et al. [13]	Transaction behavior analysis + machine learning	89.50	12.5	Contracts with transaction history
Proposed method	LSTM-Attention-CNN + source code	98.97	43.3	Large-scale batch detection

5. Conclusion and Prospect

5.1. Conclusion

Aiming at the problems of low efficiency, insufficient accuracy and difficulty in adapting to large-scale scenarios of existing Smart Contract Honeypot detection methods, this paper proposes a binary classification detection method for Smart Contract Honeypots based on LSTM-Attention-CNN. The main work is as follows:

(1) A set of data preprocessing process for smart contract source code is designed. Through cleaning, lexical analysis and sequence encoding, the original code is converted into standardized training data, eliminating the interference of format differences and irrelevant information and laying a foundation for model training.

(2) A hybrid LSTM-Attention-CNN model is constructed, which combines the temporal

feature extraction ability of LSTM, the key feature focusing ability of Attention mechanism and the local feature extraction ability of TextCNN to comprehensively capture Smart Contract Honeypot features and improve binary classification detection accuracy.

(3) Adam optimization algorithm, cross-entropy loss function and early stopping strategy are introduced to optimize the model training process, avoid overfitting, and improve the stability and generalization ability of the model; the optimal hyperparameters are determined through grid search to further optimize model performance.

Experimental results show that the accuracy of the proposed model on the test set reaches 98.97% and the F1-score reaches 98.89%. The detection efficiency meets the requirements of large-scale batch detection. Compared with existing methods, it has significant advantages and can effectively identify various Smart Contract Honeypots, providing technical support for smart contract security audit.

5.2. Research Limitations and Future Prospect

There are still some limitations in this research: first, model training relies on a large amount of labeled data, and the detection performance for new unlabeled Smart Contract Honeypots needs to be improved; second, the model is only designed for Solidity language contracts and has insufficient adaptability to other smart contract languages (such as Vyper); third, the impact of contract obfuscation technology on detection performance is not considered.

Future research directions mainly include:

(1) Introduce semi-supervised learning or transfer learning technology to reduce the dependence on labeled data and improve the detection ability of new Smart Contract Honeypots;

(2) Expand model adaptability, optimize data preprocessing and feature extraction modules to support multi-language smart contract detection;

(3) Research confrontation methods of contract obfuscation technology to improve the anti-interference ability of the model through feature enhancement;

(4) Combine binary classification detection with fine-grained type judgment to further improve the Smart Contract Honeypot detection system.

References

[1] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Working Paper.*
[2] Buterin, V. (2014). Ethereum: A next-generation smart contract and decentralized application platform. *White Paper.*
[3] Ronin Network. (2022). Ronin bridge exploit post-mortem. *Technical Report.*

[4] Torres, C. F., Steichen, M., & State, R. (2019). The art of the scam: Demystifying honeypots in Ethereum smart contracts. *In Proceedings of the 28th USENIX Security Symposium* (pp. 1591–1607). USENIX Association.
[5] Torres, C. F., & Steichen, M. (2019). The art of the scam: Demystifying honeypots in ethereum smart contracts. *In 28th USENIX Security Symposium (USENIX Security 19)* (pp. 1591-1607).
[6] Liu, Z., Jiang, M., Zhang, S., Zhang, J., & Liu, Y. (2023). A smart contract vulnerability detection mechanism based on deep learning and expert rules. *IEEE Access*, 11, 77990-77999.
[7] Kim, Y. (2014). Convolutional neural networks for sentence classification. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (pp. 1746–1751).
[8] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
[9] Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv*:1409.0473.
[10] Zhang, Y., & Wallace, B. (2015). A sensitivity analysis of convolutional neural networks for sentence classification. *arXiv preprint arXiv*:1510.03820.
[11] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv*:1412.6980.
[12] Rahardja, U., & Aini, Q. (2026). Enhancing Blockchain Security Through Smart Contract Vulnerability Classification Using BiLSTM and Attention Mechanism. *Journal of Current Research in Blockchain*, 3(1), 28-45.
[13] Otoum, Y., Asad, A., & Nayak, A. (2025). Blockchain meets adaptive honeypots: A trust-aware approach to next-gen iot security. *IEEE Transactions on Network Science and Engineering.*

Online ISSN: 3104-9338
Print ISSN: 3104-932X

Investigation of Partial Image Classification Methods

Ziwen Dong[1], Ijazul Haq[2], Shan Huang[1], Jin Y. Du[2*]

[1] Guangdong Janus Biotechnology Co., Ltd.

[2] Guangdong CAS Angels Biotechnology Co., Ltd.

Received: April 7, 2026

Revised: April 14, 2026

Accepted: April 15, 2026

Published online: April 23, 2026

To appear in: *International Journal of Advanced AI Applications,* Vol. 2, No. 5 (May 2026)

* Corresponding Author: Jin Y. Du (jinyduphd@gmail.com)

Abstract. Recognizing an object based on only partial information is a common task that humans perform every day. In this study, we explore how accurate several computer algorithms, including traditional methods and LVMs (Large Vision Models), perform at image classification using a novel dataset comprised of 10 different animal classes. The traditional methods we use are Resnet and Transformer, while the LVMs are GPT-4, Claude, Gemini, LLaVa, Qwen, and CLIP (Contrastive Language-Image Pre-training). The dataset consists of 16K manually cropped images, providing a unique challenge in assessing the models' ability to recognize images based on incomplete information. The results indicate significant variations in model performance. Swin Transformer achieves the best accuracy, outperforming even humans. On the other hand, LVMs under zero-shot underperform humans; but benefit from few-shot preparation.

Keywords: *Image Classification; Large Vision Models; Computer Vision; GPT; Generative AI; Machine Learning*

1. Introduction

Large Language Models (LLMs), such as GPT-3 [1], PaLM [2], LlaMA [3] and Vicuna [4], are well known for making advancements in the field of NLP (Natural Language Processing) using extensive pretraining and vast network architectures. Recently, GPT-4 [5] extended these advancements to multimodal data, spurring the rapid development of Large Multimodal Models (LMMs). These multimodal models harness the knowledge from LLMs to effectively align visual features with textual data. LMMs have been used in a variety of tasks, ranging from object detection to complex scene understanding [6]. However, the robustness of these models in dealing with partial or incomplete visual data remains relatively underexplored. This gap is

particularly evident in the realm of image classification, where the integrity of the visual input is typically assumed to be complete.

Partial image classification presents a unique set of challenges and opportunities. It involves the identification and classification of objects from images where only fragments are visible. This scenario is common in real-world applications such as surveillance, where objects of interest are often occluded or only partially visible due to various obstructions. Additionally, partial image classification can enhance the capability of autonomous systems, such as drones, robots, or self-driving cars, which must operate effectively in dynamic environments with incomplete information.

The objective of this study is to evaluate the performance of vision models on a novel dataset specifically designed for this purpose. The dataset comprises 16K instances of partial animal images, carefully crafted to simulate a range of occlusions and partial visibility scenarios. The models investigated in this study include: GPT-4 [5], Claude [7], Gemini [8], LLaVa [9], Qwen [10], CLIP [11], Resnet [12], and Swin-Transformer [13]. Furthermore, we compared the models' performance with that of humans. To test human performance, we developed a specialized crowdsourcing platform and asked volunteer annotators to identify the correct animal category from the partial images presented to them.

Preliminary investigations into this dataset have revealed substantial variations in the performance of different LVMs, indicating that the ability to effectively handle partial images may be an important discriminator of model capability. This paper seeks to understand these variations in depth, exploring how different architectures and training strategies affect the efficacy of LVMs in partial image classification.

Figure 1. Images of partially visible animals.

This study may find applications in the fields of computer vision, robotics, and self-driving cars, among others. For instance, robots are now capable enough to identify and recognize objects or animals in images where a sufficient portion of the image is visible; however, they

may struggle in cases such as shown in Figure 1 (left), where only part of the animal (cat) is visible. Similarly, self-driving cars can easily identify animals when they are clearly visible, but in cases such as Figure 1 (right), where only a portion of the animal's body is visible, the car may overlook the animal.

The remainder of this paper is organized as follows: Section 2 provides background information and reviews existing literature on image classification, especially with LVMs. Section 3 describes our dataset collection and preparation. Section 4 introduces the methods used. Section 5 discusses how we prompt LVMs. Section 6 details the hyperparameter settings and testing details. Section 7 presents the testing results and the conclusions. Finally, discussions for future work are given in Section 8.

2. Background and Literature Review

Computer vision is the branch of AI concerned with making decisions based on input images or videos. This paper is concerned with one of the central problems of computer vision: image classification. To address this problem, researchers have proposed various models and techniques such as deep learning convolutional neural networks (CNNs) [21]. Within the realm of CNNs, some renowned models include VGGNet [29], Resnet [12], GoogLeNet [30] and MobileNets [31]. The emergence of the Vision Transformer [32] and Swin Transformer [13] model architectures has brought about a new transformation in the field of computer vision, largely due to their innovative application of self-attention mechanisms. These mechanisms allow the models to capture complex relationships and dependencies within images, leading to improved performance compared to traditional CNNs [32]. Building upon this foundation, PMANet [38] further explores the potential of attention mechanisms to address the challenges of skin disease classification. More recently, the application of large AI models has become the trend. Models such as GPT-4 [5], which use more than one trillion parameters, have shown the potential to handle a wide and complex range of tasks.

2.1. Large Vision Models for Image Classification

Li et al. [14] conduct a systematic study on the reliability of Large Vision-Language Models in image understanding tasks, highlighting their tendency to produce semantically inconsistent outputs when interpreting visual content. To address this issue, they propose a polling-based query strategy that reformulates model evaluation as a binary decision problem, thereby improving the robustness of LVM predictions. Liu et al. [15] comprehensively evaluate the performance of LVLMs in Optical Character Recognition (OCR). Xu et al. [16] introduce

LVLM-Ehub, a comprehensive benchmarking framework for evaluating LVLMs, providing a dual evaluation approach combining quantitative capability assessments and an online arena platform for more dynamic, user-involved testing. MM-Vet, a benchmark developed by Yu et al. [17], evaluates LMMs on complex tasks that integrate multiple Vision-Language (VL) capabilities including recognition, OCR, knowledge, language generation, spatial awareness, and mathematics. MM-Vet encompasses 16 tasks derived from these capabilities, offering a nuanced framework to assess how well models perform across varied and integrated tasks. Li et al. [18] design a benchmark and toolkit, ELEVATER (Evaluation of Language-augmented Visual Task-level Transfer), to evaluate the performance of language-augmented visual models, addressing the need for standardized testing methods to identify the challenges associated with assessing these models' transferability across diverse datasets and tasks. It features 20 image classification datasets and 35 object detection datasets, each enhanced with external knowledge to test models under zero-shot, few-shot, and full model fine-tuning scenarios. Yin et al. [19] introduce the Language-Assisted Multi-Modal (LAMM) framework, an open-source endeavor for evaluating and enhancing MLLMs (Multi-model Large Language Models), which focuses on the integration of language and visual data, creating an ecosystem that supports the development of AI agents capable of complex, real-world tasks.

2.2. Partial Image Classification

Making classification decisions based on partial images poses a unique and significant challenge in the field of computer vision. Unlike typical image classifications, where the completeness and clarity of images are generally assumed, partial image classification requires advanced models to maintain high accuracy even when critical visual information is missing. This scenario tests the models' capability to leverage contextual clues and extract meaningful insights from incomplete data to make informed predictions. Such capabilities are particularly crucial in applications like surveillance systems and autonomous navigation, where understanding partially visible objects can make a substantial difference. Several studies have investigated the complexities of image classification under conditions of partial visibility or object occlusion. For instance, subspace decomposition-based methods attempt to estimate missing deep features from occluded images to improve overall classification robustness [39]. Additionally, deep feature augmentation strategies further address this pressing issue by enriching feature representations during the training phase, thereby alleviating performance degradation that is often caused by partial observation [40]. Jeong, Lee, and Son [37] conduct an insightful study specifically focused on classification based on partial images, but they

concentrate on vehicles as the primary subject matter. Furthermore, recent benchmark studies systematically evaluate the robustness of modern deep learning models under varying degrees of object occlusion, revealing that even state-of-the-art architectures experience significant performance drops when faced with incomplete visual information [41]. This highlights the ongoing need for innovative approaches in the domain of partial image classification.

2.3. Datasets/Benchmarks for Image Classification

Benchmark datasets play a crucial role in the field of image classification, serving as essential tools for evaluating and comparing the performance of different algorithms. These datasets provide standardized samples that enable researchers and practitioners to measure the effectiveness of their approaches objectively. As illustrated in Table 1, several notable benchmark datasets are commonly utilized in image classification tasks, each offering unique characteristics and challenges that contribute to advancing the development and refinement of classification algorithms.

Table 1. Benchmark datasets for image classification.

Dataset	Number of Instances	Number of Categories	Application	Reference
ImageNet -1K [26]	≈1.4million	1000	Image classification, Object detection, Image segmentation, etc.	https://www.image-net.org/
CIFAR-10 [33]	60000	10	Image classification	https://www.cs.toronto.edu/~kriz/cifar.html
MNIST [34]	70000	10	Image classification, Handwritten digit recognition	http://yann.lecun.com/exdb/mnist/
COCO [35]	330000	80	Image classification, Object detection, Image segmentation, etc.	https://cocodataset.org/
PASCAL-VOC [36]	11530	20	Image classification, Object detection, Image segmentation, etc.	http://host.robots.ox.ac.uk/pascal/VOC/

3. Dataset Development

This section details how we collected and prepared the dataset used for this paper.

Collection Stage. For this experiment, we manually constructed an artificial dataset by gathering 4,000 full-body images across 10 distinct animal classes (bear, cat, cow, dog, donkey, elephant, horse, lion, sheep, and tiger), each comprising 400 images. These images were sourced from multiple websites using a combination of methods, including web searches with

specific keywords and accessing public image datasets.

Preprocessing Stage. During the preprocessing stage, each image was carefully manually cropped to isolate four partial images, as shown in Figure.2, each representing a distinct part of the animal—head, torso, tail, and leg. The cropping rectangles was chosen to fully encompass the target part and not much more. No fixed aspect ratio was imposed, as the contour of each body part varied across images. The background was not removed.

The result was a dataset comprising 16,000 partial images, organized into four subsets corresponding to each body part.

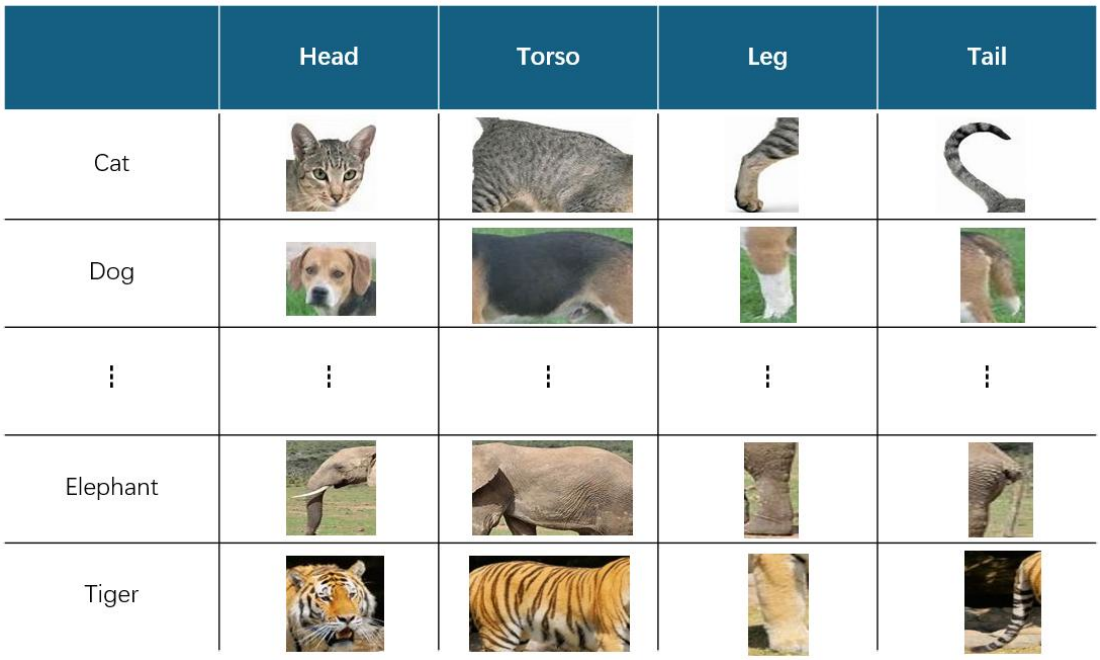

Figure 2. Images of partially visible animals.

4. Methods

This section details the algorithms/methods used in this paper.

4.1. Resnet

Convolutional Neural Networks (CNNs) are a cornerstone in the advancement of deep learning techniques for computer vision. These networks excel in automatically extracting features from images by using convolutional layers in a neural network, leading to breakthrough performances in a wide array of visual tasks, from image classification [21] to object localization [22] and semantic segmentation [23].

Residual Network (ResNet) [12] is a widely recognized and influential CNN architecture that introduced the concept of residual units specifically to tackle the vanishing and exploding

gradient problems frequently encountered during the training of deep neural networks. By incorporating skip connections, ResNet significantly mitigates these gradient-related issues, thereby facilitating the successful training of much deeper networks than was previously feasible. This innovative architectural design not only stabilizes the training process of deep networks but also leads to marked improvements in model performance across various tasks. For the purposes of our experiment, we have chosen to utilize ResNet-18, which serves as a suitable balance between complexity and efficiency while still delivering robust results.

4.2. Swin Transformer

Swin Transformer [13] is a vision modeling approach that incorporates the self-attention mechanism [24] into a deep neural network. This mechanism allows a model to compute relationships between each element in a sequence and all other elements. The architecture of Swin Transformer leverages a hierarchical feature map and a technique known as shifted windowing to synergistically fuse local and global modeling capabilities. This integration not only achieves superior performance on multiple visual tasks but also offers a novel perspective for the design of visual models. We test three Swin-Transformer models: Tiny, Base and Large. They differ in their architectural complexity, particularly in their embedding dimensions, depths and attention heads.

4.3. Large Vision Models

LVMs are a significant leap in the development of AI applied to computer vision. These models, developed using deep learning techniques, leverage vast amounts of visual data to learn rich, complex representations of images. This capability enables them to achieve superior performance on a variety of visual tasks, ranging from basic image classification to more complex applications like scene reconstruction and semantic segmentation. LVMs are transfer-based models, characterized by their large number of parameters, enabling them to capture intricate patterns that are indiscernible to simpler models.

4.3.1. GPT-4

GPT-4 [5] is widely recognized as the most well-known large language model (LLM) available today. In this study, we test its offshoot large vision model (LVM): GPT-4 Vision. Developed by OpenAI and founded on the sophisticated Transformer architecture [24], this advanced model is capable of processing information across diverse modalities, including natural language text and images. By effectively combining its capabilities in both domains, GPT-4 Vision can thoroughly analyze the content and characteristics of images, enabling

precise classification, recognition, and interpretation of visual data in various contexts. This multifaceted approach enhances its utility in applications requiring multimodal understanding.

4.3.2. Gemini

Gemini [8] is a sophisticated family of AI models developed by Google, designed to process an extensive range of modalities, including natural language text, audio recordings, videos, and even programming code. This versatile model family is available in three distinct sizes: Ultra, Pro, and Nano [25], with each variant tailored for different computational scales—from handling highly complex tasks on data centers to optimizing applications that run efficiently on mobile devices. According to the findings presented in [25], Gemini Ultra sets a new standard by advancing the state of the art on 30 out of 32 established benchmarks. For our study, we have chosen to utilize the Gemini-Pro-Vision model, which strikes a balance between performance and computational efficiency, making it well-suited for our specific requirements.

4.3.3. Claude

Claude, a family of transformer-based language models developed by the American AI startup Anthropic [7], represents a significant advancement in natural language processing. Currently on version 3, Claude is available in three scales: Opus, Sonnet and Haiku. Claude's capabilities include advanced reasoning, vision analysis, code generation and multilingual processing. We use Sonnet.

4.3.4. Qwen-VL

Qwen-VL is a series of large-scale vision-language models developed by the Chinese company Alibaba. Starting from a text-only language model, it was trained on a dataset of image-caption-box tuples. Qwen-VL set new records for generalist models of comparable scale across a range of vision benchmarks including image captioning, question answering and visual grounding [10]. We use Qwen-VL-Max.

4.3.5. LLaVA

LLaVA (Large Language and Vision Assistant) is a publicly available large multi-modal model developed by Haotian Liu from the University of Wisconsin-Madison and Chunyuan Li from Microsoft Research through instruction tuning of GPT-4. It effectively combines a vision encoder with a large language model (LLM). The developers conducted extensive testing of LLaVA on two primary tasks: describing images with text, evaluated using GPT-4, and answering multiple-choice questions from Science QA; results are reported in [9]. For our study, we utilize LLaVA version 1.6 to leverage its capabilities in multimodal understanding.

4.3.6. CLIP

CLIP (Contrastive Language-Image Pre-training) is an advanced neural network developed by OpenAI that effectively combines the fields of natural language processing and computer vision. This innovative model is specifically designed for tasks related to image classification, harnessing a diverse dataset consisting of a wide variety of images paired with corresponding natural language captions. One of CLIP's standout features is its high degree of flexibility, which allows users to select specific categories for image classification from which the model can make choices. Notably, CLIP is intended for general use and does not support further training; therefore, our experiment focuses solely on zero-shot results. Radford, Kim et al. [11] conducted comprehensive testing of CLIP across more than 30 existing computer vision datasets, demonstrating that in some instances, its accuracy is comparable to that of models specifically trained on subsets of these datasets. For our analysis, we evaluate four distinct versions of CLIP that were downloaded from the Hugging Face website: vit-base-patch32, vit-large-patch14, plip, and metaclip-b32-400m, each selected for their unique characteristics and capabilities.

5. Prompting Large Vision Models

In the field of computer vision and multimodal AI, prompting techniques are employed to optimize the performance of pre-trained models on specific tasks by guiding the model towards what specifically is desired for output. As the development of large vision models has progressed, the importance of prompt engineering has become increasingly evident. A model’s performance may be sensitive to the specific wording of the prompt, in ways that are highly non-obvious to humans. For text-only tasks, the most advanced models have advanced to the point that the best prompts can often be obtained by simply asking the model to write the prompt itself. However, for multimodal tasks, designing suitable prompts is still very much an active area of research, involving a back-and-forth between the user and the model.

In our study, we manually designed our prompts, using a little bit of trial and error to see which prompts worked best. We aimed to enhance the performance of LVMs, thus narrowing the gap between pre-trained capabilities and specialized task performance. A prompt that works best for one model may not be best for a different model; thus, the prompts we settled on differed slightly between models.

5.1. Prompt Engineering and Experimental Design

In this research, we use a structured prompt engineering strategy for LVMs. The prompts we

used for each of the LVMs are shown in Appendixes A and B. But all designs are based on the following principles:

5.1.1. Structure of the Prompt

To ensure the accuracy of model classification and facilitate automated post-processing, the core prompt used in our experiments is composed of the following components:

Task Orientation and Role-Setting: The model is explicitly instructed to act as an "image classifier" and is informed that the task is part of an academic research study aimed at evaluating its performance. This helps to activate the model's knowledge representations relevant to this specific domain.

Label Set Constraint: The classification scope is strictly defined as a fixed set of labels: [bear, cat, cow, dog, donkey, elephant, horse, lion, sheep, tiger]. By clearly defining this label set, we mitigate the risk of the model outputting irrelevant categories.

Response Formatting: The Mandatory Response Formatting component is designed to enforce output consistency and robustness through two key constraints. First, the model's output is strictly limited to the category name, devoid of any explanatory text or punctuation. Second, the model is prohibited from issuing refusal responses, such as "the image is blurry", compelling it to make its best guess even in situations of uncertainty.

5.1.2. Robustness and Consistency

To accommodate the demands of large-scale dataset evaluation and ensure the reliability of experimental results, the following mechanisms were implemented:

First, the LVM temperature parameter was set to 0.01 to maximize consistency and reproducibility of results.

Second, a robust error-handling process was set up to address potential anomalies during large-scale API calls, such as network fluctuations or content filtering interceptions. If the intended classification process fails, then "unknow" is outputted and we move on to the next image.

5.2. 3-shot setting

To further enhance the performance of Large Vision Models on specific recognition tasks, this study uses a few-shot learning strategy when available. The core of this strategy lies in constructing a multi-turn dialogue structure that includes a System Role, a User Role, and an Assistant Role, thereby providing the model with task guidance and a paradigmatic reference.

During the inference process, the functions of each role are as follows.

System Role: Responsible for establishing task rules. Defines the model's identity as an image classifier and the label constraints it must follow.

User Role: During the 3-shot phase, inputs 3 example images. During the testing phase, inputs the test image.

Assistant Role: During the 3-shot phase, inputs the labels of the 3 images.

Note that LVMs typically have policies constraining usage. These constraints may include restrictions on inputs and outputs. Due to these policies, we encountered instances where models refused to provide answers or offered invalid responses; we counted these as incorrect answers when calculating accuracy. The number of such refusals, and common invalid responses, are detailed in Appendix C.

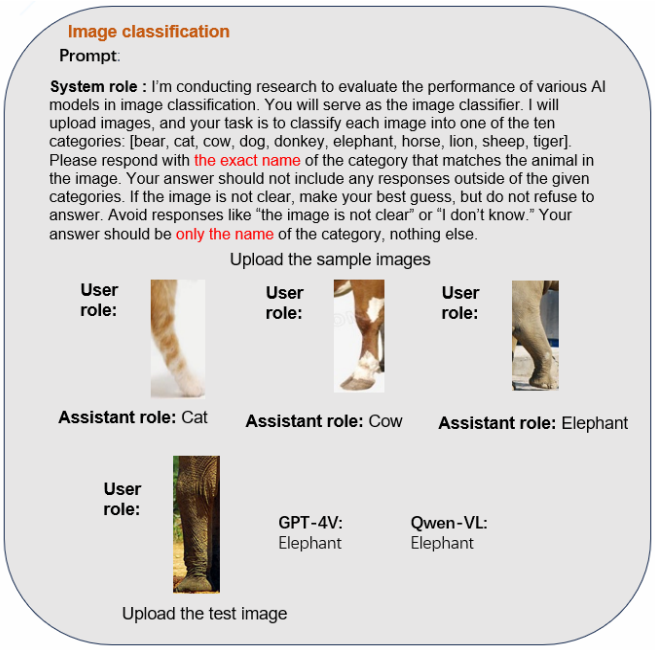

Figure 3. 3-shot experiment with GPT-4 and Qwen-VL.

6. Experimental Setup

This section details our experiments.

The dataset was partitioned into a 7:3 ratio, to form the training and testing sets respectively. That is, the training set consisted of 11200 images, 2800 from each part, and the testing set consisted of 4800 images, 1200 from each part.

6.1. Human Annotation

Comparison with human judgment is common practice for evaluating computer models,

especially the large AI models developed recently [20]. By comparing the accuracy of human annotators with that of the models, we can better understand the strengths and limitations of these models in partial classification tasks. This comparison also provides insights into the potential areas where AI models need further improvement to match or surpass human performance.

We developed a specialized online crowdsourcing platform. The platform was designed to ensure a user-friendly interface, facilitating efficient and accurate responses from the participants. First, volunteers were recruited through a mobile messaging system. Then they were required to register on the platform, to ensure the validity and integrity of the responses. They were informed about the nature of the task and provided with a brief tutorial on how to use the platform. Images from the test set of the dataset were presented one by one to the annotators. Each annotator was shown a series of partial animal images, with each image accompanied by a list of the 10 possible categories. The annotators were required to select the category they believed the partial image belonged to. Volunteers were free to stop whenever they chose. Each annotator was allowed a maximum of 200 annotations to avoid placing excessive weight to any one annotator. This task was considered finished once all the testing images had been annotated exactly once. Figure 4 is a screenshot of the platform.

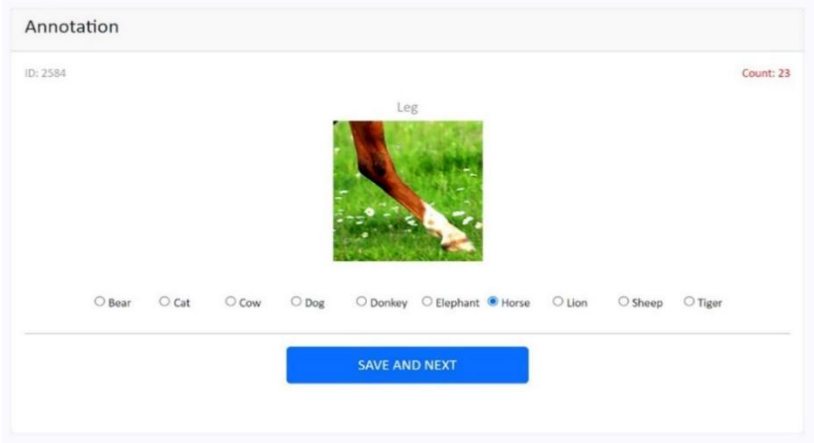

Figure 4. Screenshot of the platform.

The results of human annotation are discussed in detail in Appendix D.

6.2. Model Preprocessing and Hyperparameters

The ResNet and Swin Transformer models that we use are both pre-trained on the ImageNet dataset. For these two models, we resized the input images to 224x224 pixels and normalized using the mean and standard deviation values for the ImageNet dataset, which are [0.485, 0.456, 0.406] and [0.229, 0.224, 0.225] respectively. The ResNet-18 architecture is trained using the

Stochastic Gradient Descent (SGD) optimizer [27], initialized with a learning rate of 0.01 and supplemented with weight decay of 1e-4. The training process utilizes a batch size of 64 and spans 30 epochs. The Swin Transformer models, encompassing the Tiny, Base and Large configurations, are trained using the Adaptive Moment Estimation with Weight Decay (AdamW) optimizer [28] with an initial learning rate of 0.0001 and a weight decay rate of 1e-6. All of these models undergo training with a batch size of 16, covering a total of 30 epochs.

The CLIP models we use are zero-shot models that align image features with text features, eliminating the need for parameter setting or prompts. The CLIP models are merely informed of the available classes.

LVMs are governed by one primary hyperparameter: temperature (T), which is set to 0.01 across all models to minimize output randomness.

To evaluate the Resnet and Swin Transformer models, the full training set was used. On the other hand, the LVMs and CLIP models are intended to be used with little or no further training by the user. For the zero-shot experiments, we of course did not use the training set. For the few-shot experiments, we randomly chose 3 examples from the training set to provide to the models as sample correct classifications. We then tested each example from the testing set as usual. Note that Gemini, GPT-4, Claude and Qwen support such few-shot training; but the other LVMs do not.

7. Results and Conclusions

To ensure a fair comparison of performance across different methods in this task, we categorize the methods into two groups for independent evaluation: traditional deep learning methods and LVMs. We include the results of Human Annotators as a reference standard in both comparison categories.

Traditional deep learning methods (Table 2) are trained on only our dataset, reflecting their learning capability on the target task. Comparing them with the human annotators for this task allows us to assess the potential of these methods to approach or surpass human-level performance after learning from the task-specific data.

Table 2. Performance Comparison of Deep Learning Methods with Human Annotators.

Method	Head only	Torso only	Tail only	Leg only
Human Annotators	0.9348	0.7289	0.5708	0.5263
Swin-Transformer (tiny)	0.9120	0.8320	0.5808	0.4930
Swin-Transformer (base)	0.9420	0.8760	0.6920	0.5370
Swin-Transformer (large)	0.9720	0.9160	0.7660	0.6590

LVMs (Table 3) are pre-trained on a large and diverse corpus. We then evaluate them under

zero-shot or few-shot settings for this task, testing their generalization ability on unseen data. Comparing their results with human annotators helps gauge the "out-of-the-box" performance of such general-purpose models with little or no task-specific training.

Table 3. Performance Comparison of LVMs with Human Annotators.

Method	Head only	Torso only	Tail only	Leg only
Human Annotators	0.9348	0.7289	0.5708	0.5263
Vit-base-patch32	0.8841	0.6267	0.3267	0.3084
Vit-large-patch14	0.9320	0.7575	0.4375	0.4220
Plip	0.6491	0.2050	0.3566	0.1908
Metaclip-b32	0.8750	0.5775	0.3366	0.2675
LLaVA-1.6	0.8083	0.5134	0.2316	0.2461
Gemini-pro-vision (0-	0.9248	0.7705	0.4183	0.3891
shot) (3-shot)	0.9550	0.8767	0.5200	0.4600
GPT4-vision (0-shot)	0.9216	0.6883	0.4358	0.3908
(3-shot)	0.9192	0.7208	0.4567	0.4175
Claude3-sonnet (0-shot)	0.7750	0.4670	0.2760	0.2640
(3-shot)	0.8767	0.6483	0.3383	0.3214
Qwen-vl-max (0-shot)	0.9107	0.8110	0.4758	0.5208
(3-shot)	0.9367	0.8925	0.5575	0.6067

As expected, both human annotators and all methods are least accurate when less informative partial images (e.g., a leg or a tail) are presented.

Based on our results, we draw the following conclusions:

(1) Among all evaluated methods, Swin Transformer (Large) achieves the best overall performance, outperforming both human annotators and all LVMs across all body parts. In particular, it reaches an accuracy of 0.9720 on head images and 0.7660 on tail images, substantially exceeding human performance (0.9348 and 0.5708, respectively).

(2) Among the LVMs, Qwen-vl-max emerges as the best-performing LVM. Under the 3-shot setting, it achieves 0.9367 accuracy on head images, 0.8925 on torso images, 0.5575 on tail images, and 0.6067 on leg images, outperforming human annotators on torso and leg classification. However, it slightly underperforms human annotators on tail classification (0.5575 vs. 0.5708).

(3) Except for Qwen-vl-max, the remaining evaluated large vision models (LVMs) consistently underperform compared to human annotators when it comes to classifying legs and tails, particularly in a zero-shot setting. While human annotators achieve impressive accuracies exceeding 0.52 on leg images and 0.57 on tail images, these LVMs continually fall below these established benchmarks. This discrepancy highlights the challenges faced by current models in accurately recognizing and classifying these specific features, underscoring the superior performance of human judgment in this context. The findings suggest a need for further

refinement and development of LVMs to enhance their capability in such classification tasks.

(4) 3-shot prompting provides a general performance improvement for the evaluated LVMs. Supplying only three examples leads to accuracy gains in most cases, with notable improvements for challenging parts such as legs and tails. For instance, Qwen-vl-max improves from below human accuracy in the zero-shot setting to above human accuracy in the 3-shot setting for leg classification (0.6067 vs. 0.5263).

8. Discussion

This study investigated the capability of large vision models (LVMs) and traditional models to recognize objects under conditions of partial visibility, reflecting real-world challenges like occlusion. By assessing performance across different body parts with varying informativeness, we evaluated the performance gap between these models under zero-shot and three-shot settings for LVMs. This analysis highlights how each model adapts to incomplete visual information.

Several limitations must be acknowledged when interpreting our results. First, the LVMs evaluated in this study were subject to usage policies and safety constraints. We observed instances where models refused to respond or produced invalid outputs, especially under zero-shot settings. These responses were treated as incorrect predictions, potentially underestimating the visual recognition capability of these models. More, while LVM accuracy improved in the 3-shot setting, these results remained sensitive to specific prompt engineering. Factors such as the choice of examples, their sequence, and wording can influence outcomes, meaning the reported performance reflects one prompting setup rather than the absolute ceiling of LVM potential.

Overall, traditional transformer-based architectures, such as the Swin-Transformer, establish a strong performance ceiling, consistently outperforming both humans and large vision models (LVMs) in partial-object recognition tasks. Current LVMs, while flexible and capable of zero-shot inference, still fall short of specialized models when faced with incomplete visual information. Their performance improves significantly under few-shot conditions, indicating they benefit from even small amounts of task-specific adaptation. While LVMs hold considerable promise, their ability to reliably interpret partial images remains limited compared to dedicated vision transformers. Future task-aligned optimization will be essential for them to match or exceed the robustness of specialized models. For tasks involving partially visible objects—such as wildlife monitoring, medical imaging, or surveillance—specialized vision transformers currently provide the most reliable performance available.

Appendix A: Prompts for LVMs (0-shot)

Table 4. Prompts for LVMs(0-shot).

GPT4-vision	Gemini-pro-vision
I'm conducting a study to evaluate the performance of different AI models in image classification for educational and research purposes. You will play the role of an image classifier. I will provide images that showcase parts of an animal, and your task is to identify the animal from a predetermined list of categories. The list consists of 10 categories: 1.dog 2. cat 3. Tiger 4. cow 5. donkey 6. elephant 7. bear8.sheep 9. horse10. lion Please respond with only the exact name of the category that matches the animal in the image. Choose your answer strictly from the provided list, and refrain from including any responses outside of these categories. If the image is not clear you can make your best guess, but avoid responses like "unknown" or "I'm sorry".	I'm conducting a study to evaluate the performance of different AI models in image classification for educational and research purposes. You will play the role of an image classifier. I will provide images that showcase parts of an animal, and your task is to identify the animal from a predetermined list of categories. The list consists of 10 categories: 1. dog 2. cat 3. tiger 4. cow 5. sonkey 6. Elephant 7. bear 8. sheep 9. horse 10. lion Please respond with only the exact name of the category that matches the animal in the image. Choose your answer strictly from the provided list, and refrain from including any responses outside of these categories. If the image is not clear you can make your best guess, but avoid responses like "unknown" or "I'm sorry". For example: User: <upload image> Assistant: cat User: <upload image> Assistant: dog User: <upload image> Assistant: horse etc.

Table 5. Prompts for LVMs(0-shot).

Claude3-sonnet	Qwen-vl-max	LLaVA-1.6
I'm conducting a study to evaluate the performance of different AI models in image classification for educational and research purposes. You will play the role of an image classifier. I will provide images that showcase parts of an animal, and your task is to identify the animal from a predetermined list of categories. The list consists of 10 possible categories： 1. dog 2. cat 3. tiger 4. cow 5. donkey 6. elephant 7. bear 8. sheep 9. horse 10. lion Your answer must be one and only one of the possible categories. You just need to output the category, nothing else. If the image is not clear you can make your best guess, but avoid responses like "Based on the provided image" or "furry".	I'm conducting a study to evaluate the performance of different AI models in image classification for educational and research purposes. You will play the role of an image classifier. I will provide images that showcase parts of an animal, and your task is to identify the animal from a predetermined list of categories. The list consists of 10 categories： 1. dog 2. cat 3. tiger 4. cow 5. donkey 6. elephant 7. bear 8. sheep 9. horse 10. lion Please respond with only the exact name of the category that matches the animal in the image. Choose your answer strictly from the provided list, and refrain from including any responses outside of these categories. If the image is not clear you can make your best guess, but avoid responses like "Based on the image provided" or "I understand".	I'm conducting a study to evaluate the performance of different AI models in image classification for educational and research purposes. You will play the role of an image classifier. I will provide images that showcase parts of an animal, and your task is to identify the animal from a predetermined list of categories. The list consists of 10 categories: 1.dog 2. cat 3. tiger 4. cow 5. donkey 6. elephant 7. bear 8. sheep 9. horse 10. lion Please respond with only the exact name of the category that matches the animal in the image. Choose your answer strictly from the provided list, and refrain from including any responses outside of these categories. If the image is not clear you can make your best guess, but avoid responses like "not clear".

Appendix B: Prompts for LVMs (3-shot)

Table 6. Prompts for LVMs(3-shot).

Gemini-pro-vision	Gemini-pro-vision	Qwen-vl-max	Qwen-vl-max
"I'm conducting research to evaluate the performance of various AI models in image classification tasks. You will serve as the image classifier for this study. I will upload images, and your task is to classify each image into one of the ten specified categories: [bear, cat, cow, dog, donkey, elephant, horse, lion, sheep, tiger]. Please respond with the exact name of the category that best matches the animal depicted in the image. Your answer should not include any responses outside of the provided categories. If the image is unclear or difficult to interpret, make your best guess without refusing to answer. Avoid phrases like 'the image is not clear' or 'I don't know.' Your response should consist solely of the name of the category, nothing more. Thank you for your cooperation." After establishing these detailed instructions, we proceeded to pass three examples to the model. For each example, we uploaded the image as the 'user' and returned the true category of the image as the 'model."	"I'm conducting research to evaluate the performance of various AI models in image classification tasks. You will serve as the image classifier for this study. I will upload images, and your task is to classify each image into one of the ten specified categories: [bear, cat, cow, dog, donkey, elephant, horse, lion, sheep, tiger]. Please respond with the exact name of the category that best matches the animal depicted in the image. Your answer should not include any responses outside of the provided categories. If the image is unclear or difficult to interpret, make your best guess without refusing to answer. Avoid phrases like 'the image is not clear' or 'I don't know.' Your response should consist solely of the name of the category, nothing more. Thank you for your cooperation." After establishing these detailed instructions, we proceeded to pass three examples to the model. For each example, we uploaded the image as the 'user' and returned the true category of the image as the ' assistant.'	"I'm conducting research to evaluate the performance of various AI models in image classification tasks. You will serve as the image classifier for this study. I will upload images, and your task is to classify each image into one of the ten specified categories: [bear, cat, cow, dog, donkey, elephant, horse, lion, sheep, tiger]. Please respond with the exact name of the category that best matches the animal depicted in the image. Your answer should not include any responses outside of the provided categories. If the image is unclear or difficult to interpret, make your best guess without refusing to answer. Avoid phrases like 'the image is not clear' or 'I don't know.' Your response should consist solely of the name of the category, nothing more. Thank you for your cooperation." After establishing these detailed instructions, we proceeded to pass three examples to the model. For each example, we uploaded the image as the 'user' and returned the true category of the image as the ' assistant.'	"I'm conducting research to evaluate the performance of various AI models in image classification tasks. You will serve as the image classifier for this study. I will upload images, and your task is to classify each image into one of the ten specified categories: [bear, cat, cow, dog, donkey, elephant, horse, lion, sheep, tiger]. Please respond with the exact name of the category that best matches the animal depicted in the image. Your answer should not include any responses outside of the provided categories. If the image is unclear or difficult to interpret, make your best guess without refusing to answer. Avoid phrases like 'the image is not clear' or 'I don't know.' Your response should consist solely of the name of the category, nothing more. Thank you for your cooperation." After establishing these detailed instructions, we proceeded to pass three examples to the model. For each example, we uploaded the image as the 'user' and returned the true category of the image as the ' assistant.'

Appendix C: Invalid Responses from LVMs

Table 7. Number of invalid responses from 0-shot.

Model	Test Images	Valid Responses	Invalid Responses	No Response	Total Invalid (No Response + Invalid Response)
GPT4	4800	4550	26	224	250
Gemini		4472	201	127	328
Claude		4630	82	88	170
Qwen		4606	184	10	194
LLaVa		4399	324	77	401

Table 8. Number of invalid responses from 3-shot.

Model	Test Images	Valid Responses	Invalid Responses	No Response	Total Invalid (No Response + Invalid Response)
GPT4	4800	4513	182	105	287
Gemini		4783	0	17	17
Claude		4726	70	4	74
Qwen		4636	52	112	164

Table 9. Examples of invalid responses and their frequencies from 0-shot.

GPT4		Gemini	
Cheetah	3	I'm sorry…	114
Giraffe	8	It's not possible to identify	3
Zebra	8	The image is too blurry to…	6
(No animal stated)	2	Wolf	4
Goat	4	Snake	10

Table 10. Examples of invalid responses and their frequencies from 0-shot.

Claude		Qwen	
(No animal stated)	65	(Image not recognized)	83
Giraffe	2	Zebra	7
Rabbit	2	Fox	2
Snake	2	Goat	3
Zebra	3	Ok.	75

Table11. Examples of invalid responses and their frequencies from 0-shot.

LLava	
The image provided is not clear enough to…	246
Not clear	21

Table 12. Examples of invalid responses and their frequencies from 3-shot.

GPT4	
The image is not clear…	118
(None of the ten categories)	20
Zebra	6
Buffalo	3

Table 13. Examples of invalid responses and their frequencies from 3-shot.

Claude		Qwen	
Goat	27	Bull	7
I apologize…	11	Leopard	2
Pig	6	Zebra	16
Zebra	5	Snake	7
Rhino	2	Mushroom	2
Leopard	2	Deer	2

Appendix D: LVM Costs

Open-source algorithms are freely available for public use, incurring no additional costs beyond the operating expenses associated with the user's computing resources. However, it is important to note that the following four large language models (LVMs) are not open source and do impose charges for usage; these fees are typically proportional to the volume of input and output processed. As of July 11, 2024, we have compiled a list of their pricing structures, which reflects the varying costs associated with their deployment in practical applications. This information will provide potential users with valuable insights into the financial implications of utilizing these proprietary models.

Table 14. Prices for Closed Source LVMs

Model	Input Price (per 1000 tokens)	Output Price (per 1000 tokens)	Source
GPT-4-vision	$ 0.06	$ 0.12	https://azure.microsoft.com/zh-cn/pricing/details/cognitive-services/openai-service/
Qwen-vl-max	$ 0.003	$ 0.003	https://help.aliyun.com/document_detail/2712568.html?spm=a2c4g.2712587.0.0.7dd53809r05POG
Claude-sonnet	$ 0.003	$0.015	https://docs.anthropic.com/zh-CN/docs/models-overview
Gemini-pro-vision	$0.0035	$0.0105	https://ai.google.dev/pricing?hl=zh-cn

Appendix E: Human Annotation Results

Unlike machines, the accuracy of human annotators is influenced by a variety of factors, including motivation, attention, and fatigue. As a result, we anticipate that human annotation will exhibit the highest degree of variability in performance when compared to all existing machine methods. This inherent variability makes our reported accuracy particularly susceptible to questioning and scrutiny. In this section, we aim to provide a comprehensive analysis along with additional information that will help assess and better understand this variability, ultimately shedding light on the complexities involved in human annotation

processes.

A total of 83 annotators participated in the study. Among these annotators, the highest accuracy achieved was an impressive 90.48%, while the lowest recorded accuracy was significantly lower at 40.00%. Although this represents a substantial range of performance, it cannot be interpreted at face value due to several critical factors. Specifically, the difficulty of the images presented varied considerably, with heads being the easiest to classify and legs and tails proving to be much more challenging. Additionally, the number of annotations attempted by each annotator also varied widely, ranging from a minimum of just 6 to a maximum of 200. These factors contribute to the nuanced understanding of accuracy levels among human annotators and highlight the importance of contextualizing these results.

To investigate how accuracy may vary depending on the specific group of human annotators selected, we randomly divided the 83 annotators into four groups consisting of either 20 or 21 members each. We then calculated the accuracies for each individual group to assess this variability in human performance. Importantly, the assignment of annotators into these groups was conducted independently of the total number of annotations each group was responsible for, ensuring a fair evaluation of their performance. The results of this analysis are presented below, providing valuable insights into the impact of group selection on annotation accuracy.

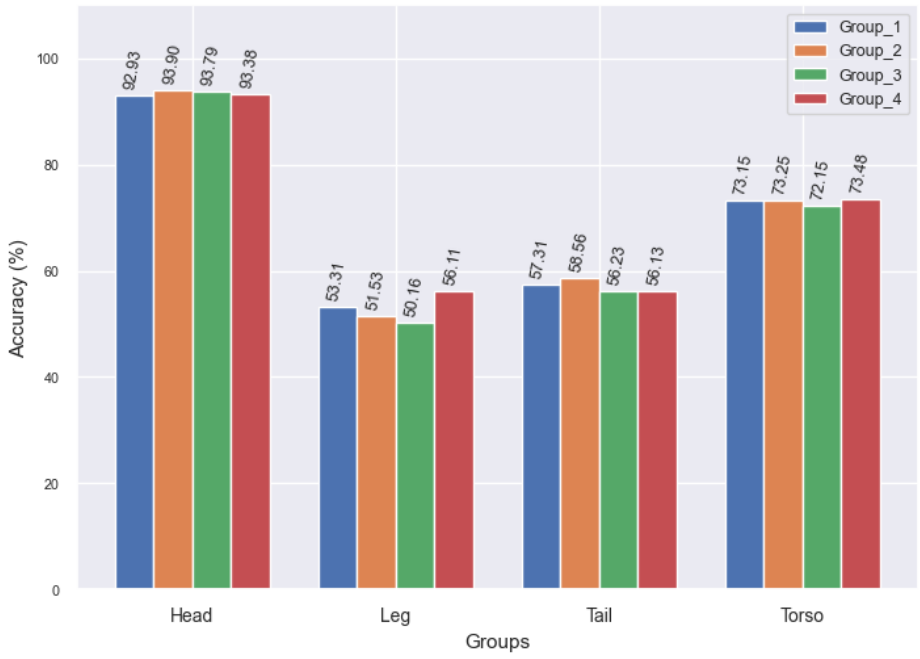

Figure 6. Variation in Annotator Accuracy across Groups

To test how accuracy may depend on which group of human annotators is selected, we randomly assigned the 83 annotators into 4 groups of 20 or 21 and calculated the accuracies for

each group. This being a test of variation in human performance, the assignment into groups was done regardless of how many or how few total annotations each group was responsible for. The results are shown below.

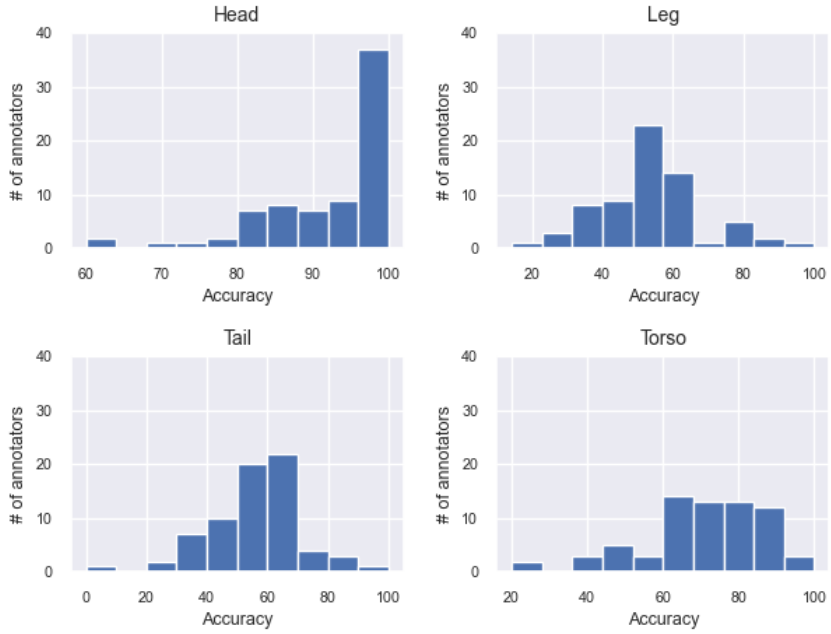

Figure 7. Variation in Annotator Accuracy across Individuals

Declarations

Availability of supporting data

The datasets used and / or analyzed during the current study are available from the first author on reasonable request.

Competing interests

The authors declare no conflict of interest.

Funding

Not applicable.

Authors' contributions

Dong, Ziwen wrote the necessary computer code, ran the algorithms and managed the results, and wrote most of the initial draft of this manuscript. Ijazul, Haq created and managed the human annotation platform, supervised the LVM prompt-writing, and wrote part of the initial draft of this manuscript. Huang, Shan helped check Dong, Ziwen's computer code. Du, Jin supervised this project and edited this manuscript.

Acknowledgements

The authors thank Du, Ruxu for the idea and motivation behind this project.

References

[1] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. A*dvances in neural information processing systems*, 33, 1877-1901.

[2] Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., ... & Fiedel, N. (2023). Palm: Scaling language modeling with pathways. *Journal of machine learning research,* 24(240), 1-113.

[3] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. A., Lacroix, T., ... & Lample, G. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:*2302.13971.

[4] Team, V. (2023). Vicuna: An open-source chatbot impressing gpt-4 with 90% chatgpt quality. *Vicuna: An open-source chatbot impressing gpt-4 with,* 90.

[5] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., ... & McGrew, B. (2023). *Gpt-4 technical report. arXiv preprint arXiv*:2303.08774.

[6] Caffagni, D., Cocchi, F., Barsellotti, L., Moratelli, N., Sarto, S., Baraldi, L., ... & Cucchiara, R. (2024). The revolution of multimodal large language models: A survey. *Findings of the association for computational linguistics: ACL 2024*, 13590-13618.

[7] Sparkman, M., & Witt, A. (2025). Claude AI and literature reviews: An experiment in utility and ethical use. *Library Trends*, 73(3), 355-380.

[8] Pichai, S., & Hassabis, D. (2023). Introducing Gemini: Our largest and most capable AI model. *Google. https://blog.google/technology/ai/google-gemini-ai/*

[9] Liu, H., Li, C., Wu, Q., & Lee, Y. J. (2023). Visual instruction tuning. *Advances in neural information processing systems*, 36, 34892-34916.

[10] Team, Q. (2023). Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. *arXiv preprint arXiv*:2308.12966.

[11] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. *In International conference on machine learning* (pp. 8748-8763).

[12] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[13] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. *In Proceedings of the IEEE/CVF international conference on computer vision* (pp. 10012-10022).

[14] Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W. X., & Wen, J. R. (2023). Evaluating object hallucination in large vision-language models. *In Proceedings of the 2023 conference on empirical methods in natural language processing* (pp. 292-305).

[15] Liu, Y., Li, Z., Huang, M., Yang, B., Yu, W., Li, C., ... & Bai, X. (2024). Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12), 220102.

[16] Xu, P., Shao, W., Zhang, K., Gao, P., Liu, S., Lei, M., ... & Luo, P. (2024). Lvlm-ehub: A comprehensive evaluation benchmark for large vision-language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(3), 1877-1893.

[17] Yu, W., Yang, Z., Li, L., Wang, J., Lin, K., Liu, Z., ... & Wang, L. (2023). Mm-vet: Evaluating large Multimodal models for integrated capabilities. *arXiv preprint arXiv*:2308.02490.

[18] Li, C., Liu, H., Li, L., Zhang, P., Aneja, J., Yang, J., ... & Gao, J. (2022). Elevater: A benchmark and toolkit for evaluating language-augmented visual models. *Advances in Neural Information Processing Systems*, 35, 9287-9301.

[19] Yin, Z., Wang, J., Cao, J., Shi, Z., Liu, D., Li, M., ... & Ouyang, W. (2023). Lamm: Language-assisted multi-modal instruction-tuning dataset, framework, and benchmark. *Advances in Neural Information Processing Systems*, 36, 26650-26685.

[20] Ziems, C., Held, W., Shaikh, O., Chen, J., Zhang, Z., & Yang, D. (2024). Can large language models transform computational social science?. *Computational Linguistics*, 50(1), 237-291.

[21] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

[22] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv*:1312.6229.

[23] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431-3440).

[24] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems, 30.*

[25] Team, G., Anil, R., Borgeaud, S., Alayrac, J. B., Yu, J., Soricut, R., ... & Blanco, L. (2023). Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv*:2312.11805.

[26] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. *In 2009 IEEE conference on computer vision and pattern recognition* (pp. 248-255). Ieee.

[27] Bottou, L. (2010, September). Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers (pp. 177-186). *Heidelberg: Physica-Verlag HD*.

[28] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv*:1412.6980.

[29] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv*:1409.1556.

[30] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

[31] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv*:1704.04861.

[32] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv*:2010.11929.

[33] Krizhevsky, A. , & Hinton, G. (2009). Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 1(4).

[34] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (2002). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

[35] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). *Cham: Springer International Publishing.*

[36] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2), 303-338.

[37] Jeong, J. , Lee, J. Y. , & Son, Y. (2018). A study of partial image classification of vehicles using finger gestures. *International Journal of Grid and Distributed Computing*, 11, 111-122.

[38] Zhao, G., Zhang, C., Wang, X., Lin, B., & Yan, F. (2024). PMANet: Progressive multi-stage attention networks for skin disease classification. *Image and Vision Computing*, 149, 105166.

[39] Cen, F., & Wang, G. (2019). Boosting occluded image classification via subspace decomposition-based estimation of deep features. *IEEE transactions on cybernetics*, 50(7), 3409-3422.

[40] Cen, F., Zhao, X., Li, W., & Wang, G. (2021). Deep feature augmentation for occluded image classification. *Pattern Recognition*, 111, 107737.

[41] Kassaw, K., Luzi, F., Collins, L. M., & Malof, J. M. (2025). Are deep learning models robust to partial object occlusion in visual recognition tasks?. *Pattern Recognition*, 112215.

Online ISSN: 3104-9338
Print ISSN: 3104-932X

A Review of Stock Index Forecasting Methods from ARIMA to Time-Series Foundation Models

Li Su*

* Chengdu University of Information Technology

Received: April 15, 2026

Revised: April 20, 2026

Accepted: April 25, 2026

Published online: April 26, 2026

To appear in: *International Journal of Advanced AI Applications,* Vol. 2, No. 5 (May 2026)

* Corresponding Author: Author Name (suli3607@foxmail.com)

Abstract. Stock index forecasting has evolved from linear statistical baselines to hybrid deep neural architectures and, more recently, to large-scale time-series foundation models. This review synthesizes the development path represented by the supplied literature, covering ARIMA, GARCH, and VAR models; classical machine learning methods such as random forests and boosting; recurrent, convolutional, and attention-based deep learning models; decomposition-driven hybrids; selective state space models; and emerging large-model approaches for time-series analysis. The review is organized around the inductive biases that different model families impose on financial data, with special attention to nonstationarity, volatility clustering, multimodal information fusion, and distribution shift. Compared with generic forecasting domains, stock index prediction places stronger demands on robustness, interpretability, and economic usefulness because signal-to-noise ratios are low and model errors can be magnified by trading decisions. Across the surveyed studies, no single architecture dominates all settings; instead, performance depends on how well a method aligns with data frequency, exogenous information, market regime, and evaluation objective. The review concludes that future progress is likely to come from financially informed hybrid systems, stronger benchmark design, and better integration between domain-specific supervision and foundation-model pretraining.

Keywords: *Stock Index Forecasting; Financial Time Series; Deep Learning; Transformers; State Space Models; Foundation Models*

1. Introduction

Forecasting stock indexes remains one of the most demanding tasks in applied time-series

analysis because market prices aggregate macroeconomic information, firm-level expectations, policy shocks, liquidity conditions, and investor behavior in a continuously changing environment. Unlike many engineering forecasting tasks, financial prediction must cope with weak explanatory signals, abrupt regime shifts, nonlinear dependence, volatility clustering, and the possibility that model-aware traders alter the very dynamics being modeled. For this reason, stock index forecasting has long served as both a practical problem and a stress test for forecasting methodology. The recent comparison of ARIMA models across the S&P 500, FTSE, and SSEC by Xu [33] illustrates that even straightforward autoregressive baselines can still provide useful diagnostic insight when the purpose is to understand persistence, trend structure, and market-specific differences rather than to claim universal predictive superiority.

The historical foundation of the field is statistical. Bollerslev's generalized autoregressive conditional heteroskedasticity model [2] formalized time-varying volatility in a way that remains central to risk-aware financial modeling, while Sims's vector autoregressive perspective [28] established a flexible multivariate framework for studying dynamic interactions without imposing overly restrictive structural assumptions at the outset. These models made explicit two enduring lessons for stock forecasting research: first, mean dynamics and variance dynamics should not be conflated; second, model usefulness depends on whether the analyst seeks point prediction, volatility estimation, structural interpretation, or policy-sensitive scenario analysis. Even when later machine learning methods outperform classical baselines on accuracy metrics, the interpretability and diagnostic clarity of statistical models remain indispensable.

The next methodological wave imported ideas from machine learning and sequence modeling. Random forests [3] and boosting [10] showed that nonlinear prediction could be improved by ensembling relatively simple learners, especially when inputs were expanded through technical indicators, lagged returns, or handcrafted macro features. Recurrent neural networks, especially long short-term memory networks [14] and gated encoder-decoder variants [6], then offered a more flexible approach to temporal dependence by learning representations directly from sequences rather than relying entirely on manually engineered summary statistics. At a broader methodological level, Bai et al. [1] challenged the assumption that recurrence is always the natural tool for sequence modeling, while the transformer architecture of Vaswani et al. [29] made attention-based sequence representation a dominant paradigm across machine learning.

Finance-specific studies reflect this progression. Fischer and Krauss [9] demonstrated that LSTM networks could extract useful predictive structure from cross-sectional financial data,

while CNNpred [15] broadened the input space by combining price, technical, and cross-market variables in a convolutional framework. Subsequent comparative and hybrid studies [11,12,13,19,21,23,24,27,34] explored richer combinations of recurrent units, convolutions, attention mechanisms, decomposition techniques, sentiment information, and cross-scale feature fusion. These works collectively suggest that financial forecasting accuracy often depends less on raw model depth than on whether the architecture matches the heterogeneous temporal scales, exogenous drivers, and nonstationary distributions of market data.

The architecture of a LSTM is illustrated in Figure 1, which shows the cell state pathway and the three gating mechanisms—forget, input, and output—that enable the network to selectively retain or discard information over long temporal horizons.

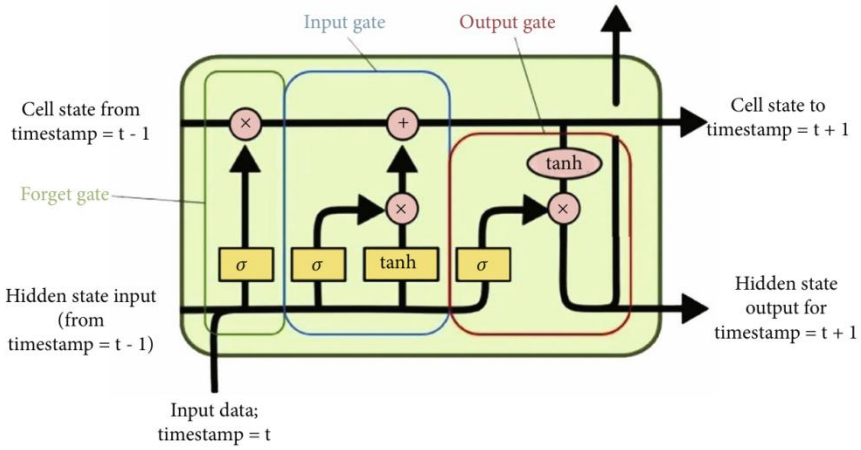

Figure 1. Architecture of a LSTM.

A further acceleration has occurred in recent years with the arrival of general-purpose time-series architectures and foundation-model thinking. Attention modules such as CBAM [31], normalization strategies such as RevIN [18], modern forecasting architectures including TimesNet [32], PatchTST [25], TiDE [7], TSMixer [5], iTransformer [20], TimeMixer [30], and SAMformer [16], and large-model approaches such as Time-LLM [17], TEMPO [4], Timer [22], decoder-only foundation models [8], and Lag-Llama [26] have expanded the design space well beyond the recurrent-versus-convolutional debate. This review examines that broader landscape from the standpoint of stock index forecasting, emphasizing inductive bias, data requirements, robustness under shift, and practical deployability rather than leaderboard-style comparison alone.

The review deliberately places finance-specific papers alongside general-purpose forecasting models because the two literatures are now tightly connected. Many of the newest stock prediction studies are not built from wholly finance-native architectures; instead, they adapt

design ideas first validated on broader time-series benchmarks and then retrofit them to market data. This creates both an opportunity and a methodological risk. The opportunity is that finance can benefit from rapid progress in sequence modeling without rediscovering every architectural idea locally. The risk is that a model may inherit benchmark success from domains whose statistical properties differ sharply from those of financial returns. A useful survey must therefore examine not only what a model is, but also why its original design assumptions may or may not survive transfer into stock index forecasting.

2. From Statistical Benchmarks to Classical Machine Learning

2.1. Statistical Models: ARIMA, GARCH, and VAR

ARIMA remains the canonical entry point for financial forecasting because it imposes a transparent decomposition of a series into autoregressive, differencing, and moving-average components. Its main strength is not that it captures every market irregularity, but that it provides a disciplined baseline against which more complex models must justify their additional complexity. Xu [33] used ARIMA to compare three major equity indexes with different institutional and regional characteristics, reinforcing the practical value of model identification, stationarity testing, and residual diagnostics before moving to heavier architectures. In review terms, ARIMA serves as the benchmark for evaluating whether newer models genuinely learn nonlinear temporal structure or merely exploit sample-specific drift that can disappear out of sample.

However, stock indexes rarely behave like homoskedastic linear processes. Periods of calm and turbulence alternate, and the conditional variance often exhibits stronger persistence than the conditional mean. Bollerslev's GARCH model [2] addressed this issue by modeling volatility as its own dynamic process, thereby improving both risk estimation and the interpretation of forecast uncertainty. For stock index research, this distinction is crucial because a model that predicts returns poorly may still be useful for forecasting volatility, tail risk, or interval width. Many later neural architectures implicitly attempt to capture similar changing variance patterns, but they often do so without the explicit probabilistic structure that made GARCH attractive to financial econometrics.

VAR modeling introduced by Sims [28] is equally important when the objective is to forecast an index using interacting macroeconomic or cross-market variables rather than the target series alone. Its contribution lies in recognizing that financial prices do not evolve in isolation; exchange rates, interest rates, commodity prices, and foreign market indexes may all transmit

information through lagged interactions. Although modern deep learning systems can represent multivariate dependence more flexibly, the VAR framework still supplies the conceptual basis for exogenous-variable modeling, impulse-style reasoning, and scenario-aware forecasting. For review purposes, VAR is best understood as an early formalization of multivariate temporal coupling rather than merely a classical baseline.

Taken together, ARIMA, GARCH, and VAR establish three baseline questions that remain relevant in modern stock index forecasting. Is the predictive task primarily about linear mean reversion or trend continuation? Does volatility dynamics carry more stable information than level dynamics? And how much of the relevant signal is endogenous to the target series versus imported from correlated variables? Any new model family that cannot answer these questions more effectively than statistical benchmarks may offer sophistication without decision value. That is why the best contemporary studies still compare against econometric baselines even when the final winning architecture is nonlinear.

There is also a broader methodological virtue in preserving statistical baselines within modern experimental pipelines. Because ARIMA, GARCH, and VAR have relatively transparent failure modes, they help reveal whether a forecasting problem is intrinsically weak-signal or whether the issue lies in model misspecification. If a sophisticated deep architecture cannot outperform a carefully tuned linear or heteroskedastic benchmark, the result may indicate not that the deep model is poorly implemented, but that the available data do not support the complexity being imposed. In finance, where overfitting is easy and economic regimes are unstable, this diagnostic role is as important as the baseline score itself.

2.2. Classical Machine Learning and Feature-Centric Forecasting

Classical machine learning shifted the emphasis from explicit stochastic assumptions to flexible nonlinear decision boundaries. Random forests [3] are attractive in financial forecasting because they handle mixed feature types, nonlinear interactions, and moderate feature redundancy without heavy preprocessing, while still supporting variable-importance style diagnostics. Boosting methods such as AdaBoost [10] provide a different route to stronger prediction by repeatedly reweighting difficult examples and combining weak learners into a high-capacity ensemble. In stock index settings, these models are especially useful when the researcher constructs features from multiple horizons, technical indicators, macro releases, and sentiment summaries, because the feature space itself becomes the main source of predictive power.

The limitation of this feature-centric paradigm is that performance depends heavily on manual representation design. If the handcrafted features fail to expose the relevant temporal dependency, even strong nonlinear learners cannot recover the missing structure. Moreover, tree ensembles and boosted learners do not natively distinguish between stable long-run information and fast transient shocks unless those distinctions are already encoded in the features. For stock indexes, where the same raw price path may support multiple meaningful temporal views such as intraday momentum, weekly reversal, and crisis-regime persistence, this becomes a serious constraint. The transition to deep learning can therefore be seen not as a rejection of classical machine learning, but as an attempt to automate representation construction while retaining nonlinear predictive capacity.

Even so, classical machine learning continues to play a vital role in the landscape of modern literature on stock index forecasting. Firstly, it establishes competitive baselines that are often significantly stronger than naive linear models, providing a valuable point of reference for assessing more complex methodologies. Secondly, classical techniques remain particularly practical in low-data settings where deep learning models may be prone to overfitting due to insufficient training data. Thirdly, these traditional methods can be effectively integrated into hybrid workflows; for example, decomposition techniques can be employed to produce multi-scale components, allowing simpler learners to be assigned to different frequency bands for enhanced performance. For a comprehensive review of stock index forecasting, the main lesson learned is that model choice should thoughtfully reflect both the scale and representation of the available data: when domain knowledge leads to the development of highly informative features, classical ensemble methods can still prove challenging to outperform on a robustness-adjusted basis. This underlines the enduring relevance of classical approaches in contemporary predictive analytics and their potential to complement more advanced modeling techniques.

Classical machine learning methods highlight an important issue that continues to be relevant for contemporary deep learning models: the choice of target formulation. Some studies focus on predicting raw index levels, while others emphasize forecasting returns, and yet others simplify the task to direction classification. Techniques such as tree ensembles and boosting methods make this distinction explicit because they can be effectively trained as either regressors or classifiers with relatively minor architectural adjustments. The same degree of flexibility is indeed possible in the realm of deep learning; however, the existing literature does not always clearly state whether a given model is optimized specifically for error minimization, directional hit rate, or for facilitating downstream portfolio decisions. From an academic review

standpoint, this ambiguity is significant because architectures should only be compared when they are addressing the same forecasting problem under comparable loss functions. Consequently, researchers must exercise caution in their comparisons to ensure validity and relevance in performance evaluations.

3. Deep Neural Forecasting Architectures in Finance

3.1. Recurrent Sequence Models and Their Financial Adoption

Deep learning entered stock forecasting through recurrent sequence models because recurrence offers a direct mechanism for processing ordered observations and retaining temporal context. LSTM [14] addressed vanishing gradients through gated memory, making it possible to learn longer-range dependencies than standard recurrent networks. GRU-style encoder-decoder models [6] simplified the recurrent machinery while preserving gating behavior, and they encouraged the view that sequence-to-sequence learning could be adapted beyond language processing. In finance, Fischer and Krauss [9] provided an influential demonstration that LSTM networks could learn useful predictive regularities from stock-related inputs, helping establish recurrent neural networks as credible tools for market prediction rather than purely experimental imports from other domains.

The appeal of recurrent models in stock index forecasting is straightforward and well-founded: financial markets exhibit path dependence, meaning that the relevance of recent information often hinges on what transpired earlier in the same sequence. However, it is essential to note that recurrence is not inherently well matched to all financial tasks across various contexts. For instance, challenges such as training instability, the computational cost associated with sequential processing, and difficulties in effectively capturing very long contexts can become significant drawbacks when researchers transition from analyzing low-frequency end-of-day data to working with larger multivariate time windows. This limitation is one reason why the work by Bai et al. [1] became so pivotal in the broader literature: their empirical comparison demonstrated that generic convolutional sequence models could rival or even surpass recurrent networks on a wide range of sequence tasks. This finding served to weaken the prevailing belief that Recurrent Neural Networks (RNNs) were the inevitable default choice for handling time-dependent data, encouraging further exploration of alternative modeling approaches in financial forecasting.

Figure 2 depicts the architecture of a standard RNN, where the recurrent hidden-state connections allow previous time-step information to flow forward, providing the model with a

basic form of memory for sequential data.

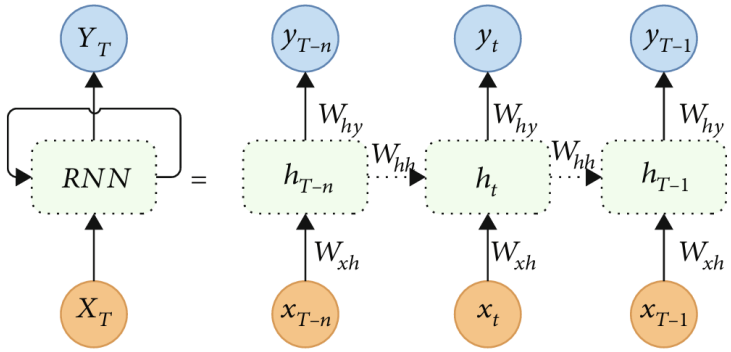

Figure 2. Architecture of RNN.

From a review perspective, recurrent models should be seen as a transitional breakthrough rather than the endpoint of deep financial forecasting. They established the feasibility of representation learning on raw temporal inputs, reduced dependence on manual lag design, and opened the door to richer hybrids with convolution, attention, and exogenous encoders. At the same time, their limitations made it natural for later work to search for architectures with more explicit multi-scale extraction, parallel computation, or stronger handling of distribution shift. The subsequent finance literature can therefore be interpreted as a sequence of attempts to preserve the temporal sensitivity of recurrence while addressing its computational and representational bottlenecks.

3.2. Convolutional and Attention-Augmented Financial Models

One of the earliest finance-specific steps beyond plain recurrence was to diversify the input representation. CNNpred [15] is significant because it used convolutional processing over a diverse set of variables, emphasizing that stock market prediction is rarely a univariate task in practice. By combining price-related features with broader market information, CNNpred implicitly treated forecasting as a structured representation problem rather than a simple autoregression problem. This insight remains central today: accuracy gains often arise not from replacing one sequence backbone with another in isolation, but from expanding the information channels the model can align across time.

Comparative studies help separate durable patterns from architecture-specific enthusiasm. The comparative review by He, Zhang, and Por [13] reflects an important stage in the literature, where different deep learning families are evaluated not simply by peak accuracy but by their sensitivity to data characteristics, parameterization, and training setup. Such comparative work matters because the stock forecasting literature is otherwise vulnerable to fragmented claims

across different markets, horizons, and preprocessing pipelines. A finance review that only reports the best result of each paper can easily exaggerate methodological progress; comparative studies instead remind us that many reported improvements are conditional on particular datasets and design choices.

As shown in Figure 3, a BiLSTM extends the unidirectional LSTM by processing the input sequence in both forward and backward directions, enabling the network to capture context from both past and future time steps simultaneously.

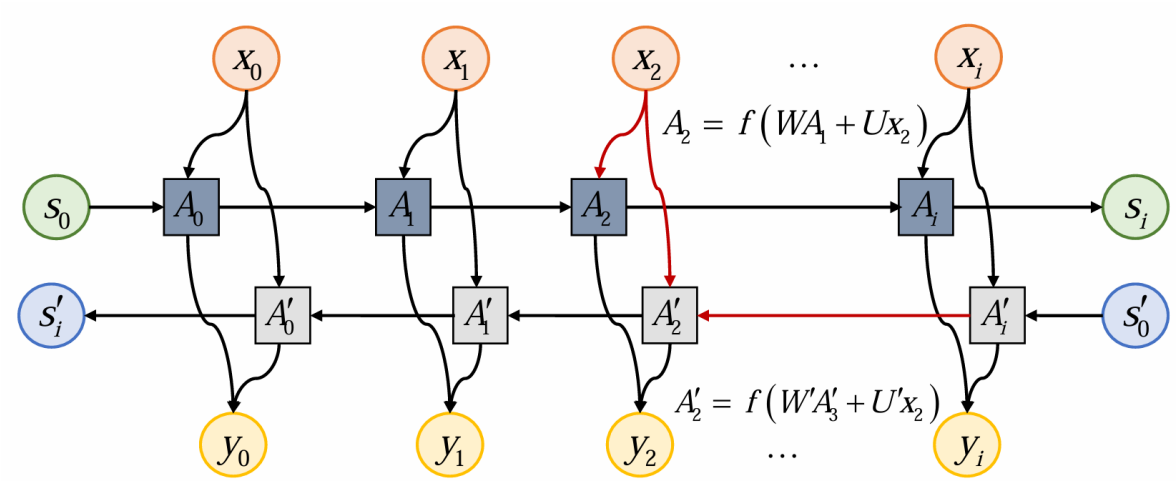

Figure 3. Architecture of a BiLSTM.

Attention-enhanced recurrent hybrids represent another significant theme in the evolution of forecasting models. In their recent work, Zhang, Ye, and Lai [34] innovatively combined Convolutional Neural Networks (CNN), Bidirectional Long Short-Term Memory networks (BiLSTM), and attention mechanisms in an effort to capture local patterns, bidirectional temporal structures, and adaptive feature weighting within a single cohesive pipeline. Similarly, Mu et al. [23] proposed a spatiotemporal attention BiLSTM model specifically aimed at improving stock index prediction. This reflects a growing recognition among researchers that valuable signals may be distributed across both temporal positions and feature dimensions, necessitating more sophisticated modeling techniques. These advanced architectures are conceptually appealing because stock indexes are shaped by layered dependencies: short-term fluctuations, medium-term trend segments, and cross-variable interactions do not contribute equally at every time step. As a result, employing a uniform hidden representation can prove to be suboptimal for accurately capturing the complexities of financial data. By integrating these various elements, such models hold promise for enhancing predictive performance and providing deeper insights into market dynamics.

Figure 4 presents the internal structure of a long short-term memory neural network, highlighting the cell state and the three gating mechanisms that together address the vanishing gradient problem inherent in vanilla recurrent architectures.

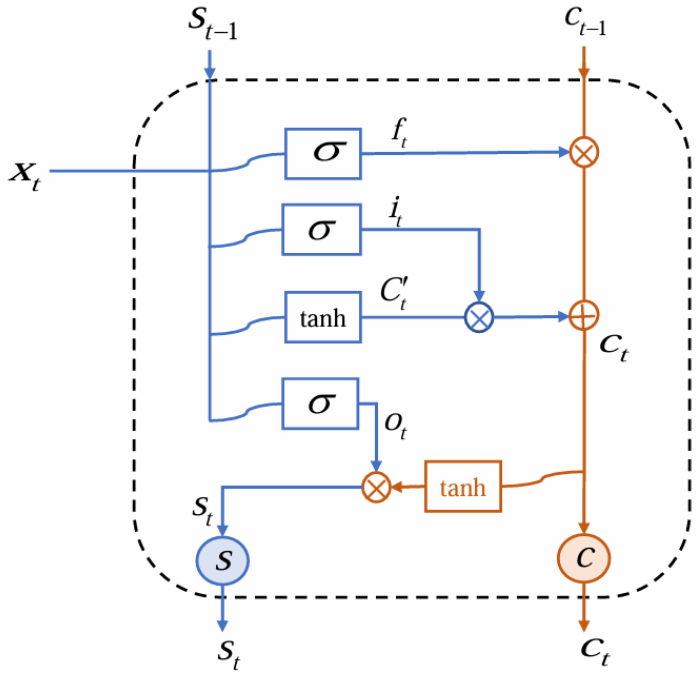

Figure 4. Long short-term memory neural network.

The incorporation of exogenous textual information is particularly notable in the innovative news-driven index prediction framework developed by Liu, Ge, and Gu [20]. By effectively combining a trellis-style temporal network with sentiment attention mechanisms, this model addresses a core weakness inherent in traditional price-only forecasting approaches: many market moves are influenced less by endogenous chart patterns and more by external narrative shocks, such as significant news events. Furthermore, news-based systems compel the field to confront complex alignment issues between the timing of textual information, the corresponding market response windows, and the challenges associated with noisy sentiment extraction. As a result, these advanced models are valuable not only for their potential gains in predictive accuracy but also for expanding the conceptual scope of stock index forecasting. They shift the focus from mere numerical extrapolation to a more nuanced form of multimodal event-aware reasoning that incorporates various types of information, thereby enhancing our understanding of market dynamics and investor behavior. This progression signifies an important step toward integrating qualitative insights into quantitative financial analysis.

The STBL architecture is visualized in Figure 5, where the spatial and temporal branches are combined through a fusion module to jointly model cross-sectional and temporal dependencies

in financial time-series data.

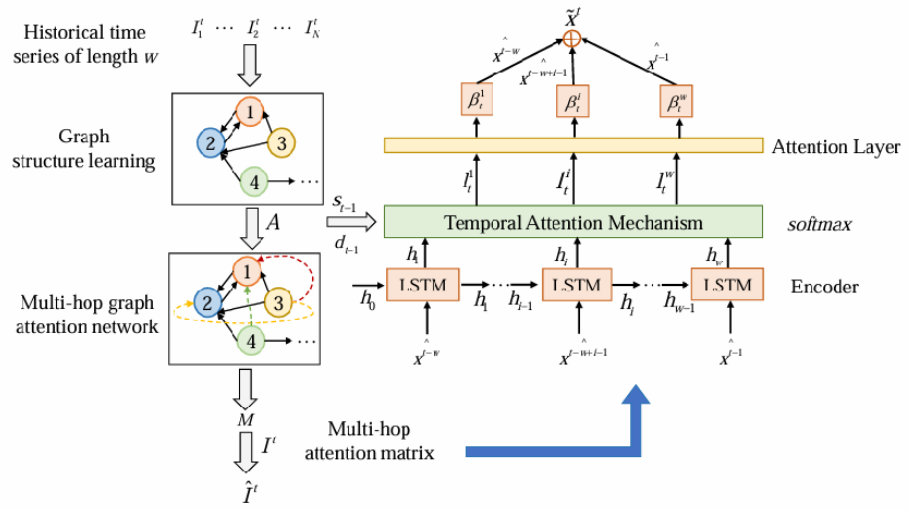

Figure 5. Visualization of the STBL architecture.

Attention-augmented financial models nonetheless require careful interpretation. In many papers, attention improves performance because it acts as a flexible weighting mechanism over already useful representations, not because it provides transparent economic explanations. A high attention score on a time step or feature channel should not automatically be read as causal market importance. This caveat is particularly relevant in stock index forecasting, where correlated signals, common shocks, and preprocessing artifacts can all produce seemingly meaningful weighting patterns. Future work should therefore distinguish more clearly between attention as an optimization device and attention as an interpretability claim.

3.3. Hybrid Decomposition Models and Selective State Spaces

A recurring conclusion in financial forecasting is that one architecture rarely handles every temporal scale equally well. This observation has motivated hybrid systems that explicitly decompose or route information before prediction. Ge [11], for example, proposed a hybrid model for forecasting S&P 500 and CSI 300 futures prices, reflecting the broader tendency to combine complementary modules rather than rely on a single backbone. The logic is economically sensible: low-frequency trend, medium-frequency cyclical movement, and high-frequency noise may each demand different inductive biases, and forcing one monolithic model to resolve all of them at once can reduce robustness.

Decomposition-based hybrids represent one of the most active branches of recent stock index research. Li et al. [19] combined CEEMDAN with a TCN-GRU-CBAM forecasting stack, while Mutinda and Geletu [24] used CEEMDAN with LSTM and BPNN components in an

ensemble decomposition model. Although the exact modules differ, the underlying hypothesis is shared: financial series become easier to model after they are separated into components with more homogeneous temporal properties. This is a practical response to nonstationarity because decomposition can isolate oscillatory behavior, slow trend, and residual noise before deep learning is applied. The price paid for this improvement is a more elaborate pipeline, more hyperparameters, and additional opportunities for data leakage if decomposition is not performed strictly within the training split.

Figure 6 illustrates the construction framework of the CEEMDAN-TCN-GRU-CBAM model, in which the original price series is first decomposed by CEEMDAN, then each sub-series is independently modeled by a TCN-GRU encoder with CBAM attention before final reconstruction.

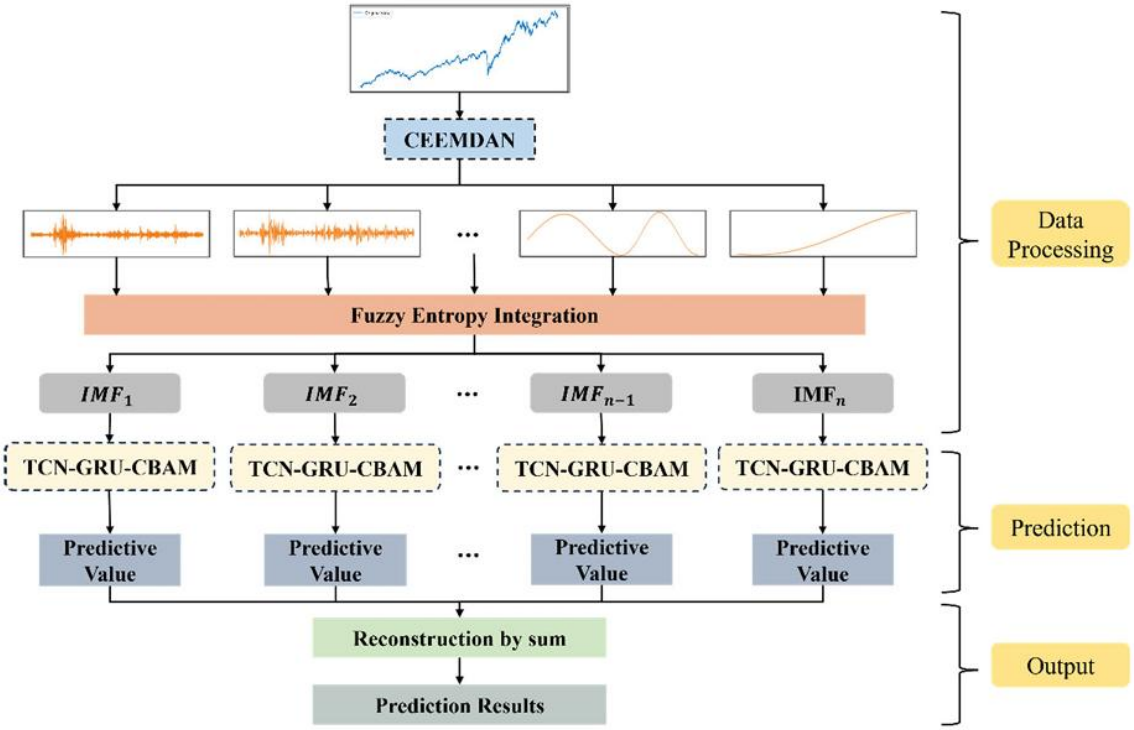

Figure 6. Construction framework of the CEEMDAN-TCN-GRU-CBAM model.

Attention modules also became increasingly modular in this hybrid literature. CBAM [31], though originally proposed in computer vision, offers a lightweight way to recalibrate channel-wise and spatial feature emphasis, and its adoption in financial hybrids signals a methodological pattern that extends beyond one paper: researchers are willing to import compact attention mechanisms when they can improve feature prioritization without the full computational burden of transformer-style global attention. In finance, such modules are especially attractive when multivariate inputs are noisy and only a subset of channels is informative under a given regime.

The specific steps of CEEMDAN decomposition are shown in Figure 7, where white Gaussian noise is added and iteratively averaged across ensemble trials to extract intrinsic mode functions from the original signal.

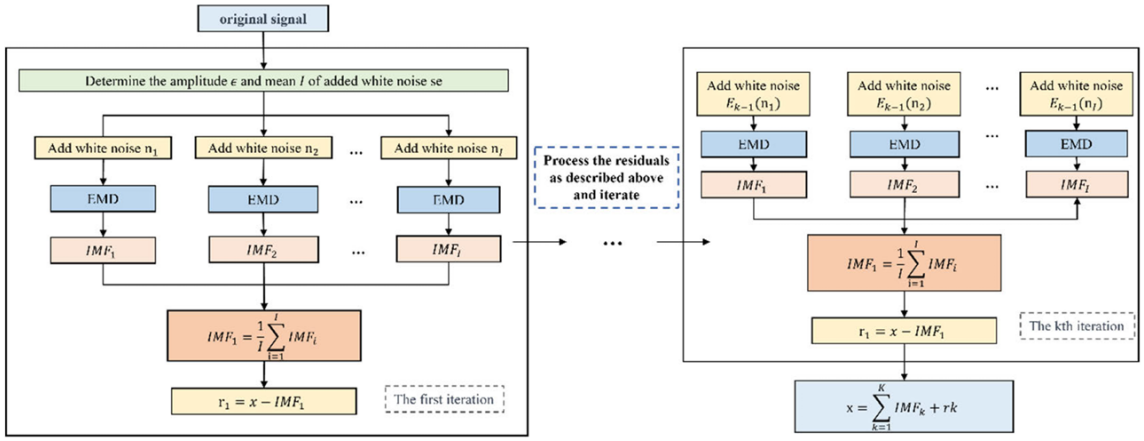

Figure 7. Specific steps of CEEMDAN decomposition.

Selective state space modeling introduces a different response to the limitations of both recurrence and full attention. Mamba [12] proposed linear-time sequence modeling through selective state spaces, offering a mechanism for content-aware state updates without quadratic attention cost. Its significance for stock forecasting lies less in immediate benchmark dominance than in architectural principle: if long-horizon dependence can be modeled through efficient selective transitions, then financial forecasting may scale to longer contexts and richer multivariate streams without the memory burden of standard transformers. Shi's MambaStock [27] is therefore noteworthy as an early domain-specific adaptation, showing how general state space advances can be specialized for stock prediction tasks.

Across these hybrid and state-space studies, the main trend is clear. Financial forecasting research is moving away from single-mechanism architectures toward modular systems that separate representation by frequency, modality, or state-update logic. The stronger these systems become, however, the more important benchmark discipline becomes as well. A complicated hybrid can outperform a plain LSTM or ARIMA in a fixed experiment and still fail to generalize under a different horizon, market, or crisis regime. For that reason, architecture design should be judged jointly with evaluation design, an issue returned to in later sections.

The decomposition literature also reminds researchers that reported gains should be interpreted in relation to task difficulty and data geography. For example, the CEEMDAN-LSTM-BPNN study of Mutinda and Geletu [24] focuses on the DAX index, while other hybrid studies consider different markets and targets [11,19]. An architecture that benefits a European

benchmark under one sampling frequency may not translate directly to U.S. or Chinese indexes under another. Consequently, the most defensible conclusion is not that one decomposition hybrid has solved stock forecasting, but that decomposition is a useful strategy for handling nonstationary mixtures when the evaluation pipeline is transparent and the forecast horizon is clearly defined.

4. General Time-Series Modeling Advances Relevant to Financial Forecasting

4.1. Attention, Normalization, and Representation Beyond Plain Recurrence

The transformer era changed stock forecasting indirectly before it changed it directly. Vaswani et al. [29] showed that attention could replace recurrence in sequence representation, enabling parallel training and long-range dependency modeling through adaptive token-to-token interaction. Although vanilla transformers are not automatically ideal for financial data, the shift in design philosophy was profound: sequence modeling no longer had to be organized around recurrent hidden-state propagation. This encouraged a wave of forecasting architectures that treated time windows as token sets, patch collections, or channel-specific sequences rather than as strictly recurrent streams.

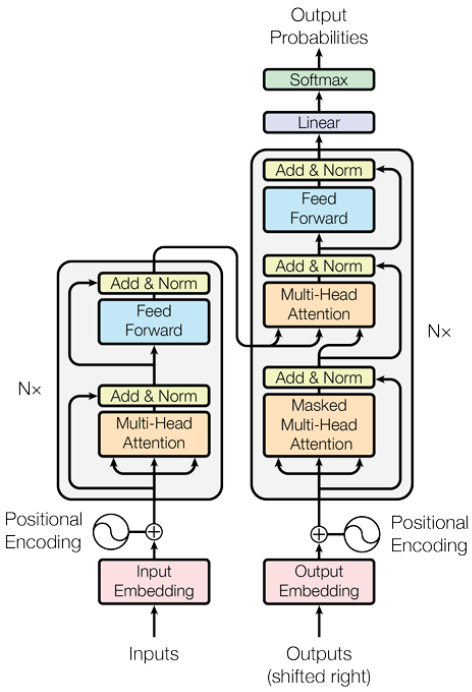

Figure 8. The Transformer model architecture.

Figure 8 shows the Transformer model architecture proposed by Vaswani et al. [29], where

multi-head self-attention replaces recurrence entirely, allowing every position in the sequence to attend to every other position in parallel.

At the same time, two practical concerns became impossible to ignore in time-series forecasting: selective feature emphasis and distribution shift. CBAM [31] offered a compact attention mechanism that many applied models could embed inside larger stacks, while RevIN [18] directly targeted distribution shift by normalizing each instance and reversing the normalization after prediction. RevIN is especially relevant to stock indexes because financial series frequently undergo level, variance, and seasonal changes across regimes. A model that performs well under one volatility regime may degrade sharply when the marginal distribution shifts, so normalization should be treated as a substantive design choice rather than a preprocessing afterthought.

4.2. Modern Forecasting Backbones: TimesNet, PatchTST, TiDE, TSMixer, iTransformer, TimeMixer, and SAMformer

A major branch of recent research focuses on designing backbones that encode temporal variation more efficiently and with stronger structural bias than generic sequence models. TimesNet [32] treats temporal patterns through two-dimensional variation modeling, reflecting the idea that multi-periodicity and local pattern repetition can be captured more naturally when time is reorganized into richer structural views. For stock indexes, where daily, weekly, and event-driven periodicities interact, this kind of representation is attractive because it avoids assuming that all useful information lies in a single one-dimensional dependency chain.

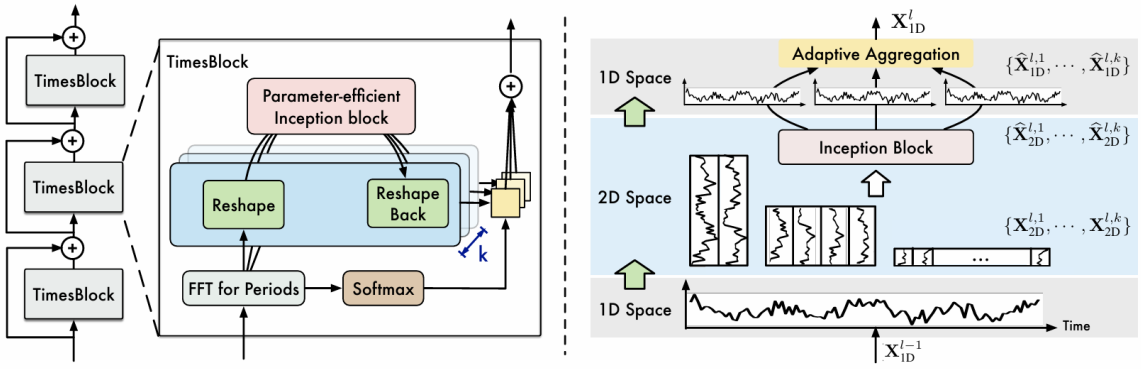

Figure 9. Overall architecture of TimesNet.

As depicted in Figure 9, TimesNet transforms one-dimensional time-series data into two-dimensional tensors by exploiting the detected period length, then applies 2D convolutional kernels to capture both intra-period and inter-period temporal patterns.

PatchTST, introduced in the work titled A Time Series Is Worth 64 Words [25], applies a patching strategy that converts contiguous time segments into token-like units. This approach is important for finance because local motifs such as breakout formation, volatility expansion, or post-news consolidation often occupy short contiguous windows rather than isolated timestamps. By working with patches and channel-wise treatment, the model reduces sequence length while preserving local temporal context. The underlying insight is that for many forecasting tasks, including stock indexes, the right unit of representation may be a short segment rather than a single scalar observation.

Other modern architectures challenge transformer dominance from different directions. TiDE [7] uses a dense encoder-decoder structure for long-term forecasting, while TSMixer [5] demonstrates that all-MLP architectures can be surprisingly competitive when temporal and feature mixing are well designed. These models matter because they show that good forecasting does not necessarily require attention at every layer. For stock prediction, where model efficiency and reproducibility are often as important as raw accuracy, simpler architectures can be attractive if they preserve enough capacity to model cross-horizon interactions and regime-sensitive features.

The transformer family itself has also diversified. iTransformer [20] rethinks the assignment of tokens and channels, arguing that inverted treatment of feature dimensions can improve forecasting effectiveness. TimeMixer [30] emphasizes decomposable multiscale mixing, explicitly acknowledging that temporal dynamics unfold across nested scales rather than a single homogeneous horizon. SAMformer [16] adds sharpness-aware optimization and channel-wise attention, highlighting that successful forecasting depends not only on representation structure but also on how the model is trained and regularized. For stock indexes, these contributions collectively strengthen the case that architecture search should be guided by scale decomposition, channel relevance, and generalization stability rather than by adopting the latest generic transformer variant uncritically.

What distinguishes these modern backbones is not simply accuracy, but the way each one defines the fundamental forecasting object. TimesNet [32] highlights temporal variation patterns, PatchTST [25] highlights local segments, TiDE [7] highlights dense temporal compression, TSMixer [5] highlights separable mixing, iTransformer [20] highlights variable-centric representation, and TimeMixer [30] highlights explicit multi-scale decomposition. This variety is useful for finance because stock indexes are simultaneously path-dependent, feature-dependent, and scale-dependent. A researcher choosing among these architectures should

therefore ask which representation lens best matches the available data and the intended decision horizon, rather than treating all transformer-era models as interchangeable upgrades.

4.3. Foundation Models and Large-Model Thinking for Time Series

The latest stage of the literature extends beyond task-specific architectures, venturing into the realm of foundation-model style forecasting. A notable example is Time-LLM [17], which reprograms large language models specifically for time-series forecasting. This innovative approach effectively poses the question of whether pretrained language representations can be adapted for temporal reasoning through appropriate interfaces and techniques. This is a conceptually bold move as it treats time series not merely as numeric sequences to be extrapolated, but rather as structured signals that may significantly benefit from the application of broad pretrained priors derived from extensive datasets. The appeal for stock forecasting is particularly strong: if large pretrained systems can successfully transfer robust sequence abstractions and insights, they may help alleviate the severe data scarcity challenges that are often encountered by domain-specific models trained solely on a single market or specific forecasting horizon. By leveraging the strengths of foundational models, researchers could unlock new avenues for enhancing predictive performance and improving decision-making in financial markets.

Related efforts deepen this emerging trend in various innovative ways. For instance, TEMPO [4] formulates a prompt-based generative pretraining approach specifically tailored for time-series forecasting, while Timer [22] advocates for the use of generative pre-trained transformers as large-scale models designed for time-series data. Additionally, the decoder-only foundation model introduced by Das et al. [7] advances the intriguing idea that large autoregressive forecasting systems can be constructed directly for time-series domains without significant modifications. Furthermore, Lag-Llama [26] introduces a probabilistic forecasting perspective, which is particularly valuable in finance because the calibration of uncertainty is just as crucial as producing mean predictions. Collectively, these studies represent a significant shift in the overarching research question from simply asking, "Which architecture performs best on a given benchmark?" to a more nuanced inquiry: "What kind of pretraining and transfer learning regime can effectively produce adaptable forecasting behavior across diverse tasks and datasets?" This evolution not only reflects a growing sophistication in modeling approaches but also emphasizes the importance of flexibility and generalization in predictive analytics within financial contexts.

Figure 10 presents the model framework of TIME-LLM, which reprograms a frozen large

language model for time-series forecasting by converting temporal data into token embeddings through a prompt-based interface.

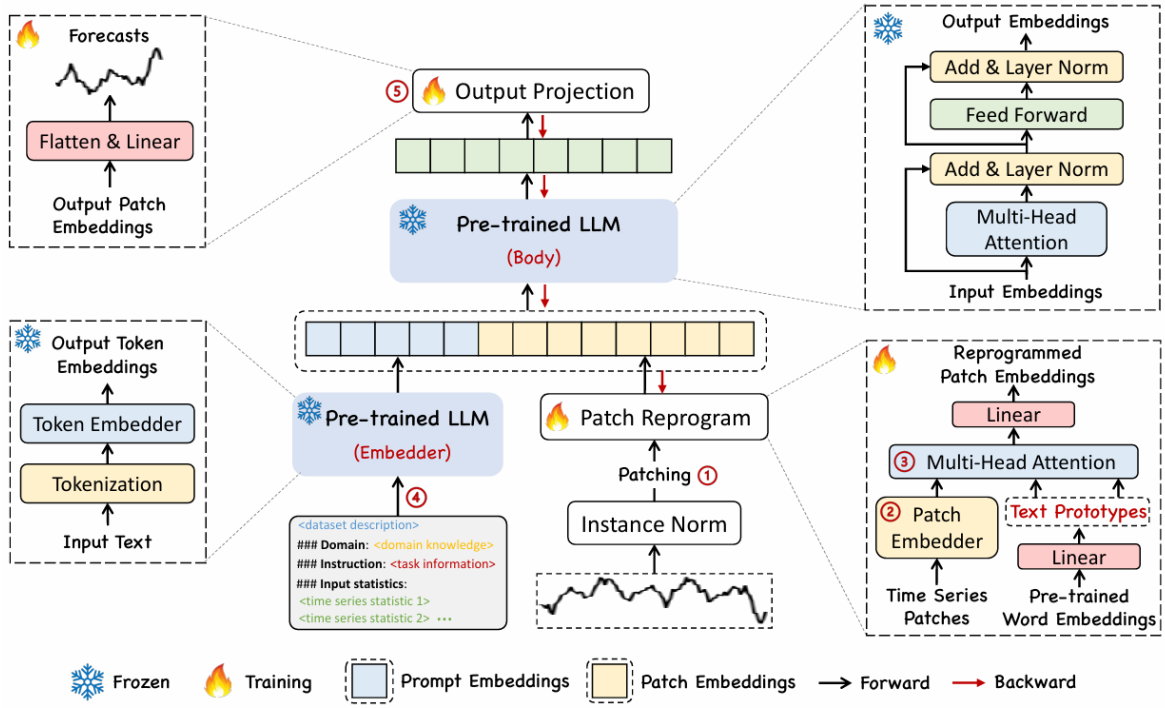

Figure 10. The model framework of TIME-LLM.

Yet stock index forecasting also exposes the limits of foundation-model optimism. Financial data are sparse relative to web-scale corpora, strongly regime dependent, and heavily influenced by institutional changes that may not repeat in the training history. A large model pretrained on generic time series or even on language may capture broad sequential regularities while still missing the economic semantics of market microstructure, policy surprise, or risk transmission. Therefore, the main value of foundation models in finance may lie in transfer-efficient representation learning, probabilistic calibration, and multimodal integration rather than zero-shot market prediction. The most promising direction is likely not replacement of domain models, but controlled fusion between pretrained sequence priors and finance-specific supervision.

Probabilistic output is especially important in this context. Large forecasting systems that report only point predictions are often less useful to financial decision makers than models that provide calibrated uncertainty, because trading, hedging, and risk budgeting all depend on the distribution around the forecast. Lag-Llama [26] is therefore conceptually important even beyond its empirical results, since it represents the move from deterministic large-model extrapolation toward probabilistic large-model forecasting. For stock index applications, the next generation of foundation models should ideally support scenario ranges, tail-aware

intervals, and confidence measures that remain stable under regime change.

Another unresolved question is how prompting and adaptation should work in financial settings. In language tasks, prompting can externalize task instructions, contextual hints, and examples. For stock forecasting, however, the useful context may consist of recent market regimes, macro calendars, event descriptions, or retrieved analog periods rather than natural-language task statements alone. This suggests that prompt-based time-series models such as TEMPO [4] and reprogramming approaches such as Time-LLM [17] may achieve their greatest financial value when combined with retrieval, regime annotation, and economically meaningful metadata rather than raw numeric windows in isolation.

5. Comparative Discussion

5.1. Inductive Bias and Data Regime

The surveyed literature clearly illustrates that model performance is fundamentally influenced by inductive bias. Statistical models tend to excel when the target process exhibits short-memory characteristics, is interpretable, and remains sufficiently stable over time for parameter estimates to retain their meaning across different periods. Classical machine learning approaches perform particularly well when domain expertise can contribute informative features that enhance predictive accuracy. Recurrent networks prove useful in scenarios where medium-range path dependence plays a significant role, while convolutional, patch-based, and multiscale architectures are better suited for capturing repeated local motifs and facilitating cross-horizon composition of data patterns. Additionally, hybrid decomposition models become increasingly attractive when the same raw time series contains multiple incompatible temporal regimes, allowing for more nuanced analysis. Meanwhile, foundation-model approaches gain plausibility when the ability to transfer knowledge across tasks can effectively compensate for limited labeled data in certain contexts. Thus, there is no architecture-independent answer to the question of what works best for stock index forecasting; instead, the choice of model should be informed by the specific characteristics of the data and the underlying processes being modeled. This highlights the importance of tailoring approaches to the unique challenges presented by financial time series data.

Dataset scale and market regime are equally important. Low-frequency index series with a few thousand observations do not justify the same parameterization as large multivariate panels or dense intraday sequences. Likewise, models tuned during tranquil periods can degrade badly during crises because the mapping from predictors to returns changes faster than the model can

adapt. This is why seemingly older ideas such as GARCH-style variance awareness [2] and RevIN-style shift handling [18] continue to matter even in otherwise modern pipelines. For practitioners, model selection should begin with the structure of the available data and the forecast use case, not with the popularity of the architecture.

5.2. Exogenous Information, Multimodality, and Representation Scope

Another important pattern is the widening scope of input representation. Price-only models remain useful benchmarks, but the stronger finance studies increasingly combine multiple information channels such as technical indicators, macro variables, cross-market covariates, and textual sentiment [15,21]. This shift reflects an economic reality: many index movements are driven by information that is not recoverable from recent prices alone. At the same time, adding modalities increases alignment risk, because timestamps, release lags, missingness, and varying update frequencies can introduce subtle leakage or dilution of signal. A robust review therefore cannot treat multimodality as automatically beneficial; its value depends on disciplined temporal synchronization and clear causal availability at forecast time.

Modern backbones also differ in how broadly they define representation scope. Patch-based and mixer-style models [5,25] widen the local receptive unit; multiscale models [30,32] widen temporal granularity; foundation models [4,8,17,22,26] widen transfer scope across tasks and datasets. These are all forms of representation expansion, but they respond to different bottlenecks. For stock indexes, the key challenge is to expand scope without drowning the weak predictive signal in irrelevant context. This is where financially grounded regularization, sparsity, and relevance weighting may become more valuable than simply increasing model size.

5.3. Evaluation, Robustness, and Economic Relevance

A persistent weakness of the stock forecasting literature is evaluation fragmentation. Studies differ in market selection, forecast horizon, target variable, split protocol, preprocessing pipeline, and reported metrics, making direct comparison difficult. Error metrics such as RMSE, MAE, or MAPE remain necessary, but they do not reveal whether a forecast is economically actionable after transaction costs, turnover, slippage, and risk constraints. This is especially relevant when comparing sophisticated hybrids or large models against simpler baselines: a marginal gain in point accuracy may not translate into a better trading or risk-management decision. Future stock index forecasting research should therefore report both statistical accuracy and decision-oriented utility whenever the intended application is financial action

rather than descriptive extrapolation.

Robustness is a second evaluation dimension that deserves more weight. A model should be tested across rolling windows, multiple market states, and realistic out-of-sample regimes rather than a single favorable split. Distribution-shift handling [18], decomposition robustness [19,24], and transfer behavior of foundation models [4,8,17,22,26] all need evaluation protocols that stress adaptation rather than just average fit. Without such testing, architectural novelty can easily be mistaken for durable forecasting ability. The literature surveyed here suggests that reproducibility, robustness, and economic interpretation are now at least as important as adding another modeling block to the pipeline.

Metric selection should also be broadened. Directional accuracy, hit rate, interval coverage, calibration error, and portfolio-level statistics such as drawdown or risk-adjusted return can reveal properties that RMSE alone conceals. A model with slightly worse point error may still be preferable if it identifies turning points more consistently or produces uncertainty estimates that prevent overconfident trading. Conversely, a model with excellent average error may fail exactly when risk management matters most, such as during crisis transitions. The next phase of stock index forecasting research should therefore evaluate models as forecasting systems embedded in decisions, not merely as function approximators optimized for one scalar score.

5.4. Practical Guidance for Model Selection

The reviewed evidence supports a pragmatic selection logic. When interpretability, data scarcity, or regulatory transparency is paramount, ARIMA, GARCH, VAR, and classical machine learning remain strong starting points [2,3,10,28,33]. When the dataset is rich enough and the objective is to exploit nonlinear sequential dependence, recurrent and convolutional financial models [1,6,9,11,12,13,14,15,22,23,27,29,34] are appropriate, especially if exogenous variables are available. When nonstationarity is severe and multi-scale behavior dominates, decomposition-driven hybrids and state-space methods [12,19,24,27] are promising. When the task requires transfer across datasets, probabilistic forecasting, or multimodal integration at scale, the more recent time-series and foundation-model literature [4,5,7,8,16,17,20,25,26,30,32] becomes relevant. The strongest research strategy is not to assume that one category supersedes all others, but to treat each family as a tool matched to a specific data regime and decision requirement.

6. Open Challenges and Research Directions

The first open challenge is benchmark quality. Stock forecasting studies still suffer from

inconsistent train-test splits, insufficient leakage control, and limited reporting of preprocessing choices. Standardized rolling-window benchmarks across multiple geographic markets and volatility regimes would make it much easier to judge whether a new model improves generalization or simply benefits from a favorable setup. Such benchmarks should include both statistical targets and decision-oriented targets, because a useful stock index model may be valuable for direction classification, volatility-aware allocation, or interval forecasting even when point error is not state of the art.

Benchmark design should also be horizon aware. A model that is competitive for one-step-ahead daily prediction may fail for weekly trend forecasting or monthly allocation support, and the reverse can also be true. Many architecture debates in the literature are implicitly debates about horizon mismatch rather than absolute model quality. Future evaluations should therefore report results across short, medium, and longer horizons whenever possible, making it clear whether a method is best interpreted as a micro-pattern detector, a regime tracker, or a long-horizon planner.

The second challenge is multimodal and multi-resolution integration. Financial signals arrive through prices, order flow, macro releases, news, and cross-asset spillovers, each with different timing conventions and noise characteristics. Future research should pay more attention to causally valid synchronization, confidence-aware fusion, and hierarchical routing across temporal scales. The success of sentiment-aware models [21], decomposition hybrids [19,24], and state-space or multiscale architectures [12,27,30,32] suggests that the next gains will come from better coordination among representations rather than from a single universal backbone.

The third challenge concerns foundation models and transfer learning. Large-model approaches [4,8,17,22,26] are promising, but finance raises unresolved questions about pretraining corpora, domain mismatch, continual adaptation, and probabilistic calibration under rare events. A stock index model that sees many generic time series may still fail on policy shocks or crisis dynamics that are economically unique. Accordingly, foundation-model research in finance should emphasize domain adaptation, uncertainty estimation, and retrieval or prompting mechanisms grounded in economically meaningful context rather than assuming that scale alone solves the forecasting problem.

Finally, interpretability and responsible deployment remain central. As models become deeper and more hybridized, it becomes harder to explain whether predictions arise from stable economic relationships, transient co-movements, or artifacts of preprocessing. This matters not only for trust, but also for model maintenance: without interpretable failure modes, it is difficult

to know when a forecasting system should be retrained, down-weighted, or suspended. The most valuable future systems will therefore combine strong predictive machinery with diagnostics that help analysts understand regime sensitivity, feature relevance, and uncertainty boundaries.

A related research direction is human-in-the-loop forecasting. In many practical settings, stock index models are not used autonomously but as inputs to analyst judgment, asset allocation meetings, or risk committees. This creates design opportunities that are underexplored in the literature, such as models that surface the dominant temporal regime, compare current conditions with retrieved historical analogues, or explain whether a forecast is driven by price action, exogenous news, or cross-market contagion. As forecasting architectures become more powerful, making them easier to interrogate may generate more practical value than pursuing another marginal accuracy gain on a fixed benchmark.

7. Conclusion

The literature reviewed here shows that stock index forecasting has progressed from linear univariate and multivariate benchmarks to a broad ecosystem of deep neural, hybrid, state-space, and foundation-model approaches. Each stage addressed a real limitation of the previous one: ARIMA clarified baseline dependence structure, GARCH modeled changing volatility, classical machine learning captured nonlinear feature interactions, recurrent and convolutional networks automated sequence representation, hybrid systems handled multi-scale nonstationarity, and foundation-model approaches opened the possibility of broader transfer and probabilistic generalization. At the same time, the field has learned that architectural novelty is not a substitute for disciplined evaluation.

For stock index forecasting, the central research problem is no longer simply how to fit a more expressive sequence model. It is how to align model inductive bias with financial data structure, preserve robustness under distribution shift, integrate exogenous information without leakage, and evaluate usefulness in economically meaningful terms. Viewed in that light, the most promising future direction is a hybrid one: transparent statistical reasoning for diagnostics, domain-aware deep architectures for representation, and carefully adapted foundation-model components for transfer, uncertainty, and multimodal context. Such an agenda is more demanding than chasing benchmark gains, but it is also more likely to produce forecasting systems that remain useful outside the narrow conditions of a single experiment.

References

[1] Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv*:1803.01271.

[2] Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3), 307–327. https://doi.org/10.1016/0304-4076(86)90063-1

[3] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. *https://doi.org/10.1023/A:1010933404324*

[4] Cao, D., Jia, F., Arik, S. O., Pfister, T., Zheng, Y., Ye, W., & Liu, Y. (2024). TEMPO: Prompt-based generative pre-trained transformer for time series forecasting. *In Proceedings of the International Conference on Learning Representations* (ICLR).

[5] Chen, S.-A., Li, C.-L., Arik, S. O., Yoder, N. C., & Pfister, T. (2023). TSMixer: An all-MLP architecture for time series forecasting. *Transactions on Machine Learning Research*.

[6] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (pp. 1724–1734). https://doi.org/10.3115/v1/D14-1179

[7] Das, A., Kong, W., Leach, A., Mathur, S., Sen, R., & Yu, R. (2023). Long-term forecasting with TiDE: Time-series dense encoder. *arXiv preprint arXiv*:2304.08424.

[8] Das, A., Kong, W., Sen, R., & Zhou, Y. (2024). A decoder-only foundation model for time-series forecasting. *In Proceedings of the 41st International Conference on Machine Learning (Vol. 235, pp. 10148–10167)*. PMLR.

[9] Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2), 654–669. https://doi.org/10.1016/j.ejor.2017.11.054

[10] Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1), 119–139. https://doi.org/10.1006/jcss.1997.1504

[11] Ge, Q. (2025). Enhancing stock market forecasting: A hybrid model for accurate prediction of S&P 500 and CSI 300 future prices. *Expert Systems with Applications*, 260, Article 125380. https://doi.org/10.1016/j.eswa.2024.125380

[12] Gu, A., & Dao, T. (2023). Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv*:2312.00752.

[13] He, Z., Zhang, H., & Por, L. Y. (2024). A comparative study on deep learning models for stock price prediction. *In Image Processing, Electronics and Computers: Advances in Transdisciplinary Engineering*. IOS Press. https://doi.org/10.3233/ATDE240502

[14] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780. *https://doi.org/10.1162/NECO.1997.9.8.1735*

[15] Hoseinzadeh, E., & Haratizadeh, S. (2019). CNNpred: CNN-based stock market prediction using a diverse set of variables. *Expert Systems with Applications*, 129, 273–285. https://doi.org/10.1016/j.eswa.2019.03.029

[16] Ilbert, R., Odonnat, A., Feofanov, V., Virmaux, A., Paolo, G., Palpanas, T., & Redko, I. (2024). SAMformer: Unlocking the potential of transformers in time series forecasting with sharpness-aware minimization and channel-wise attention. *In Proceedings of the 41st International Conference on Machine Learning* (Vol. 235, pp. 20924–20954). PMLR.

[17] Jin, M., Wang, S., Ma, L., Chu, Z., Zhang, J. Y., Shi, X., Chen, P.-Y., Liang, Y., Li, Y.-F., Pan, S., & Wen, Q. (2024). Time-LLM: Time series forecasting by reprogramming large language models. *In Proceedings of the International Conference on Learning*

Representations (ICLR).

[18] Kim, T., Kim, J., Tae, Y., Park, C., Choi, J.-H., & Choo, J. (2022). Reversible instance normalization for accurate time-series forecasting against distribution shift. *In Proceedings of the Tenth International Conference on Learning Representations (ICLR)*.

[19] Li, S., Tang, G., Chen, X., et al. (2024). Stock index forecasting using a novel integrated model based on CEEMDAN and TCN-GRU-CBAM. *IEEE Access*, 12, 122524–122543. https://doi.org/10.1109/ACCESS.2024.3452426

[20] Liu, W. J., Ge, Y. B., & Gu, Y. C. (2024). News-driven stock market index prediction based on trellis network and sentiment attention mechanism. *Expert Systems with Applications*, 250, Article 123966. https://doi.org/10.1016/j.eswa.2024.123966

[21] Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., & Long, M. (2024). iTransformer: Inverted transformers are effective for time series forecasting. *In Proceedings of the International Conference on Learning Representations (ICLR)*.

[22] Liu, Y., Zhang, H., Li, C., Huang, X., Wang, J., & Long, M. (2024). Timer: Generative pre-trained transformers are large time series models. *In Proceedings of the 41st International Conference on Machine Learning (Vol. 235, pp. 32369–32399)*. PMLR.

[23] Mu, S., Liu, B., Gu, J., et al. (2024). Research on stock index prediction based on the spatiotemporal attention BiLSTM model. *Mathematics*, 12(18), Article 2812. https://doi.org/10.3390/math12182812

[24] Mutinda, J. K., & Geletu, A. (2025). Stock market index prediction using CEEMDAN-LSTM-BPNN-decomposition ensemble model. *Journal of Applied Mathematics*, 2025, Article 7706431. https://doi.org/10.1155/jama/7706431

[25] Nie, Y., Nguyen, N. H., Sinthong, P., & Kalagnanam, J. (2023). A time series is worth 64 words: Long-term forecasting with transformers. *In Proceedings of the International Conference on Learning Representations (ICLR)*.

[26] Rasul, K., Ashok, A., Williams, A. R., Ghonia, H., Bhagwatkar, R., Khorasani, A., Darvishi Bayazi, M. J., Adamopoulos, G., Riachi, R., Hassen, N., Bilos, M., Garg, S., Schneider, A., Chapados, N., Drouin, A., Zantedeschi, V., Nevmyvaka, Y., & Rish, I. (2023). Lag-Llama: Towards foundation models for probabilistic time series forecasting. *arXiv preprint arXiv*:2310.08278.

[27] Shi, Z. (2024). MambaStock: Selective state space model for stock prediction. *arXiv preprint arXiv*:2402.18959.

[28] Sims, C. A. (1980). Macroeconomics and reality. *Econometrica,* 48(1), 1–48. https://doi.org/10.2307/1912017

[29] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *In Advances in Neural Information Processing Systems 30* (pp. 5998–6008).

[30] Wang, S., Wu, H., Shi, X., Hu, T., Luo, H., Ma, L., Zhang, J. Y., & Zhou, J. (2024). TimeMixer: Decomposable multiscale mixing for time series forecasting. *In Proceedings of the International Conference on Learning Representations (ICLR)*.

[31] Woo, S., Park, J., Lee, J. Y., & Kweon, I. S. (2018). CBAM: Convolutional block attention module. *In Proceedings of the European Conference on Computer Vision* (pp. 3–19). https://doi.org/10.1007/978-3-030-01234-2_1

[32] Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., & Long, M. (2023). TimesNet: Temporal 2D-variation modeling for general time series analysis. *In Proceedings of the International Conference on Learning Representations (ICLR)*.

[33] Xu, R. (2025). Modeling and comparing S&P 500, FTSE and SSEC stock price with ARIMA model. Frontiers in Economics and Management, 6(6), 60–69. *https://doi.org/10.6981/FEM.202506_6(6).0008*

[34] Zhang, J., Ye, L., & Lai, Y. (2023). Stock price prediction using CNN-BiLSTM-attention

model. *Mathematics*, 11(9), Article 1985. https://doi.org/10.3390/math11091985

Online ISSN: 3104-9338
Print ISSN: 3104-932X

Constructing a Generative AI Assistant for the Reform of University Experimental Teaching: A Case Study of the Advanced Language Programming (C Language) Course

Xinyu Song*

* School of Electronic Information Engineering, Geely University of China, ChengDu, China, 641423

Received: April 13, 2026

Revised: April 19, 2026

Accepted: April 22, 2026

Published online: April 27, 2026

To appear in: *International Journal of Advanced AI Applications,* Vol. 2, No. 5 (May 2026)

* Corresponding Author: Author Name (songxinyu@guc.edu.cn)

Abstract. With the rapid advancement of information technology, Advanced Language Programming (C Language) has become a core curriculum in computer-related disciplines in higher education institutions. However, the course's intrinsic characteristics—such as its strong practicality, rapid technological updates, and significant variance in student foundational knowledge—pose considerable challenges to traditional experimental teaching. To address these challenges and enhance instructional quality and personalized learning experience, this paper proposes and implements a Generative AI Teaching Assistant System (GenAI-TA) specifically designed for the "Advanced Language Programming (C Language)" experimental course. The system is built upon an advanced Large Language Model (LLM) and integrates Retrieval-Augmented Generation (RAG) technology. It is fine-tuned by incorporating a course-specific knowledge base (including the syllabus, lab manuals, code examples, and a collection of common errors) to provide precise, real-time, and personalized tutoring. This paper elaborates on the GenAI-TA's system architecture, key technical implementation details, and its integration scheme with the Geely University OJ Platform development environment. To evaluate its pedagogical effectiveness, a one-semester quasi-experimental study was conducted. The results indicate that, compared to the control group under the traditional teaching model, students in the experimental group using GenAI-TA achieved significant improvements in programming skills, project completion quality, and problem-solving abilities ($p < 0.05$). Furthermore, the System Usability Scale (SUS) score of 85.7 suggests a high level of student acceptance and satisfaction with the system. This research validates the immense potential of Generative AI teaching assistants in reforming the experimental instruction of practical courses and provides empirical evidence and a feasible technical pathway for the deeper application of AI in the higher education sector.

Keywords: *Generative Artificial Intelligence (GenAI), Advanced Language Programming, Large Language Model (LLM)*

1. Introduction

The course “Advanced Language Programming (C Language)” is an indispensable and crucial practical course in contemporary higher education programs such as Computer Science and Software Engineering. This course aims to equip students with the fundamental principles, methodologies, and core technologies of C language programming, enabling them to design and implement practical C language programs and lay a solid foundation for subsequent programming-related courses (such as Android or iOS development). The pedagogical core of the course lies in extensive programming experiments and project practice, which are vital for students to consolidate theoretical knowledge and cultivate engineering capabilities [1].

However, in the traditional experimental teaching model, the "Advanced Language Programming (C Language)" course faces numerous challenges. Firstly, the rapid iteration of programming technologies necessitates frequent updates to course content and lab projects, imposing extremely high demands on instructors' knowledge reserves and their capacity for developing teaching resources. Secondly, students exhibit significant differences in their programming foundation, logical thinking, and learning pace, making a uniform teaching schedule difficult to meet the personalized learning needs of all students. During the experimental process, students inevitably encounter various unexpected programming errors and environment configuration issues. Yet, the limited time and energy of instructors or traditional TAs make it challenging to provide 24/7, one-on-one, instantaneous feedback and guidance [2]. This lag in guidance often interrupts the student's learning flow (or "flow state"), dampens their enthusiasm, and can even lead some students to abandon complex lab assignments.

In recent years, Generative AI (GenAI) technology, represented by Large Language Models (LLMs), has achieved breakthrough progress and demonstrated immense potential for application in the education sector. GenAI's ability to comprehend natural language, generate code, explain complex concepts, and offer personalized, conversational interaction makes it an ideal tool for assisting experimental teaching and addressing the aforementioned challenges. By developing an AI teaching assistant specifically tailored to a particular course, it can provide students with a 24/7 available "expert partner" that instantly answers questions, debugs code, and offers learning suggestions, thereby achieving large-scale personalized education.

Although existing research has explored the application of AI in programming education and as a general instructional tutoring tool, studies focusing specifically on the systematic design, implementation, and empirical evaluation of an AI solution for a highly practical university lab

course like “Advanced Language Programming (C Language)” remain scarce. Specifically, there is a lack of cases that are deployed in a real classroom environment and measure its impact on student learning outcomes, engagement, and satisfaction [3-5].

In view of this, this paper takes the “Advanced Language Programming (C Language)” course as a case study, aiming to design, develop, and evaluate a Generative AI Teaching Assistant system (GenAI-TA) for the reform of experimental teaching in higher education. The main contributions of this research include:

(1) Proposing a Generative AI Teaching Assistant system architecture that integrates a course-specific private knowledge base, utilizing Retrieval-Augmented Generation (RAG) technology to ensure the accuracy and relevance of responses.

(2) Implementing the TA system and seamlessly integrating it as a plugin into the mainstream Geely University OJ Platform Integrated Development Environment, providing immersive learning support to students.

(3) Conducting a semester-long quasi-experimental study to quantitatively and qualitatively evaluate the pedagogical effectiveness of GenAI-TA from multiple dimensions, including learning gain, system usability, and student satisfaction.

The structure of this paper is organized as follows: Section 2 reviews related work; Section 3 details the system architecture and implementation of GenAI-TA; Section 4 describes the experimental design, data collection methods, and results; Section 5 provides an in-depth discussion and analysis of the experimental results; finally, Section 6 concludes the paper and outlines future work.

2. Related Work

This study is built upon two main foundational areas: the instructional reform of Advanced Language Programming (C Language) courses, and the application of Generative AI in the education sector.

2.1. Instructional Status Quo and Reform Exploration of Advanced Language Programming (C Language) Courses

The syllabus for the "Advanced Language Programming (C Language)" course typically covers several modules, including programming environment fundamentals, data structure application, file operation, pointer usage, function design, and basic algorithm implementation. In terms of pedagogical methods, there is a general emphasis on "learning by doing," utilizing

approaches such as case studies, Project-Based Learning (PBL), and the Flipped Classroom model to strengthen students' hands-on practical abilities. The experimental component is the core focus of the course, with lab types gradually progressing from basic verification and design-oriented tasks to comprehensive and innovative projects. For instance, students are required to complete a series of tasks ranging from simple ones, like "Basic Calculator Program Development", to more complex ones, such as "Student Information Management System Based on File Storage", and then "Simple Data Sorting and Searching Algorithm Implementation."[6]

To address the shortcomings of traditional instruction, numerous reform efforts have been explored. For example, some studies propose constructing a multi-level, modular laboratory system to meet the needs of students at different proficiency levels. Other research has explored cloud-based online lab environments to solve the difficulties students face with local environment setup. However, these reforms primarily focus on course content organization and pedagogical models. They still lack effective technical support tools for the instantaneous and personalized problems students encounter during the experimental process. Students' learning experience largely remains dependent on the on-site guidance of the instructor, which is often inadequate in the context of large-scale teaching.[7]

2.2. Application of Generative AI in the Education Sector

The emergence of Generative AI has brought about a new paradigm for educational innovation. In recent years (2022–2025), a large volume of research has emerged, exploring the role of AI as a learning partner, tutor, or teaching assistant. These studies generally agree that Generative AI can provide personalized feedback, stimulate students' learning interest, and enhance learning efficiency [8,9].

In terms of evaluation, researchers employ various metrics to gauge the effectiveness of AI TAs. For example, a 2025 study used an experimental design to assess the impact of Generative AI on student learning satisfaction, self-efficacy, and learning outcomes, finding that personalized AI support significantly enhanced these indicators. Other studies have also focused on student engagement, the usability of the AI tutor, and student acceptance. Regarding evaluation tools, the System Usability Scale (SUS) is widely used to measure users' subjective perception of an AI system's usability [10].

In the field of Computer Science education, AI code generators (such as GitHub Copilot) and LLM-based chatbots (such as ChatGPT) have been utilized to assist in programming learning. Research indicates that these tools can help students write code and debug errors more quickly.

However, concerns also exist, such as the potential for students to develop over-reliance, thereby weakening their ability to solve problems independently, and the possibility of the model generating insecure or inefficient code.

2.3. Research Gap

In summary, despite the increasing number of studies on the application of Generative AI in education, several research gaps persist: most existing research focuses on general educational scenarios or pure theoretical programming instruction, with highly limited research on AI Teaching Assistant systems specifically designed for highly practical lab courses like "Advanced Language Programming (C Language)"—which involves complex IDEs, data structure application, file operation, and pointer-based programming[11]; many AI tutoring tools exist as independent web pages or chatbot applications rather than being deeply integrated with the professional development tools (e.g., Geely University OJ Platform) that students use daily, disrupting the continuity of the learning process [12]; and although some preliminary user experience studies exist, long-term (e.g., a whole semester), controlled, multi-dimensional quantitative empirical evaluations—including metrics such as learning gain, project quality, and SUS scores—in real university lab courses are still rare.

To bridge these gaps, this study develops a GenAI-TA that is deeply integrated into Geely University OJ Platform and features a highly customized knowledge base, and conducts a comprehensive empirical evaluation of its effectiveness in a genuine instructional environment.

3. System Architecture and Implementation

To provide precise and efficient tutoring for the "Advanced Language Programming (C Language)" course, the GenAI-TA system we designed must possess the following core capabilities: (1) Deep comprehension of course-specific knowledge; (2) Accurate answering of factual questions and explanation of complex concepts; (3) Generation of high-quality example code conforming to course standards; (4) Understanding and assisting in the debugging of student errors; and (5) Providing a seamless interaction experience.

3.1. System Overview Architecture

Presentation Layer: This is the front-end interface for user interaction with the system. We designed it as a Geely University OJ Platform Plugin. The plugin provides a chat window in the IDE's sidebar, allowing students to ask questions using natural language directly within their C language programming environment without switching applications. The UI design is concise,

supports C language code block highlighting, one-click copy, and insertion functionality, ensuring a smooth user experience.

Application Layer: As the system's backend service, it is responsible for handling requests from the front-end, coordinating business logic, and calling the underlying models. It includes three core modules: Session Manager: Responsible for maintaining the conversation history of each user, providing context support for multi-turn dialogue. Intent Recognition Module: Conducts preliminary analysis of user input to determine the intent, categorizing it as "Conceptual Question" (e.g., pointer usage, file operation principles), "Code Request" (e.g., C language function implementation, loop structure writing), "Error Debugging" (e.g., compilation errors, logical errors in C code), or "Lab Guidance" (e.g., C language lab task completion tips).

The Query Processor constructs the appropriate Prompt based on the recognized intent and decides whether to directly call the LLM or first go through the RAG process. The Model & Data Layer serves as the intelligent core of the system, consisting of the Large Language Model (LLM) and the course-specific knowledge base; we selected an advanced open-source LLM (such as Llama 3 70B) as the foundation model, and choosing an open-source model facilitates localized deployment and deep fine-tuning, thereby better protecting student data privacy and controlling costs.

The Course Knowledge Base is crucial for ensuring that the AI Teaching Assistant's responses are professional and accurate, so we built a comprehensive, structured knowledge base for the "Advanced Language Programming (C Language)" course, with data sources including course instructional materials (course outline, instructor PPTs, text transcripts of official instructional videos focusing on C language core knowledge like data types, pointers, functions, and file operations), lab guidance documentation (objectives, steps, requirements, and grading criteria for all C language labs such as basic syntax practice, function development, and file operation experiments), code assets (all C language example code, project templates like student information management system and simple calculator program, and best practice samples provided by the instructor that adhere to C language coding standards), and FAQ & Error Collection (accumulated frequently asked questions from previous years such as pointer operation confusion and file opening/closing errors, typical C language compilation and logical errors and their corresponding solutions). All documents in the knowledge base are chunked, cleaned, and converted into high-dimensional vectors using a text embedding model (such as m3e-base), then stored in a vector database (such as ChromaDB), which enables quick retrieval

of C language-specific knowledge to assist the LLM in generating accurate responses.

4. Experimental Design and Results

To evaluate the effectiveness of the GenAI-TA system in a real instructional environment, we conducted a 16-week quasi-experimental study.

4.1. Participants and Grouping

The participants in this study were 98 undergraduate students enrolled in the “Advanced Language Programming (C Language)” course during the first semester of the 2024-2025 academic year at our university. These students were randomly assigned to two parallel experimental classes:

Experimental Group (N=49): Students were permitted to use the GenAI-TA system, integrated into Geely University OJ Platform, during lab sessions. They could also access the system for assistance outside of class hours.

Control Group (N=49): Students adopted the traditional learning approach. Guidance during lab sessions was provided by a human Teaching Assistant (a postgraduate student), and post-class questions were addressed via forums or email to the instructor or TA.

Both groups utilized the same syllabus, textbooks, lab assignments, and grading criteria. To ensure fairness, the query time for the human TA was restricted to the fixed lab hours. Before the experiment began, we conducted a t-test on the students' programming foundations (based on grades from the prerequisite course "Java Programming") and learning motivation questionnaire. The results showed no significant difference between the two groups ($p > 0.1$), indicating that the grouping was balanced.

4.2. Experimental Tasks and Data Collection

The course comprised 8 lab assignments, ranging from simple C language syntax practice to complex file operation and data structure application tasks, and the final major assignment required students to complete a comprehensive C language program independently or in small groups (maximum 2 people). We collected both quantitative and qualitative data through various means, including lab report scores, final project scores, system usage logs, the System Usability Scale (SUS), satisfaction questionnaires, and semi-structured interviews.

Specifically, we recorded the average score of the eight lab reports for each student, with grading criteria covering functionality implementation, C language code quality, and reporting standards across multiple dimensions; three instructors independently scored the final projects

based on completeness, innovativeness, technical difficulty, and code standardization, using the average score. For the experimental group, we recorded anonymized data on each student's interaction frequency with GenAI-TA, session duration, and question types (e.g., conceptual questions about pointers, code debugging, file operation guidance), while for the control group, we recorded the number of times students asked questions to the human TA during lab sessions. Additionally, students in the experimental group completed a 10-item SUS questionnaire (scoring 0 to 100) at the end of the semester to evaluate GenAI-TA's usability, both groups completed a 5-point Likert scale satisfaction questionnaire on their course learning experience and TA support effectiveness, and eight students were randomly selected from each group for in-depth interviews to understand their perceptions of the TA (AI or human), usage experience, and impact on their C language learning process.

4.3. Experimental Results

As shown in Table I, students in the experimental group achieved significantly higher scores in both the average lab report score and the final project score compared to the control group. Independent samples t-tests revealed that this difference was statistically significant ($p < 0.05$), indicating a positive impact of GenAI-TA's introduction on enhancing students' C language practical abilities and project completion quality.

Table 1. Comparison of practical performance metrics between the experimental group and the control group.

Evaluation Metri	Experimental Group (N=49)	Control Group (N=49)	t-value	p-value
Average Lab Report Score	89.5±5.2	84.1±6.8	4.312	0.001
Final Project Score	91.2±4.9	86.7±6.1	3.987	0.003

System logs indicated that students in the experimental group interacted with GenAI-TA an average of 15.3 times per week, which is significantly higher than the average number of questions asked by control group students to the human TA (approximately 2.1 times per week). Notably, the usage frequency of GenAI-TA increased significantly during non-working hours (evenings and weekends), accounting for over 65% of total interactions. This demonstrates that the AI Teaching Assistant greatly satisfied the students' need for "anytime, anywhere" C language learning support.

The average SUS score for GenAI-TA in the experimental group was 85.7 (SD=7.9), which is generally considered an "Excellent" level of usability. GenAI-TA System Usability (SUS) Score Distribution. In the satisfaction questionnaire, students in the experimental group rated

the "Timeliness of TA support" (average score 4.8/5.0) and the "Effectiveness of TA answers" (average score 4.6/5.0) significantly higher than the control group (which scored 3.2 and 4.1, respectively).

5. Discussion and Analysis

The experimental results of this study strongly confirm the positive role of GenAI-TA in the experimental instruction of the “Advanced Language Programming (C Language)” course.

Firstly, GenAI-TA significantly enhanced students' learning outcomes. The improvement in the experimental group's scores can be attributed to the following points: (1) Instant Feedback: Students could receive immediate answers when encountering C language bugs (e.g., compilation errors, logical errors) or conceptual queries (e.g., pointer usage, file operation principles), preventing prolonged periods of stagnation and frustration, thus maintaining learning continuity. (2) Deeper Exploration: The AI teaching assistant could provide relevant extended knowledge (e.g., advanced pointer applications, efficient file operation methods) and C language code examples based on students' questions, encouraging students to engage in more in-depth, exploratory learning. (3) Code Quality Improvement: GenAI-TA not only points out C language code errors but also explains the reasons behind them (e.g., syntax non-compliance, logical flaws) and offers suggestions for modifications that follow C language best practices, subtly fostering good programming habits in students.

Secondly, GenAI-TA greatly increased students' learning engagement and autonomy. The 24/7 availability feature broke the constraints of time and space, shifting learning behavior from being "concentrated in class" to "happening anytime, anywhere." In the interviews, one student from the experimental group mentioned: "Before, I had to wait until the next day's lab session to ask the teacher if I ran into a C language problem. Now, if I get stuck while coding C language programs late at night, I can ask the AI TA and get ideas within seconds. That feeling is fantastic." This indicates that GenAI-TA effectively supported students' personalized learning pace and promoted the cultivation of self-regulated learning ability in C language programming.

Thirdly, high usability and satisfaction are key to the successful application of the AI teaching assistant. The high SUS score of 85.7 confirms the success of our strategy to integrate GenAI-TA as a Geely University OJ Platform plugin. Students did not have to leave their familiar C language development environment, making AI assistance "readily accessible." In the interviews, students generally praised its conversational interaction style, feeling that it was "much more efficient than searching for fragmented C language answers online." This validates

the importance of good human-computer interaction design for the successful implementation of technology in education.

However, this study also uncovered some issues warranting attention. Some students mentioned in the interviews that they sometimes "subconsciously" ask the AI for complete C language code directly, rather than thinking through the problem first. This suggests that preventing student over-reliance is a crucial optimization direction for future work. Furthermore, although RAG technology significantly improved the accuracy of answers, GenAI-TA's responses could sometimes be generic when dealing with highly unconventional or innovative C language problems involving the complex interworking of multiple modules (e.g., integrating file operations with pointer applications), lacking profound insight.

The limitations of this study are as follows: (1) The experiment was limited to a single course ("Advanced Language Programming (C Language)") and a single institution, and the generalizability of its conclusions requires further verification. (2) The long-term impact of using GenAI-TA on students' independent C language problem-solving skills and creative thinking needs to be revealed through a longer-term longitudinal study.

6. Conclusion and Future Work

This study addressed the instructional pain points of the "Advanced Language Programming (C Language)" experimental course by designing, implementing, and empirically evaluating a Generative AI Teaching Assistant system named GenAI-TA. The research findings demonstrate that the system, by providing instant and personalized tutoring for C language programming, can significantly enhance students' learning outcomes, engagement, and satisfaction. This validates the immense potential of Generative AI in reforming the instruction of engineering practice-oriented courses, especially programming courses like "Advanced Language Programming (C Language)". This research provides a successful exemplar and a feasible technical solution for the deep integration of AI technology with specialized programming course experimental instruction.

Future work will focus on the following aspects:

(1) Intelligent Pedagogical Intervention: Incorporate a learner model into GenAI-TA to actively identify students' C language knowledge gaps (e.g., deficiencies in pointer operation, file handling, or data structure application) by analyzing their questioning patterns and C language coding behaviors. The system will then push personalized C language learning resources (e.g., targeted practice questions, code examples) and exercises, enabling a shift from

"passive Q&A" to "active guidance."

(2) Optimizing Interaction and Anti-Dependence Mechanism: Design a "Socratic" or "heuristic" answering mode. When the system detects potential student over-reliance (e.g., directly requesting complete C language code without independent thinking), it will prioritize offering conceptual hints and guiding questions rather than directly providing answers. This aims to cultivate students' critical thinking and independent C language problem-solving abilities.

(3) Generalization and Expansion: Generalize the GenAI-TA framework so that it can be rapidly adapted to other highly practical programming courses (such as "Web Development," "Machine Learning," etc.) by simply loading different course knowledge bases, thereby building a scalable AI Teaching Assistant platform.

(4) Conducting Larger-Scale Longitudinal Studies: Promote the application of the system in more universities and programming courses, and conduct long-term tracking research spanning several years to fully assess the long-term impact of Generative AI on students' C language professional competencies and lifelong learning capabilities.

We believe that with the continuous maturity of technology and the persistent optimization of pedagogical design, Generative AI will become a crucial engine driving the personalized, intelligent, and efficient development of higher education, particularly in the field of programming instruction.

References

[1] Krusche, S., & Alperowitz, L. (2018). Introduction of continuous delivery in multi-customer project courses. *In Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training* (pp. 31–40). IEEE. https://doi.org/10.1145/3183377.3183378

[2] Villegas-Ch, W., Román-Cañizares, M., & Palacios-Pacheco, X. (2020). Improvement of an education online model with the integration of machine learning and data analysis in an institution of higher education. *Sensors*, 20(5), Article 1396. https://doi.org/10.3390/s20051396

[3] Chiu, T. K. F. (2023). The impact of generative AI (GenAI) on practices of higher education: Challenges and opportunities. *Smart Learning Environments*, 10, Article 39. https://doi.org/10.1186/s40561-023-00258-w

[4] Kilde-Westberg, S., & Bøe, M. V. (2025). Generative AI as a lab partner: A case study in a university physics course. *Computers and Education: Artificial Intelligence*, 8, Article 100344. https://doi.org/10.1016/j.caeai.2024.100344

[5] Lau, S., & Guo, P. J. (2023). Banter: A pedagogically-appropriate AI tutor for computer science education. *In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM. https://doi.org/10.1145/3544548.3581564

[6] Essel, H. B., Vlachopoulos, D., Tachie-Menson, A., Johnson, E. E., & Baah, P. K. (2024).

The impact of generative AI (ChatGPT) on university students' engagement and learning gains: A systematic review. *Higher Education Quarterly*. Advance online publication. https://doi.org/10.1111/hequ.12504

[7] Brooke, J. (1996). SUS: A "quick and dirty" usability scale. *In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & I. L. McClelland (Eds.), Usability evaluation in industry* (pp. 189–194). Taylor & Francis.

[8] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *In Advances in Neural Information Processing Systems 33* (pp. 9459–9474). Curran Associates, Inc.

[9] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv*:2106.09685.

[10] Barke, S., James, M. B., & Polikarpova, N. (2023). Grounded copilot: How programmers interact with code-generating models. *Proceedings of the ACM on Programming Languages*, 7(OOPSLA2), 85–111. https://doi.org/10.1145/3618307

[11] Holmes, W., Persson, J., Chounta, I. R., Wasson, B., & Dimitrova, V. (2022). Ethics of artificial intelligence in education: Towards a community-wide framework. *International Journal of Artificial Intelligence in Education*, 32(3), 504–526. https://doi.org/10.1007/s40593-022-00292-9

[12] Farrokhnia, M., Konijn, E. A., Akyüz, N., & Beer, N. (2024). A favourable buddy or a fearful beast? A systematic review of ChatGPT's roles, benefits, and challenges in education. *Educational Technology Research and Development*, 72(1), 1–43. https://doi.org/10.1007/s11423-023-10298-w

Impressum

Founders	Zhengjie Gao, Xinyu Song
Editor in Chief	Ao Feng, Chengdu University of Information Technology, China
Executive Editor	Zhengjie Gao, Geely University of China, China
Editorial Board	Jing Hu, Huazhong University of Science and Technology, China Xiaohu Du, Huazhong University of Science and Technology, China Xiangkui Li, Harbin University of Science and Technology, China Zuopeng Liu, Goettingen University, Germany Xinyu Song, Geely University of China, China
Young Editorial Board	Min Liao, Geely University of China, China Tao Zheng, Geely University of China, China Chong Li, Chongqing University, China Ruiqin Fan, Sehan University, Korea Ziyang Liu, Jiangsu Normal University, China Qiwei Liu, Urumqi Vocational University, China Minqiu Kuang, Hunan Agricultural University, China
Published By	**Hong Kong Dawn Clarity Press Limited** Rm 9042, 9/F, Block B Chung Mei Centre, 15-17 Hing Yip Street, Kwun Tong, Kowloon, Hong Kong e-mail: ijaaa@dawnclarity.press ***International Journal of Advanced AI Applications*** is published monthly.
Editorial Policy	***International Journal of Advanced AI Applications*** is directed to the international communities of scientific researchers in artificial intelligence, computers and electronic, from the universities, research units and industry.To differentiate from other similar journals, the editorial policy of IJAAA encourages the submission of original scientific papers that focus on the integration of the advanced AI applications. In particular, the following topics are expected to be addressed by authors: (1) Natural Language Processing (NLP): Conversational AI, machine translation, sentiment analysis, and context-aware dialogue systems. (2) Smart Cities and IoT Integration: AI for traffic optimization, energy management, waste reduction, and urban infrastructure. (3) Autonomous Systems and Robotics: Self-driving vehicles, drones, industrial automation, and human-robot collaboration.

	(4) Edge AI and Distributed Systems: Real-time processing, federated learning, and low-latency AI at the network edge. (5) Creative and Generative AI: Art, music, and content generation using generative adversarial networks (GANs) and transformers. (6) AI in Education and Industry: Adaptive learning platforms, intelligent tutoring systems, and AI-driven supply chain optimization. Ethical and Explainable AI (XAI): Fairness, transparency, and accountability in real-world AI deployment.