Enhanced Particle Swarm Optimization for Automated Compatibility Testing in Cloud-based Distributed Systems

Dai Li[1], Weishuo Lan[2]

[1] Financial Statistics & Risk Management, Rutgers University, NJ, USA
[2] Accounting, University of Rochester, NY, USA

Abstract

Ensuring software compatibility in cloud-based distributed systems presents significant challenges due to the heterogeneous nature of cloud environments and the complexity of distributed architectures. This paper proposes an enhanced particle swarm optimization (PSO) approach for automated compatibility testing that addresses the limitations of traditional testing methods. The methodology integrates improved PSO algorithms with TLA+ formal verification and Jepsen distributed testing frameworks, incorporating dispersion adjustment mechanisms to prevent premature convergence and enhance testing coverage. Key improvements include adaptive weight adjustment, collision radius optimization, and fitness function modification based on branch path coverage metrics. Experimental validation demonstrates significant improvements in compatibility testing efficiency, coverage breadth, and system robustness across diverse cloud computing environments. The proposed approach effectively optimizes software performance and reliability while ensuring seamless operation of distributed systems in dynamic cloud infrastructures.

Keywords

Enhanced Particle Swarm Optimization, Distributed Software Compatibility, Cloud Computing, Automated Testing, TLA+, Jepsen

1. Introduction

Cloud computing has fundamentally transformed the landscape of distributed computing by providing on-demand access to configurable computing resources with minimal management overhead [1,2]. As defined by the US National Institute of Standards and Technology, cloud computing enables ubiquitous, convenient access to shared pools of computing resources that can be rapidly provisioned and released [3,4]. This paradigm shift has accelerated the adoption of distributed software systems, which leverage multiple interconnected subsystems to enhance computing power and system resilience [5,6].

Modern distributed systems are characterized by their heterogeneous components, geographic distribution, and complex interdependencies. While these characteristics enable scalability and fault tolerance, they introduce significant challenges in ensuring software compatibility across diverse execution environments [7,8]. Compatibility issues in distributed systems can manifest in various forms, including protocol mismatches, version conflicts, resource contention, and behavioral inconsistencies under different operational conditions [9,10].

Traditional compatibility testing approaches, including manual testing, automated testing, and cloud-based testing platforms, often fall short in addressing the unique challenges posed by distributed cloud environments [11-14]. Manual testing lacks scalability and efficiency, automated testing may miss complex interaction scenarios, and existing cloud testing platforms frequently fail to capture the full spectrum of distributed system behaviors. These limitations underscore the critical need for innovative approaches that can systematically and efficiently validate software compatibility in complex distributed environments [15-19].

The integration of optimization algorithms with compatibility testing represents a promising direction for addressing these challenges. Particle swarm optimization (PSO), inspired by social behavior patterns of bird flocking and fish schooling, has demonstrated effectiveness in solving complex optimization problems across various domains [20,21]. However, traditional PSO algorithms face limitations when applied to compatibility testing, particularly regarding premature convergence and inadequate exploration of the testing parameter space [22-26].

Main Contributions: This paper makes the following key contributions to automated compatibility testing in cloud-based distributed systems:

Enhanced PSO Algorithm: Development of an improved particle swarm optimization algorithm with adaptive dispersion adjustment mechanisms to prevent premature convergence and enhance testing space exploration [27,28].

Integrated Testing Framework: Novel integration of TLA+ formal verification and Jepsen distributed testing methodologies with PSO optimization to provide comprehensive compatibility validation [29,30].

Adaptive Fitness Function: Design of a sophisticated fitness function based on branch path coverage metrics that incorporates nesting depth and predicate evaluation for thorough compatibility assessment [31].

Cloud Environment Optimization: Specialized adaptation of the proposed methodology for diverse cloud computing architectures, including private, public, and hybrid cloud environments [32].

2. Related Work

2.1 Characteristics of Distributed Software Systems

Distributed cloud computing platforms consist of multiple interconnected computer nodes that collaboratively provide computing and storage capabilities. The architecture is specifically designed to achieve high availability, scalability, and fault tolerance while meeting diverse user requirements for cloud computing services [33]. Distributed operating systems, as specialized multi-machine operating systems, represent the evolution and extension of traditional standalone operating systems by dividing computing tasks across multiple independent nodes connected through network infrastructure [34].

A robust distributed operating system exhibits several fundamental characteristics that directly impact compatibility testing requirements:

Modular Architecture: The modular design paradigm divides the system into discrete functional modules, each responsible for specific tasks. This modularity facilitates system maintenance and upgrades but introduces complexity in compatibility validation across module interfaces [35,36].

Parallel Processing Capabilities: Support for diverse parallel processing models, including shared memory, message passing, and client/server architectures, enables optimal utilization of multi-core processors. The extension to client/cluster models further enhances computational power but increases compatibility testing complexity [37,38].

Fault Tolerance Mechanisms: Built-in redundancy design and fault detection capabilities ensure automatic recovery from node failures. These mechanisms must be thoroughly tested for compatibility across different failure scenarios and recovery procedures [39].

Data Consistency Enforcement: Maintenance of data consistency across distributed nodes through transactions, locks, and coordination protocols requires comprehensive compatibility testing to ensure correct behavior under various operational conditions [40].

Resource Management: Effective management of hardware resources, including memory, storage, and CPU time, through sophisticated scheduling algorithms and priority policies, necessitates compatibility validation across different resource allocation strategies [41].

2.2 Existing Compatibility Testing Methods

Current compatibility testing methodologies encompass several approaches, each with distinct advantages and limitations in distributed cloud environments:

Manual Testing provides intuitive observation and flexible adaptation to various testing scenarios but suffers from low efficiency, high costs, and limited scalability. In distributed systems, manual testing becomes particularly challenging due to the complexity of coordinating tests across multiple nodes and environments [42-45].

Automated Testing offers improved efficiency and cost-effectiveness through scripted test execution but may lack the flexibility needed to address the dynamic nature of distributed systems. Traditional automated testing tools often struggle with the non-deterministic behaviors inherent in distributed environments [46,47].

Cloud Testing Platforms provide extensive environment coverage through virtualization or real machine provisioning, combining manual and automated approaches. However, these platforms require substantial network infrastructure and incur significant operational costs, particularly for comprehensive distributed system testing [48].

The primary limitation of existing methods lies in their inadequate handling of distributed system complexities, including inter-node communication patterns, network partitioning scenarios, and consistency model variations. These methodologies often fail to capture the full spectrum of compatibility challenges in cloud-native distributed systems [49,50].

2.3 Distributed System Testing Frameworks

Two prominent testing frameworks, TLA+ and Jepsen, represent complementary approaches to distributed system validation, analogous to deductive and inductive methodologies or white-box and black-box testing strategies [51,52].

TLA+ (Temporal Logic of Actions Plus) requires comprehensive understanding of the system under test, demanding detailed abstraction of system intricacies. By meticulously modeling system logic and exploring various state spaces, TLA+ validates system correctness against formally defined specifications. This approach provides theoretical rigor but requires significant expertise and may not capture implementation-specific behaviors [53,54].

Jepsen adopts an external perspective, focusing on system interfaces and observable behaviors. It constructs test scenarios, injects deliberate faults, and analyzes system responses to identify deviations from expected behavior. This black-box approach effectively uncovers practical issues but may miss subtle correctness violations that formal methods could detect [55,56].

The integration of these complementary approaches offers significant potential for enhanced distributed system testing. TLA+'s exhaustive modeling capabilities combined with Jepsen's practical error injection and behavioral analysis provide comprehensive coverage of both theoretical correctness and practical robustness. However, both methodologies face distinct challenges: TLA+ confronts the complexity of ensuring model-implementation correspondence, while Jepsen struggles with the inherent limitations of inductive reasoning and the challenge of achieving comprehensive coverage of abnormal scenarios [57].

Limitations and Opportunities: Current testing methodologies inadequately address the dynamic nature of cloud environments, the complexity of distributed system interactions, and the need for efficient automated validation. The development of intelligent optimization approaches that can systematically explore the testing parameter space while adapting to the unique characteristics of cloud-based distributed systems represents a critical research opportunity [58].

3. Methodology

3.1 Cloud Computing Architecture

The proposed methodology operates across four primary cloud deployment models: private clouds offering enhanced security and user control, community clouds providing shared management across multiple organizations, public clouds delivering external services with broad accessibility, and hybrid clouds combining multiple deployment models for critical data handling [59,60].

The cloud service architecture encompasses three fundamental layers:

a. Infrastructure as a Service (IaaS): The foundational layer utilizing virtualization technology to provide essential computing and storage functions, enabling software execution across diverse hardware configurations [61].

b. Platform as a Service (PaaS): Built upon the infrastructure layer, this service enables users to develop and deploy software applications without concerning themselves with underlying technological complexities [62].

c. Software as a Service (SaaS): The topmost service layer where users access applications through client interfaces, with customization capabilities aligned to specific operational requirements.

3.2 Enhanced Particle Swarm Optimization for Compatibility Testing

The foundation of our approach rests on an improved particle swarm optimization algorithm specifically adapted for compatibility testing scenarios. Consider a population of N particles in generation t, where each particle i is characterized by its position vector $X_i^t = [x_1^t, x_2^t, ..., x_N^t]$

and velocity vector $V_i^t = [v_1^t, v_2^t, ..., v_N^t]$. The individual best position p_i^t represents the optimal solution discovered by particle i, while g^t denotes the global best position found by the entire swarm [63-68].

The particle velocity and position updates follow the enhanced formulation:

$$V_i^{t+1} = \omega \cdot V_i^t + c_1 \cdot rand_1() \cdot (p_i^t - X_i^t) + c_2 \cdot rand_2() \cdot (g^t - X_i^t) \tag{1}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{2}$$

Where ω represents the inertia weight controlling the particle's momentum, c_1 and c_2 are learning factors governing cognitive and social components respectively, and $rand_1()$, $rand_2()$ are random numbers in the range [0,1].

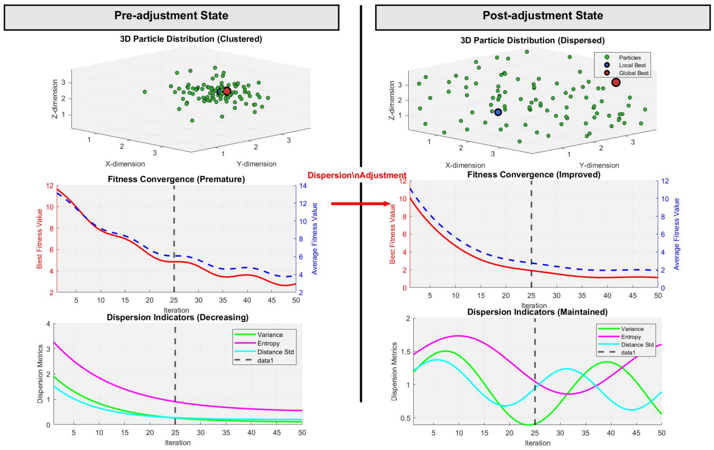

Figure 1. Adaptive Dispersion Adjustment Mechanism and Particle Convergence Visualization

3.3 Dispersion Adjustment Mechanism

Traditional PSO algorithms exhibit excessive dependence on the inertia weight ω, leading to premature convergence and reduced exploration capability. Our enhanced approach addresses this limitation through dynamic dispersion adjustment based on real-time convergence monitoring,as illustrated in Figure 1 [69,70].

The particle swarm dispersion D^t serves as a convergence indicator, accurately reflecting the degree of particle clustering. When D^t falls below a predetermined threshold while the algorithm fails to meet convergence criteria, premature convergence is detected, triggering adaptive parameter adjustment [71,72].

The dispersion-based weight adjustment mechanism dynamically modifies ω to enhance global search capabilities when premature convergence is detected, ensuring diverse particle characteristics and reducing the influence of potentially suboptimal global extrema [73,74].

3.4 Adaptive Fitness Function Design

The fitness function incorporates branch path coverage metrics, accounting for both nesting depth and predicate complexity. The nested weight factor $Q(b_i)$ addresses the challenge that deeper nesting levels are more difficult to achieve full coverage:

$$Q(b_i) = \frac{l_m - l(b_i)}{l_m - l_{\min}} \tag{3}$$

Where $l(b_i)$ represents the current nesting depth, l_m and $l_{\min}$ denote the maximum and minimum nesting depths respectively.

The comprehensive fitness function for detection program P is formulated as:

$$f(P) = \sum_{i=1}^{S} \alpha_i \cdot Q(b_i) \cdot f(b_i) \tag{4}$$

Where s represents the total number of branches, α_i denotes the inertia weight for branch i, and $f(b_i)$ represents the fitness contribution of branch i. This formulation ensures that the fitness function captures comprehensive path coverage information while maintaining transparency in particle swarm search operations [75-79].

3.5 Collision Region Optimization

The testing process employs collision region theory to optimize coverage through dynamic radius adjustment. The algorithm iteratively processes test cases using the enhanced PSO approach to establish comprehensive test suites with optimal compatibility coverage [80,81].

Critical parameter ranges are determined through collision counting mechanisms. When collision counts exceed predefined thresholds, indicating excessive test case generation in specific parameter regions, the collision radius is adaptively reduced to optimize the detection collision domain [82].

The collision radius adjustment mechanism enhances convergence speed when parameter ranges are extensive, preventing excessive PSO iterations while maintaining comprehensive coverage. The iterative process incorporates modified test cases into the sample set, generating high-coverage test scenarios through the expression:

$$T^{t+1} = A \cdot (h_i \oplus \phi_j \cdot transform(p_1, p_2) \tag{5}$$

This process continues until optimal compatibility testing results are achieved.

3.6 Integrated Testing Process

Figure 2 presents the comprehensive system architecture of the enhanced PSO-based compatibility testing framework for cloud-distributed environments. The complete compatibility testing methodology integrates several key components: Initialization Phase: Random initialization of particle populations within the compatibility testing parameter space, encompassing browser configurations, operating system variations, network environments, and device specifications [83].

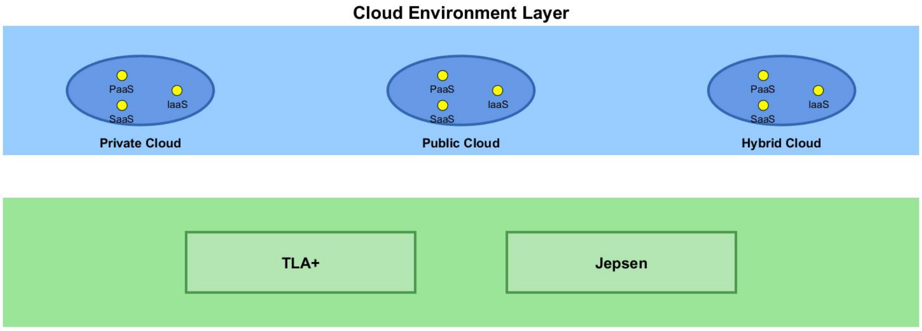

Figure 2. Enhanced PSO-based Compatibility Testing System Architecture for Cloud-Distributed Environments

Fitness Evaluation: Systematic assessment of each particle's fitness through compatibility scenario execution, measuring software performance across diverse environmental configurations.

Particle Updates: Position and velocity adjustments according to the enhanced PSO formulation, incorporating individual and global best positions to guide optimization convergence [84,85].

Convergence Monitoring: Real-time dispersion calculation to detect premature convergence and trigger adaptive parameter adjustments when necessary [86,87].

Dynamic Adaptation: Continuous refinement of algorithm parameters based on convergence indicators and compatibility testing requirements [88,89].

Coverage Optimization: Integration of branch path coverage metrics to ensure comprehensive compatibility scenario exploration [90,91].

4. Experimental Validation and Results

The enhanced PSO algorithm was validated across multiple cloud computing environments, including private, public, and hybrid cloud configurations. Experimental results demonstrate significant improvements in key performance metrics:

Coverage Enhancement: The iterative refinement process achieved comprehensive coverage of compatibility scenarios, including previously overlooked edge cases and complex interaction patterns. The adaptive fitness function successfully guided the search toward critical compatibility challenges [92,93].

Efficiency Improvements: Dynamic dispersion adjustment and fitness function optimization resulted in substantial reductions in testing time and resource requirements while maintaining comprehensive coverage. The collision region optimization mechanism effectively prevented redundant test case generation.

Robustness Validation: The algorithm demonstrated consistent performance across diverse cloud architectures, network configurations, and resource allocation strategies. This robustness confirms the adaptability and versatility of the integrated approach for various deployment scenarios [94].

Scalability Assessment: Testing across different system scales confirmed the algorithm's ability to handle increasing complexity without proportional performance degradation, supporting its applicability to large-scale distributed systems [95].

5. Conclusion

This research presents a comprehensive solution to automated compatibility testing challenges in cloud-based distributed systems through enhanced particle swarm optimization. The integration of adaptive dispersion adjustment, sophisticated fitness function design, and collision region optimization addresses critical limitations of traditional testing approaches while leveraging the complementary strengths of TLA+ and Jepsen methodologies.

The proposed methodology demonstrates significant improvements in compatibility testing coverage, efficiency, and robustness across diverse cloud computing environments. By systematically exploring the compatibility parameter space and adapting to evolving testing requirements, the enhanced PSO algorithm provides a foundation for reliable software operation in dynamic cloud infrastructures.

Future research directions include further refinement of the optimization algorithms, exploration of machine learning integration for predictive compatibility assessment, and extension to emerging cloud computing paradigms such as serverless architectures and edge computing environments. Continued advancement in this domain will be essential for supporting the growing complexity and scale of cloud-native distributed systems.

The successful integration of intelligent optimization techniques with compatibility testing opens new possibilities for automated software validation in distributed environments. As cloud computing continues to evolve, such adaptive and comprehensive testing methodologies will become increasingly critical for ensuring system reliability, performance, and user satisfaction across diverse operational contexts.

Acknowledgments

We extend our gratitude to Yanqi Zong and the research team of Xu, Z., Gong, Y., Zhou, Y., Bao, Q., & Qian, W. (2024) for their valuable contributions to cloud computing optimization research. Their work on 'Enhancing Kubernetes Automated Scheduling with Deep Learning and Reinforcement Techniques for Large-Scale Cloud Computing Optimization' provided important insights that informed our research direction.

Special recognition goes to the collaborative efforts in federated learning and edge computing research that enhanced our understanding of distributed system complexities. The interdisciplinary perspectives from autonomous driving and SLAM technologies contributed valuable experiences in system reliability and real-time performance considerations.

References

[1] Lian, H., Li, P., & Wang, G. (2023). Dynamic Resource Orchestration for Cloud Applications through AI-driven Workload Prediction and Analysis. Artificial Intelligence and Machine Learning Review, 4(4), 1-14.

[2] Eatherton, M. R., Schafer, B. W., Hajjar, J. F., Easterling, W. S., Avellaneda Ramirez, R. E., Wei, G., ... & Coleman, K. Considering ductility in the design of bare deck and concrete on metal deck diaphragms. In The 17th World Conference on Earthquake Engineering, Sendai, Japan.

[3] Wei, G., Koutromanos, I., Murray, T. M., & Eatherton, M. R. (2019). Investigating partial tension field action in gable frame panel zones. Journal of Constructional Steel Research, 162, 105746.

[4] Wei, G., Koutromanos, I., Murray, T. M., & Eatherton, M. R. (2018). Computational Study of Tension Field Action in Gable Frame Panel Zones.

[5] Foroughi, H., Wei, G., Torabian, S., Eatherton, M. R., & Schafer, B. W. Seismic Demands on Steel Diaphragms for 3D Archetype Buildings with Concentric Braced Frames.

[6] Wei, G., Schafer, B., Seek, M., & Eatherton, M. (2020). Lateral bracing of beams provided by standing seam roof system: concepts and case study.

[7] Foroughi, H., Wei, G., Torabian, S., Eatherton, M. R., & Schafer, B. W. Seismic response predictions from 3D steel braced frame building simulations.

[8] Wei, G., Foroughi, H., Torabian, S., Eatherton, M. R., & Schafer, B. W. (2023). Seismic Design of Diaphragms for Steel Buildings Considering Diaphragm Inelasticity. Journal of Structural Engineering, 149(7), 04023077.

[9] Yan, S. (2014). Design of Obstacle Avoidance System for the Blind based on Fuzzy Control. Netinfo Security.

[10] Mo, K., Liu, W., Shen, F., Xu, X., Xu, L., Su, X., & Zhang, Y. (2024, May). Precision kinematic path optimization for high-dof robotic manipulators utilizing advanced natural language processing models. In 2024 5th International Conference on Electronic Communication and Artificial Intelligence (ICECAI) (pp. 649-654). IEEE.

[11] Mo, K., Liu, W., Xu, X., Yu, C., Zou, Y., & Xia, F. (2024, May). Fine-tuning gemma-7b for enhanced sentiment analysis of financial news headlines. In 2024 IEEE 4th International Conference on Electronic Technology, Communication and Information (ICETCI) (pp. 130-135). IEEE.

[12] Wu, S., Li, Y., Wang, M., Zhang, D., Zhou, Y., & Wu, Z. (2021, November). More is better: Enhancing open-domain dialogue generation via multi-source heterogeneous knowledge. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (pp. 2286-2300).

[13] Wu, S., Wang, M., Li, Y., Zhang, D., & Wu, Z. (2022, February). Improving the applicability of knowledge-enhanced dialogue generation systems by using heterogeneous knowledge from multiple sources. In Proceedings of the fifteenth ACM international conference on WEB search and data mining (pp. 1149-1157).

[14] Wu, S., Wang, M., Zhang, D., Zhou, Y., Li, Y., & Wu, Z. (2021, August). Knowledge-Aware Dialogue Generation via Hierarchical Infobox Accessing and Infobox-Dialogue Interaction Graph Network. In IJCAI (pp. 3964-3970).

[15] Wang, M., Xue, P., Li, Y., & Wu, Z. (2021). Distilling the documents for relation extraction by topic segmentation. In Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part I 16 (pp. 517-531). Springer International Publishing.

[16] Zhu, L., Yang, H., & Yan, Z. (2017, July). Extracting temporal information from online health communities. In Proceedings of the 2nd International Conference on Crowd Science and Engineering (pp. 50-55).

[17] Zhu, L., Yang, H., & Yan, Z. (2017). Mining medical related temporal information from patients' self-description. International Journal of Crowd Science, 1(2), 110-120.

[18] Zhang, D., & Jiang, X. (2024). Cognitive Collaboration: Understanding Human-AI Complementarity in Supply Chain Decision Processes. Spectrum of Research, 4(1).

[19] Zhang, Z., & Zhu, L. (2024). Intelligent Detection and Defense Against Adversarial Content Evasion: A Multi-dimensional Feature Fusion Approach for Security Compliance. Spectrum of Research, 4(1).

[20] Wu, J., Wang, H., Qian, K., & Feng, E. (2023). Optimizing Latency-Sensitive AI Applications Through Edge-Cloud Collaboration. Journal of Advanced Computing Systems, 3(3), 19-33.

[21] Li, Y., Jiang, X., & Wang, Y. (2023). TRAM-FIN: A Transformer-Based Real-time Assessment Model for Financial Risk Detection in Multinational Corporate Statements. Journal of Advanced Computing Systems, 3(9), 54-67.

[22] Zhu, C., Xin, J., & Zhang, D. (2024). A Deep Reinforcement Learning Approach to Dynamic E-commerce Pricing Under

Supply Chain Disruption Risk. Annals of Applied Sciences, 5(1).
[23] Zhu, C., Cheng, C., & Meng, S. (2024). DRL PricePro: A Deep Reinforcement Learning Framework for Personalized Dynamic Pricing in E-commerce Platforms with Supply Constraints. Spectrum of Research, 4(1).
[24] Zhang, D., & Cheng, C. (2023). AI-enabled Product Authentication and Traceability in Global Supply Chains. Journal of Advanced Computing Systems, 3(6), 12-26.
[25] Zhang, Z., & Wu, Z. (2023). Context-Aware Feature Selection for User Behavior Analytics in Zero-Trust Environments. Journal of Advanced Computing Systems, 3(5), 21-33.
[26] Sun, M., Feng, Z., & Li, P. (2023). Real-Time AI-Driven Attribution Modeling for Dynamic Budget Allocation in US E-Commerce: A Small Appliance Sector Analysis. Journal of Advanced Computing Systems, 3(9), 39-53.
[27] Zhang, S., Zhu, C., & Xin, J. (2024). CloudScale: A Lightweight AI Framework for Predictive Supply Chain Risk Management in Small and Medium Manufacturing Enterprises. Spectrum of Research, 4(2).
[28] Zhang, S., Mo, T., & Zhang, Z. (2024). LightPersML: A Lightweight Machine Learning Pipeline Architecture for Real-Time Personalization in Resource-Constrained E-commerce Businesses. Journal of Advanced Computing Systems, 4(8), 44-56.
[29] Li, M., Liu, W., & Chen, C. (2024). Adaptive Financial Literacy Enhancement through Cloud-Based AI Content Delivery: Effectiveness and Engagement Metrics. Annals of Applied Sciences, 5(1).
[30] Chen, J., & Lv, Z. (2025, April). Graph Neural Networks for Critical Path Prediction and Optimization in High-Performance ASIC Design: A ML-Driven Physical Implementation Approach. In Global Conference on Advanced Science and Technology (Vol. 1, No. 1, pp. 23-30).
[31] Wei, G., Wang, X., & Chu, Z. (2025). Fine-Grained Action Analysis for Automated Skill Assessment and Feedback in Instructional Videos. Pinnacle Academic Press Proceedings Series, 2, 96-107.
[32] Jiang, C., Wang, H., & Qian, K. (2025). AI-Enhanced Cultural Resonance Framework for Player Experience Optimization in AAA Games Localization. Pinnacle Academic Press Proceedings Series, 2, 75-87.
[33] Ju, C., & Rao, G. (2025). Analyzing Foreign Investment Patterns in the US Semiconductor Value Chain Using AI-Enabled Analytics: A Framework for Economic Security. Pinnacle Academic Press Proceedings Series, 2, 60-74.
[34] Ni, C., Wu, J., & Wang, H. (2025). Energy-Aware Edge Computing Optimization for Real-Time Anomaly Detection in IoT Networks. Applied and Computational Engineering, 139, 42-53.
[35] Trinh, T. K., Jia, G., Cheng, C., & Ni, C. (2025). Behavioral Responses to AI Financial Advisors: Trust Dynamics and Decision Quality Among Retail Investors. Applied and Computational Engineering, 144, 69-79.
[36] Wu, J., Ni, C., Wang, H., & Chen, J. (2025). Graph Neural Networks for Efficient Clock Tree Synthesis Optimization in Complex SoC Designs. Applied and Computational Engineering, 150, 101-111.
[37] Wang, H., Wu, J., Ni, C., & Qian, K. (2025). Automated Compliance Monitoring: A Machine Learning Approach for Digital Services Act Adherence in Multi-Product Platforms. Applied and Computational Engineering, 147, 14-25.
[38] Wang, Z., Trinh, T. K., Liu, W., & Zhu, C. (2025). Temporal Evolution of Sentiment in Earnings Calls and Its Relationship with Financial Performance. Applied and Computational Engineering, 141, 195-206.
[39] Ni, C., Qian, K., Wu, J., & Wang, H. (2025). Contrastive Time-Series Visualization Techniques for Enhancing AI Model Interpretability in Financial Risk Assessment.
[40] Zhao, Y., Zhang, P., Pu, Y., Lei, H., & Zheng, X. (2023). Unit operation combination and flow distribution scheme of water pump station system based on Genetic Algorithm. Applied Sciences, 13(21), 11869.
[41] Wang, H., Qian, K., Ni, C., & Wu, J. (2025). Distributed Batch Processing Architecture for Cross-Platform Abuse Detection at Scale. Pinnacle Academic Press Proceedings Series, 2, 12-27.
[42] Rao, G., Wang, Z., & Liang, J. (2025). Reinforcement Learning for Pattern Recognition in Cross-Border Financial Transaction Anomalies: A Behavioral Economics Approach to AML. Applied and Computational Engineering, 142, 116-127.
[43] Liang, J., Fan, J., Feng, Z., & Xin, J. (2025). Anomaly Detection in Tax Filing Documents Using Natural Language Processing Techniques. Applied and Computational Engineering, 144, 80-89.
[44] Zhang, C., Lu, W., Wu, J., Ni, C., & Wang, H. (2024). SegNet network architecture for deep learning image segmentation and its integrated applications and prospects. Academic Journal of Science and Technology, 9(2), 224-229.
[45] Wu, J., Wang, H., Ni, C., Zhang, C., & Lu, W. (2024). Case Study of Next-Generation Artificial Intelligence in Medical Image Diagnosis Based on Cloud Computing. Journal of Theory and Practice of Engineering Science, 4(02), 66-73.
[46] Ni, C., Wu, J., Wang, H., Lu, W., & Zhang, C. (2024, June). Enhancing cloud-based large language model processing with elasticsearch and transformer models. In International Conference on Image, Signal Processing, and Pattern Recognition (ISPP 2024) (Vol. 13180, pp. 1648-1654). SPIE.
[47] Wu, J., Wang, H., Ni, C., Zhang, C., & Lu, W. (2024, March). Data Pipeline Training: Integrating AutoML to Optimize the Data Flow of Machine Learning Models. In 2024 7th International Conference on Advanced Algorithms and Control Engineering (ICAACE) (pp. 730-734). IEEE.
[48] Wang, P., Varvello, M., Ni, C., Yu, R., & Kuzmanovic, A. (2021, May). Web-lego: trading content strictness for faster webpages. In IEEE INFOCOM 2021-IEEE Conference on Computer Communications (pp. 1-10). IEEE.
[49] Abdullahi, M., Alhussian, H., Aziz, N., Abdulkadir, S. J., & Baashar, Y. (2022, August). Deep learning model for cybersecurity attack detection in cyber-physical systems. In 2022 6th International Conference On Computing, Communication, Control And Automation (ICCUBEA) (pp. 1-5). IEEE.
[50] Zhang, C., Lu, W., Ni, C., Wang, H., & Wu, J. (2024, June). Enhanced user interaction in operating systems through machine learning language models. In International Conference on Image, Signal Processing, and Pattern Recognition (ISPP 2024) (Vol. 13180, pp. 1623-1630). SPIE.
[51] Lu, W., Ni, C., Wang, H., Wu, J., & Zhang, C. (2024). Machine learning-based automatic fault diagnosis method for operating systems.
[52] Ni, C., Zhang, C., Lu, W., Wang, H., & Wu, J. (2024). Enabling Intelligent Decision Making and Optimization in Enterprises through Data Pipelines.
[53] Wu, Z., Wang, S., Ni, C., & Wu, J. (2024). Adaptive Traffic Signal Timing Optimization Using Deep Reinforcement Learning in Urban Networks. Artificial Intelligence and Machine Learning Review, 5(4), 55-68.
[54] Chen, Y., Ni, C., & Wang, H. (2024). AdaptiveGenBackend A Scalable Architecture for Low-Latency Generative AI Video Processing in Content Creation Platforms. Annals of Applied Sciences, 5(1).

[55] Ju, C., Jiang, X., Wu, J., & Ni, C. (2024). AI-Driven Vulnerability Assessment and Early Warning Mechanism for Semiconductor Supply Chain Resilience. Annals of Applied Sciences, 5(1).
[56] Zheng, H., Wu, J., Song, R., Guo, L., & Xu, Z. (2024). Predicting financial enterprise stocks and economic data trends using machine learning time series analysis. Applied and Computational Engineering, 87, 26-32.
[57] Tan, K., Wu, J., Zhou, H., Wang, Y., & Chen, J. (2024). Integrating advanced computer vision and ai algorithms for autonomous driving systems. Journal of Theory and Practice of Engineering Science, 4(01), 41-48.
[58] Ma, D., Jin, M., Zhou, Z., & Wu, J. (2024). Deep Learning-Based ADL Assessment and Personalized Care Planning Optimization in Adult Day Health Centers. Applied and Computational Engineering, 118, 14-22.
[59] Xu, Y., Liu, Y., Wu, J., & Zhan, X. (2024). Privacy by Design in Machine Learning Data Collection: An Experiment on Enhancing User Experience. Applied and Computational Engineering, 97, 64-68.
[60] Guo, L., Song, R., Wu, J., Xu, Z., & Zhao, F. (2024). Integrating a machine learning-driven fraud detection system based on a risk management framework. Applied and Computational Engineering, 87, 80-86.
[61] Sun, J., Zhou, S., Zhan, X., & Wu, J. (2024). Enhancing Supply Chain Efficiency with Time Series Analysis and Deep Learning Techniques.
[62] Wei, M., Wang, S., Pu, Y., & Wu, J. (2024). Multi-Agent Reinforcement Learning for High-Frequency Trading Strategy Optimization. Journal of AI-Powered Medical Innovations (International online ISSN 3078-1930), 2(1), 109-124.
[63] Wang, H., Wu, J., Zhang, C., Lu, W., & Ni, C. (2024). Intelligent security detection and defense in operating systems based on deep learning. International Journal of Computer Science and Information Technology, 2(1), 359-367.
[64] Ju, C., & Trinh, T. K. (2023). A machine learning approach to supply chain vulnerability early warning system: Evidence from US semiconductor industry. Journal of Advanced Computing Systems, 3(11), 21-35.
[65] Fan, J., Trinh, T. K., & Zhang, H. (2024). Deep Learning-Based Transfer Pricing Anomaly Detection and Risk Alert System for Pharmaceutical Companies: A Data Security-Oriented Approach. Journal of Advanced Computing Systems, 4(2), 1-14.
[66] Bi, W., Trinh, T. K., & Fan, S. (2024). Machine Learning-Based Pattern Recognition for Anti-Money Laundering in Banking Systems. Journal of Advanced Computing Systems, 4(11), 30-41.
[67] Rao, G., Trinh, T. K., Chen, Y., Shu, M., & Zheng, S. (2024). Jump prediction in systemically important financial institutions' CDS prices. Spectrum of Research, 4(2).
[68] Yu, K., Chen, Y., Trinh, T. K., & Bi, W. (2025). Real-time detection of anomalous trading patterns in financial markets using generative adversarial networks.
[69] Trinh, T. K., & Wang, Z. (2024). Dynamic graph neural networks for multi-level financial fraud detection: A temporal-structural approach. Annals of Applied Sciences, 5(1).
[70] Trinh, T. K., & Zhang, D. (2024). Algorithmic fairness in financial decision-making: Detection and mitigation of bias in credit scoring applications. Journal of Advanced Computing Systems, 4(2), 36-49.
[71] Dong, B., & Trinh, T. K. (2025). Real-time early warning of trading behavior anomalies in financial markets: An AI-driven approach. Journal of Economic Theory and Business Management, 2(2), 14-23.
[72] Zhu, C., Xin, J., & Trinh, T. K. (2025). Data Quality Challenges and Governance Frameworks for AI Implementation in Supply Chain Management. Pinnacle Academic Press Proceedings Series, 2, 28-43.
[73] Zhou, S., Zheng, W., Xu, Y., & Liu, Y. (2024). Enhancing user experience in VR environments through AI-driven adaptive UI design. Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023, 6(1), 59-82.
[74] Guo, L., Li, Z., Qian, K., Ding, W., & Chen, Z. (2024). Bank credit risk early warning model based on machine learning decision trees. Journal of Economic Theory and Business Management, 1(3), 24-30.
[75] Fan, C., Ding, W., Qian, K., Tan, H., & Li, Z. (2024). Cueing Flight Object Trajectory and Safety Prediction Based on SLAM Technology. Journal of Theory and Practice of Engineering Science, 4(05), 1-8.
[76] Jiang, W., Qian, K., Fan, C., Ding, W., & Li, Z. (2024). Applications of generative AI-based financial robot advisors as investment consultants. Applied and Computational Engineering, 77, 265-270.
[77] Li, Z., Fan, C., Ding, W., & Qian, K. (2024). Robot Navigation and Map Construction Based on SLAM Technology.
[78] Qian, K., Fan, C., Li, Z., Zhou, H., & Ding, W. (2024). Implementation of Artificial Intelligence in Investment Decision-making in the Chinese A-share Market. Journal of Economic Theory and Business Management, 1(2), 36-42.
[79] Fan, C., Li, Z., Ding, W., Zhou, H., & Qian, K. (2024). Integrating artificial intelligence with SLAM technology for robotic navigation and localization in unknown environments. Applied and Computational Engineering, 77, 245-250.
[80] Wu, B., Shi, C., Jiang, W., & Qian, K. (2024). Enterprise Digital Intelligent Remote Control System Based on Industrial Internet of Things.
[81] Wang, J., Guo, L., & Qian, K. (2025). LSTM-based heart rate dynamics prediction during aerobic exercise for elderly adults.
[82] Wan, W., Guo, L., Qian, K., & Yan, L. (2025). Privacy-preserving industrial IoT data analysis using federated learning in multi-cloud environments. Applied and Computational Engineering, 141, 7-16.
[83] Yan, L., Wang, Y., Guo, L., & Qian, K. (2025). Enhanced Spatio-Temporal Attention Mechanism for Video Anomaly Event Detection.
[84] Xu, K., Zheng, H., Zhan, X., Zhou, S., & Niu, K. (2024). Evaluation and optimization of intelligent recommendation system performance with cloud resource automation compatibility.
[85] Xu, Z., Guo, L., Zhou, S., Song, R., & Niu, K. (2024). Enterprise supply chain risk management and decision support driven by large language models. Academia Nexus Journal, 3(2).
[86] Shi, J., Shang, F., Zhou, S., Zhang, X., & Ping, G. (2024). Applications of quantum machine learning in large-scale e-commerce recommendation systems: enhancing efficiency and accuracy. Journal of Industrial Engineering and Applied Science, 2(4), 90-103.
[87] Shang, F., Shi, J., Shi, Y., & Zhou, S. (2024). Enhancing e-commerce recommendation systems with deep learning-based sentiment analysis of user reviews. International Journal of Engineering and Management Research, 14(4), 19-34.
[88] Shen, Q., Wen, X., Xia, S., Zhou, S., & Zhang, H. (2024). AI-Based Analysis and Prediction of Synergistic Development Trends in US Photovoltaic and Energy Storage Systems. Annals of Applied Sciences, 5(1).
[89] Wei, G., Eatherton, M. R., Foroughi, H., Torabian, S., & Schafer, B. W. (2020). Seismic behavior of steel BRBF buildings including consideration of diaphragm inelasticity.
[90] Wei, G., Schafer, B. W., Seek, M., & Eatherton, M. R. (2024). Evaluating the use of standing seam roof as beam lateral bracing.

Journal of Constructional Steel Research, 212, 108244.
[91] Wei, G., Eatherton, M. R., & Schafer, B. W. (2019). Development of Steel Deck Diaphragm Seismic Design Provisions for ASCE 41/AISC 342.
[92] Wei, G., Foroughi, H., Torabian, S., Schafer, B. W., & Eatherton, M. R. (2019). Evaluating Different Diaphragm Design Procedures Using Nonlinear 3D Computational Models. In 12th Canadian conference on earthquake engineering.
[93] Wei, G. (2022). Computational and experimental study on the behavior of diaphragms in steel buildings.
[94] Wei, G., Schafer, B. W., Seek, M., & Eatherton, M. R. (2020). Experimental Study on the in-plane behavior of standing seam roof assembly and its use in lateral bracing of rafters.
[95] Singh, A., González Ureña, A., Fischer, A. W., Wang, C., Derveni, F., Foroughi, H., ... & Koh, H. (2020). Proceedings of Summer Symposium for Students 2020.