

Performance Bottleneck Analysis and Optimization of an Order Book Event-Driven Stream Processing Framework—An Integrated Study for Low-Latency Matching and Real-Time Risk Control

Wenjie Jiang

Guangzhou College of Commerce, School of Information Technology & Engineering, Guangzhou, 511363

Keywords: Order book; event-driven; stream processing framework; performance bottleneck; state management; low latency

Abstract: Order book data is characterized by fine-grained events, strong peak bursts, dense state access, and strict time-series constraints, which dictates that its stream processing framework must simultaneously satisfy low latency, strong consistency, and high throughput. Focusing on the issue of "performance bottleneck analysis and optimization of order book event-driven stream processing frameworks," this paper, based on a review of research on stream processing systems, checkpointing protocols, elastic scheduling, order book modeling, and event stream prediction over the past three years, proposes a five-layer bottleneck analysis framework addressing market data access, event routing, state management, checkpointing, and elastic scaling. Furthermore, considering the load structure of order book event streams, a three-objective model of throughput, latency, and resources is constructed, and collaborative optimization strategies including fine-grained partitioning, hotspot isolation, asynchronous state access, layered checkpointing, and hybrid resource deployment are presented. The study argues that the core contradiction in the order book scenario is not insufficient computation of a single operator, but rather the superposition of high-frequency state read/write, out-of-order event correction, peak load propagation, and the overhead of fault tolerance mechanisms. Through comparison of literature data and mechanism analysis, it can be seen that the joint optimization of state decoupling, predictive elasticity and topology-level parameters can significantly improve end-to-end latency stability and enhance processing resilience under extreme conditions.

1 Introduction

In scenarios such as high-frequency trading, market making, real-time risk control, and intraday

monitoring, order book event streams arrive continuously at the millisecond or even microsecond level, including various types of signals such as pending orders, order cancellations, executions, snapshot updates, and derivative risk control triggers. Unlike traditional business streams, order book event streams must maintain the temporal semantics of events and continuously increment the order book status. Therefore, its processing is essentially a combination of "high-arrival-rate input, high-frequency state access, and low-tolerance output." In recent years, stream processing systems have continuously evolved in terms of state management, fault tolerance, elastic scheduling, and transactional execution [1-5], while order book research has also developed rapidly in terms of predictive representation, information persistence, and event generation modeling [6-10]. However, these two research lines have not been fully integrated in addressing the performance bottleneck problem of event-driven frameworks for order books.

In engineering terms, most systems can maintain high throughput under normal loads. However, when faced with highly volatile states such as opening, message publishing, rapid price increases, or continuous order cancellations, end-to-end latency can increase dramatically due to backend access to state data, backpressure propagation, checkpoint blocking, hotspot key skew, and scaling lag. This is especially true for computational topologies that primarily use order books as the core state object, updating data such as optimal sell and buy prices, transaction queue lengths, and imbalance levels in real time. If a node experiences local congestion, it propagates upwards through the data flow topology, leading to strategy execution delays or false alarms in risk control. Therefore, this paper analyzes the main bottlenecks of order book event-driven stream processing systems from the framework layer rather than the single model layer, and presents an optimization solution that is both theoretically explainable and engineering-practical.

2. Current Status Analysis of the Research Topic

Research in recent years has found that the development focus of stream processing systems is on seven aspects, namely state management, fault tolerance protocol, elastic scaling, load balancing, transaction consistency, event time processing and resource cost control. Fragkoulis et al. systematically reviewed modern stream processing systems from operator execution to high availability, and believed that state, time and fault tolerance together determine the upper limit of the system [1]. Henning and Hasselbring found in their comparison under microservice deployment conditions that different mainstream stream processing frameworks have significant differences in resource consumption rate and available throughput, and none of them can be considered the best under high load. Therefore, the order book scenario cannot simply apply the general stream processing benchmark, and must be specially designed according to the load characteristics of dense state and high peak intensity.

Siachamis et al. conducted a systematic analysis of fault tolerance and state backends, compared three types of checkpoint protocols, and concluded that coordinated checkpoints perform better under uniform load, but non-coordinated schemes have potential when the key distribution is skewed [3]. Sun et al. proposed Sgp-Stream, which uses scheduling, grouping and parallelism to achieve continuous performance improvement, indicating that adjusting a single parameter is difficult to achieve excellent performance [4]. Wladdimiro et al. proposed the predictive elastic control framework PA-SPS, which combines load prediction and reconfiguration strategies, so that the system can prepare resources in advance when there is a sudden surge in traffic [5]. From these results, it can be seen that the performance optimization of modern stream processing frameworks has changed from single-point

parameter tuning to topology-level collaborative optimization.

Based on large-scale empirical experiments, Luccese et al. showed that the short-term predictability of returns and the representation method of the order book are strongly correlated, and a stable representation structure can significantly improve the performance of high-frequency prediction [6]. Briola et al. explained the persistence of the order book information and the differences in the hierarchical structure from the perspective of microstructure, providing a reference for state compression and multi-layer depth maintenance in the process of streaming feature extraction [9]. Zheng et al. proposed LOBDIF, which learns the joint distribution of time and event based on the diffusion model to establish a modeling method for irregular event flow [10]. Although the main purpose of these studies is prediction or generation, they all show two common engineering facts: first, the arrival of events is not a stationary Poisson process, but has obvious clustering and stage characteristics; second, the order book state is not a simple sliding window statistical result, but is based on a complex state object with multiple levels, multiple event types and strong path dependence. Therefore, if the stream processing framework is still designed using the general log stream approach, it will be limited in terms of parsing, routing, state access and fault tolerance overhead.

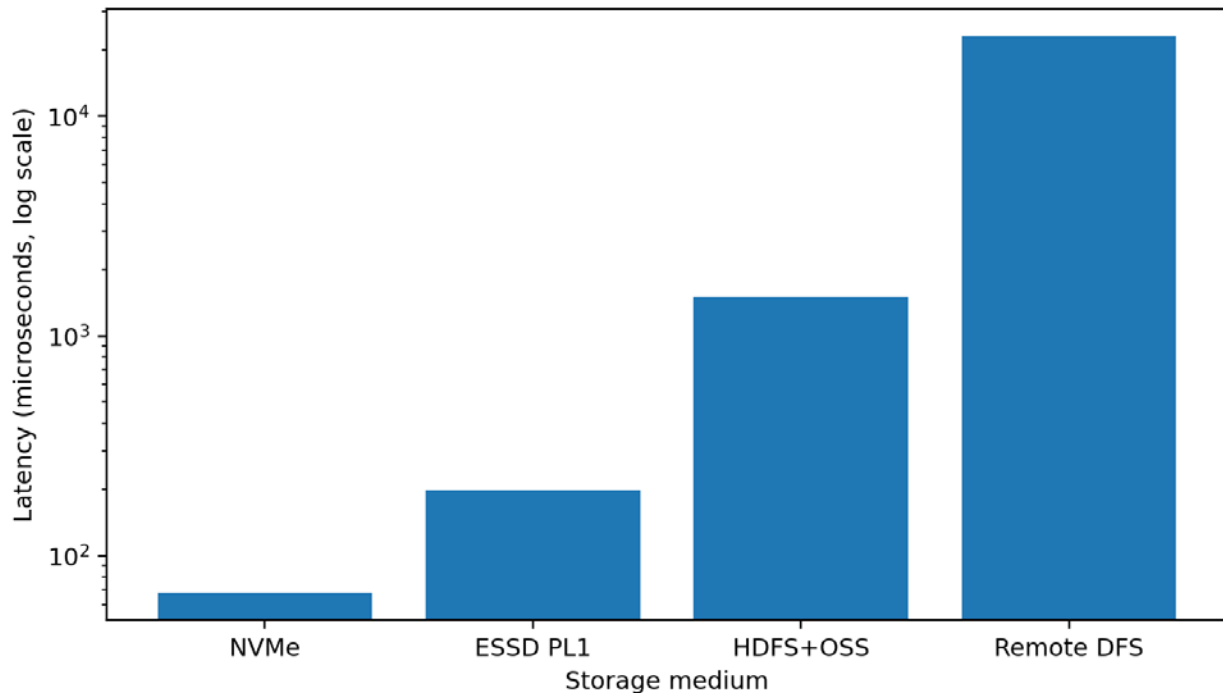


Figure 1 compares the access latency of storage media in different states, with the vertical axis using a logarithmic scale.

Figure 1 is based on Table 1 from the Flink 2.0 paper. Local NVMe and local ESSD accesses are in the microsecond range, while access times under remote HDFS or remote OSS reach milliseconds to tens of milliseconds. When each event in an order book scenario involves reading the order book status, price changes, and writing back derived features, the differences in the state backend media are amplified exponentially, ultimately leading to increased P99 latency and accumulated backpressure. Therefore, order book stream processing systems should not only focus on operator CPU utilization but

also minimize the frequency of cross-node and cross-media state accesses, using hot state localization and asynchronous prefetching to reduce tail latency.

Table 1 Recent studies and engineering implications

Study	Focus	Reported result	Implication
Henning 2024	Framework scalability	Up to 1M msg/s; near-linear scaling	Framework choice changes cost curve
CheckMate 2024	Checkpoint protocols	Skew can favor uncoordinated schemes	Fault tolerance must match workload shape
Sun 2024	Sgp orchestration	Single-level tuning reaches an upper limit	Joint tuning is necessary
Mei 2025	Disaggregated state	Remote state needs async hiding	State path dominates tail latency
Psarakis 2025	Transactional dataflows	Order-of-magnitude gain over baselines	State-placement and execution model matter

Table 1 reveals a clear trend in recent years: performance bottlenecks in the framework are increasingly less caused by insufficient computing power and more frequently stem from coupling issues between state access paths, checkpoint protocols, and scheduling strategies. For highly stateful event streams like order books, simply increasing parallelism while neglecting key grouping, state storage, and checkpoint frequency adjustments can easily lead to increased throughput but decreased tail latency. Therefore, order book framework optimization should shift towards multi-objective, multi-level collaboration.

3. Problem identified: The core performance bottleneck of the order book event-driven framework

Based on existing research and the characteristics of order book workload, this paper summarizes performance bottlenecks into five types. First, the bottleneck of access and parsing. When multiple transaction channels input, message encoding/unpacking, event normalization, and timestamp correction all cause significant CPU consumption. The small batch processing granularity also leads to increased latency for the first byte. Second, the bottleneck of routing and partitioning. If securities codes or price levels are used as the basis for grouping, hot contracts or popular stocks will cause task slot saturation during fluctuations, resulting in an overloaded task slot. Third, the bottleneck of state access. The order book state needs to maintain updates to multiple price and volume levels, queue positions, order cancellation counts, and derived indicators. Under frequent read/write operations, the probability of local cache loss, the cost of remote state access, and serialization overhead become the main latency factors. Fourth, the defects of tolerance and consistency. To meet the needs of auditing and risk control traceability, the system generally uses higher checkpoints or transaction semantics, which generates more I/O and synchronization blocking. Fifth, the constraint of elastic expansion and contraction. Market shocks often occur suddenly. If the decision to expand capacity lags behind the increase in traffic, the system will have already become unstable during the back pressure process, and it will be difficult for new resources to eliminate the backlog at the tail in a timely manner.

This paper uses metrics such as throughput, utilization, queue length, checkpoint overhead, and P99 latency to create an analytical framework. Let λ_{in} be the number of order book events arriving per unit time, T_{e2e} be the end-to-end latency at the topology level, and ρ_i be the utilization of each operator. Then, whether the system is close to congestion can be characterized by the following

relationship.

$$\lambda_{in} = N_e / T \quad (1)$$

The number of events arriving per unit time is the sum of all events N_e within the observation window, with T representing the window length. λ_{in} increases rapidly during the opening, closing, and macroeconomic announcement periods; this is the first signal of flexible capacity expansion and backpressure control.

$$T_{e2e} = T_{parse} + T_{route} + T_{state} + T_{sync} + T_{emit} \quad (2)$$

Equation (2) decomposes the end-to-end latency into five parts: parsing, routing, state access, synchronization fault tolerance, and result output. In the order book scenario, T_{state} and T_{sync} often account for a much higher proportion than ordinary log streams, so the phenomenon of "fast operators but slow system" often occurs.

$$\rho_i = \lambda_i / (\mu_i \cdot p_i) \quad (3)$$

Equation (3) gives the effective utilization rate of the i -th operator, where λ_i is the input rate, μ_i is the single parallel instance service rate, and p is the number of parallel instances. When ρ_i is close to 1 for a long time, it means that this operator has become a backpressure source. The actual ρ_i of hot order book keys is larger than the average value.

$$L_q = \rho_i^2 / (1 - \rho_i) \quad (4)$$

Equation (4) uses a simplified queuing relationship to characterize the trend of a sharp increase in queue length during congestion. Although the real system is not an ideal M/M/1, this equation can show that when the utilization rate increases from 0.8 to 0.95, the waiting queue will not increase linearly, but will instead increase suddenly. This is the most fundamental reason for the increased tail delay in the order book system.

$$J = \alpha \cdot (T_{99} / SLO) + \beta \cdot (C_{res} / C_{budget}) + \gamma \cdot (R_{ckpt} / R_{base}) \quad (5)$$

Equation (5) defines the comprehensive optimization objective J , where T_{99} is the P99 delay, C_{res} is the resource cost, R_{ckpt} is the checkpoint occupancy ratio, and α , β , and γ are weights. The optimization strategies proposed later in the paper all revolve around minimizing J under the SLO constraint.

This shows that the order book model is not a single bottleneck. Even if scaling reduces ρ_i , the state backend will still frequently access nodes, so T_{state} still plays a crucial role. Even if the local cached T_{state} is increased, if the checkpoint cycle is too short, T_{sync} will again become the main constraint. Therefore, measures should be taken simultaneously from four aspects: topology, state, protocol, and resources.

Figure 2 is based on Table II-V from the IC2E 2025 paper. The results show that under steady-state conditions, the latency differences between Dedicated, Burstable, and Combined deployments are not significant; however, under short bursts, long bursts, and real LogHub loads, the latency of the burstable deployment increases dramatically, with a difference of orders of magnitude in long bursts. These results have direct implications for order book scenarios: core channels for low-latency transactions should not rely excessively on shared or credit-based computing resources, but should instead adopt a hybrid deployment approach of "dedicated resources for the critical path + elastic resources for auxiliary channels" to maintain tail latency stability during extreme market conditions.

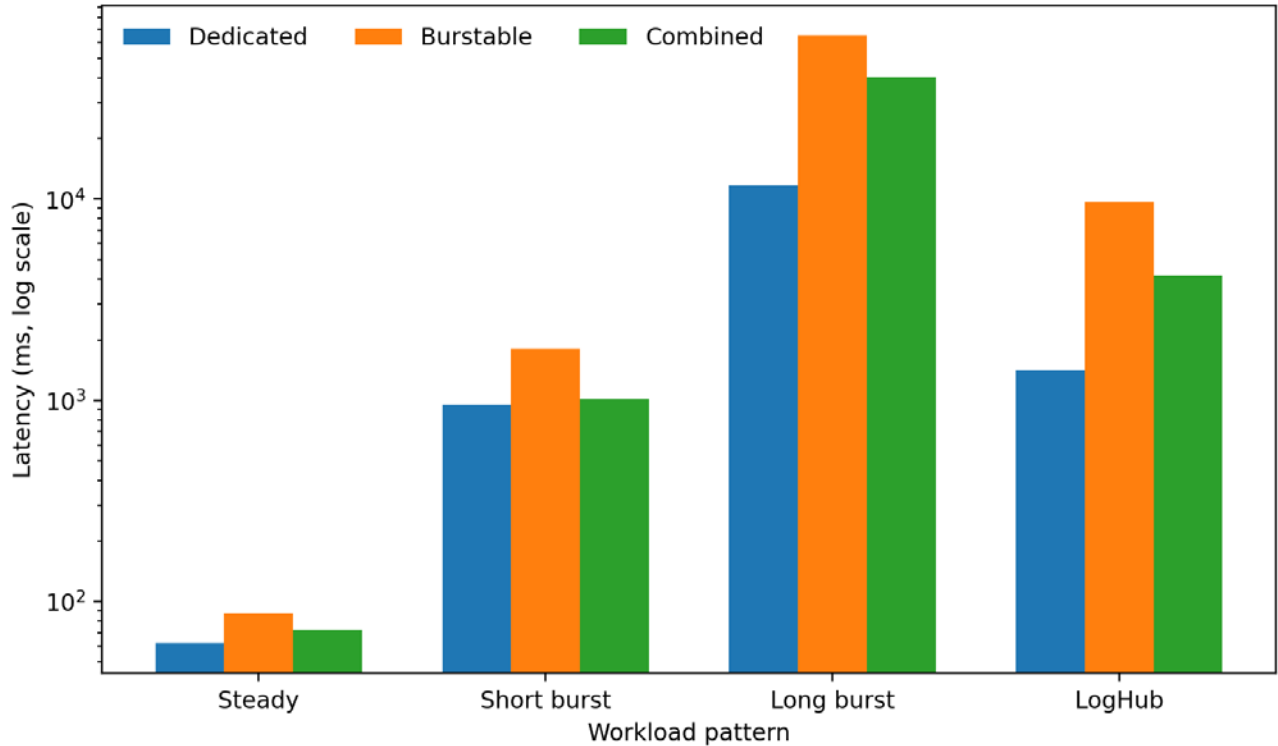


Figure 2 compares the processing latency of the three deployment methods under different load conditions (vertical axis is logarithmic scale).

Table 2-Bottleneck-to-optimizationMapping

Bottleneck	Symptom	Root cause	Optimization
Ingress parse	CPU spikes	Heavy decoding and normalization	Binary codec + vectorized parsing
Hot partition	P99 jump	Key skew and burst clustering	Adaptive repartition + hotspot isolation
State access	Tail latency	Remote reads and serialization	Local cache + async prefetch
Checkpoint	Periodic stalls	Synchronous barriers and large snapshots	Incremental checkpoint + protocol adaptation
Elasticity lag	Backpressure	Reactive scaling after backlog forms	Predictive scaling + warm standby

Table 2 shows the one-to-one correspondence between performance bottlenecks and optimization methods. It reveals that truly effective improvement methods do not simply pursue the optimal performance of a single metric, but rather select appropriate control granularity based on the type of bottleneck. Regarding state access overhead, the ideal approach is not to expand the cache entirely, but rather to employ strategies such as hot key isolation, local snapshots, and asynchronous prefetching to compress the critical path. For elastic lag, predictive scaling and instance preheating should be used to allocate resources before peak periods occur.

4. Problem Solving/Strategy: Collaborative Optimization Based on Order Book Event Flow

Based on the preceding analysis, this paper presents a four-layer collaborative optimization strategy for an order book event-driven stream processing framework. The first strategy is topology layer

optimization. The data flow is split into two parts: the "core trading link" and the "auxiliary analysis link." The core link only handles market data access, order book status updates, risk threshold determination, and necessary signal output, avoiding the use of significant CPU resources for non-critical derived indicators. The auxiliary link primarily handles complex feature calculations, playback, archiving, and preparation of offline retraining data. This topology layering significantly shortens the critical path.

Second, the state layer is improved. Since the order book has multiple price and quantity levels and the queue state is accessed frequently, it is necessary to create a layered state structure. Hot states are kept in local memory or local high-speed disks, while cold states are asynchronously written to the remote backend. The most active price, the best buy and sell order, and the short window imbalance are locally cached and structured and compressed to reduce the number of serializations and remote retrievals. Combining the characteristics of Flink 2.0 in which computation and state are decoupled but the latency is hidden by the asynchronous mechanism [7], the order book system can use local caching and prefetching to cache and prefetch important key spaces while maintaining decoupling.

The third type is protocol improvement. Checkpoints and consistency guarantees are two indispensable aspects in low-latency transaction scenarios, but this does not mean that only a fixed high-frequency coordinated scheme can be used. According to CheckMate's conclusion [3], when the load is highly skewed and there are local hot spots, non-coordinated or hybrid mechanisms are more advantageous. Therefore, the order book system uses checkpoint frequency, barrier propagation method, and incremental snapshot granularity as adjustable parameters and adjusts them according to the load pattern. For the risk control audit link, a higher level of consistency can be maintained, and for the statistics and monitoring link, a lower synchronization intensity can be used to reduce the degree of blockage.

Third, improvements are needed in resource management and scheduling. The joint optimization framework and predictive scheduling approach of PA-SPS proposed by Sun et al. demonstrate that under high load, simply increasing resources based on CPU or queue length is insufficient to solve the problem effectively. This paper suggests, based on an order book event intensity prediction model, using order cancellation clustering rate, market imbalance volatility, and transaction message density to proactively initiate resource preheating and key remapping. Dedicated instances are used for the core transaction chain to ensure stable tail latency, while Combined resources are used to control costs for feature concatenation, archiving, and non-critical asynchronous analysis.

From an implementation perspective, these four levels of strategies need to form a closed-loop system. First, the load-aware module estimates the distribution of the future short window λ_{in} and hotspot keys. Then, the optimizer provides the parallelism, partitioning scheme, checkpoint cycle, and state placement scheme according to equation (5), and finally sends the results to the execution layer. The execution layer then performs lossless reconfiguration of the current node's state based on the received configuration instructions. This is essentially a closed loop, transforming the original static configuration system into an adaptive system that can be adjusted according to SLO. The success of this system does not depend on the complexity of the algorithm, but rather on whether it can combine state, topology, and resources.

Furthermore, the paper argues that the marginal cost of event timing and out-of-order processing should not be underestimated. In order book flows, there will inevitably be timing misalignments in matching results, order book reconstruction, and external message orders. If watermarking is implemented too quickly, the cost of correcting late data will increase; if watermarking is implemented too slowly, it will lead to delays in downstream decision-making. Therefore, for order book scenarios, a

watermark implementation method layered by transaction object and by flow should be adopted, rather than using a globally uniform threshold. This can control the additional waiting time caused by out-of-order correction while maintaining the accuracy of temporal semantics.

5. Conclusion

This paper discusses the key performance bottlenecks and improvement methods of an order book event-driven stream processing architecture, starting from a low-latency financial scenario. Research shows that the essential characteristics of order book workloads are the superposition of three factors: event mutation, state density, and time sensitivity. The system bottlenecks are not primarily addressed by operator computational power, but rather are concentrated in state media access, hotspot partition skew, checkpoint blocking, and scaling lag. This paper establishes a unified analysis model of throughput, latency, and cost, demonstrating that optimization at any single point cannot lead to a sustained improvement in end-to-end performance.

Based on literature and empirical data from the past three years, we can conclude that: first, the state path length determines the upper limit of tail latency in the order book system; second, checkpoints and consistency protocols must match the load distribution pattern; third, predictive elasticity, hotspot isolation, and hybrid resource deployment are suitable for handling sudden market impacts; and fourth, the order book-oriented stream processing framework should adhere to the general principles of shortest critical path, closest hot state, controllable fault tolerance overhead, and proactive scaling. Future research could incorporate microstructure prediction signals, so that system scheduling resonates after market state is perceived, ultimately achieving adaptive stream processing in low-latency financial scenarios.

References

- [1] Dingyuan Liu. *Measuring the Sensitivity of Local Skill Structures to AI Substitution Risks Based on Occupational Task Decomposition*. *Socio-Economic Statistics Research* (2025), Vol. 6, Issue 2: 177-184
- [2] Zhen Zhong. *Big Data Engineering and Intelligent Analysis Framework for Compliance Investigation*. *Academic Journal of Computing & Information Science* (2025), Vol. 8, Issue 11: 107-115
- [3] Chen, X. (2024, November). *Cloud Storage User Behavior Analysis and Dynamic Replica Strategy Optimization Based on Improved RFM and Fuzzy Clustering*. In *International Conference on Cognitive based Information Processing and Applications* (pp. 425-434). Singapore: Springer Nature Singapore.
- [4] Wu, L. (2025, December). *Design and Application of Automatic Data Set Generation Tool Based on KLEE in Embedded Memory Management Performance Test Framework*. In *2025 IEEE 17th International Conference on Computational Intelligence and Communication Networks (CICN)* (pp. 1111-1117). IEEE.
- [5] Wu, L. (2025, December). *Design and Application of Automatic Data Set Generation Tool Based on KLEE in Embedded Memory Management Performance Test Framework*. In *2025 IEEE 17th International Conference on Computational Intelligence and Communication Networks (CICN)* (pp. 1111-1117). IEEE.

- [6] Truong, T. H. (2026). *Research on Risks and Countermeasures of Enterprise E-Commerce Transformation in the Cross-Border E-Commerce Environment*.
- [7] Qi, Y. (2025, October). *Research on Privacy Protection of AI Models in Big Data Using Differential Privacy Technology*. In *2025 2nd International Conference on Software, Systems and Information Technology (SSITCON)* (pp. 1-5). IEEE.
- [8] Hua, X. (2024, November). *Design and Implementation of a Game QoE Monitoring and Evaluation System Driven by Network Traffic Analysis*. In *International Conference on Cognitive based Information Processing and Applications* (pp. 149-161). Singapore: Springer Nature Singapore.
- [9] Huang, J. (2025, August). *Research on Multi-Model Fusion Machine Learning Demand Intelligent Forecasting System in Cloud Computing Environment*. In *2025 2nd International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS)* (pp. 1-7). IEEE.
- [10] Huang, J. (2025, September). *Performance Evaluation Index System and Engineering Best Practice of Production-Level Time Series Machine Learning System*. In *2025 International Conference on Intelligent Communication Networks and Computational Techniques (ICICNCT)* (pp. 01-07). IEEE.
- [11] Zhou, Y. (2024, November). *Construction of a Multi-factor Quantitative Stock Selection System for the New Energy Industry Based on Microservices Architecture and Machine Learning Components*. In *International Conference on Cognitive based Information Processing and Applications* (pp. 163-174). Singapore: Springer Nature Singapore.
- [12] Sun, Q. (2026). *Research on Lightweight Intelligent Dialogue Systems Based on Semantic Entity Enhanced Intention Recognition and Rule Retrieval Generation Hybrid Models*.
- [13] Wang, Y. (2026). *Research on the Application of Artificial Intelligence in Supply Chain Risk Early Warning*.
- [14] Liu, H. (2026). *Research on the Application of Causal Reasoning Method in Content Compliance Experimental Evaluation*.
- [15] Liu, H. (2026). *Research on Dynamic Price Prediction of E-commerce Based on Time Series Modeling*.
- [16] Yu, X. (2025). *Digital Transformation Empowers Growth Marketing with Marketing Data Analysis Integration and Real-Time Display Strategy*.
- [17] Yu, X. (2026). *Strategy Models and Practical Research of Growth Marketing under the Background of Digital Transformation*.
- [18] Yin, J. (2026). *Research on Financial Time Series Prediction and Multiscale Correlation Based on the Fusion of Network Big Data and Deep Learning*.
- [19] Hou, Y. (2026). *Research on BIOS and BMC Compatibility Optimization Methods for Cross-Generation Servers in Production Environments*.
- [20] Sun, Q. (2026). *Research on a Robotic Natural Language Intelligent Decision-Making Framework Based on Large Language Models, Thinking Chain Reasoning, and Multi-Agent Collaboration*.
- [21] Hou, Y. (2026). *Research on Server Performance Stability Assurance Mechanisms during Cross-Generation Computing Platform Upgrades*.
- [22] Han, X. (2026). *Research on Process Decision-Making Behavior under Incomplete Information Conditions in Automobile Manufacturing Systems*.
- [23] Han, X. (2026). *Research on Automotive Manufacturing Process Optimization Methods for Multi-Supplier Collaboration*.

- [24] Truong, T. H. (2025). *Research on the Application of Digital Healthcare Platforms in Chronic Disease Management*. *Advances in Computer and Communication*, 6(5).
- [25] Hong, Y. (2026). *Research on Warehouse Capacity Optimization Methods Based on Predictive Modeling*. *Engineering Advances*, 6(1).
- [26] Hong, Y. (2025). *Architecture Design and Performance Optimization of a Large-scale Online Simulation Platform for Business Decision-making*. *Advances in Computer and Communication*, 6(4).
- [27] Xuanrui Zhang. *Exploration of the Application of Big Data Technology in Financial Fraud Monitoring*. *Socio-Economic Statistics Research* (2026), Vol. 7, Issue 1: 19-26
- [28] Zhang, X. (2025, May). *Automobile Finance Credit Fraud Risk Early Warning System based on Louvain Algorithm and XGBoost Model*. In *2025 3rd International Conference on Data Science and Information System (ICDSIS)* (pp. 1-7). IEEE.
- [29] Wang, C. (2025). *Research on Market Evaluation Strategies for Financial Institutions Based on Big Data Analysis*.
- [30] Lu, Z. (2025). *AI-Driven Cross-Cloud Operations Language Standardisation and Knowledge Sharing System*. *European Journal of AI, Computing & Informatics*, 1(4), 43-50.
- [31] Lu, Z. (2025). *Design and Practice of AI Intelligent Mentor System for DevOps Education*. *European Journal of Education Science*, 1(3), 25-31.
- [32] Huijie Pan. *Discussion on Low-Latency Computing Strategies in Real-Time Hardware Generation*. *International Journal of Neural Network* (2025), Vol. 4, Issue 1: 57-64.
- [33] Linwei Wu. *Application and Development of Blockchain Technology in Financial Infrastructure Innovation*. *International Journal of Business Management and Economics and Trade* (2025), Vol. 6, Issue 1: 213-220.
- [34] Yilin Fu. *Data-driven Optimization of Capital Market Trading Strategies and Risk Management*. *International Journal of Business Management and Economics and Trade* (2025), Vol. 6, Issue 1: 221-228.
- [35] Qifeng Hu. *Optimization and Upgrade Path of Tax Management Software System Based on Cloud Platform*. *Socio-Economic Statistics Research* (2025), Vol. 6, Issue 2: 194-200.
- [36] Jin Li. *Performance Analysis of Efficient Microservice Architecture in the Financial Industry*. *Machine Learning Theory and Practice* (2026), Vol. 6, Issue 1: 1-9.
- [37] Yixian Jiang. *Performance Optimization and Improvement of Advertising Machine Learning Platform Based on Distributed Systems*. *International Journal of Big Data Intelligent Technology* (2026), Vol. 7, Issue 1: 9-17
- [38] Bukun Ren. *Multimodal Learning Method for Cross-Modal Data Alignment and Retrieval*. *International Journal of Multimedia Computing* (2026), Vol. 7, Issue 1: 1-8.
- [39] Shuang Yuan. *Research on Abnormal Detection and Transaction Risk Management Based on Machine Learning*. *International Journal of Social Sciences and Economic Management* (2026), Vol. 7, Issue 1: 10-18
- [40] Zhengle Wei. *Research on Innovative Design of Financial Derivatives and Market Risk Management Strategies*. *International Journal of Social Sciences and Economic Management* (2026), Vol. 7, Issue 1: 19-27
- [41] Yuhan Zhou. *Green Bonds and Sustainable Financing Models in Energy Finance*. *International Journal of Social Sciences and Economic Management* (2026), Vol. 7, Issue 1: 28-35
- [42] Yilin Fu. *Research on the Application of Innovative Financial Technologies in Capital Market Risk Management*. *Socio-Economic Statistics Research* (2026), Vol. 7, Issue 1: 1-9

- [43] Linwei Wu. *Data Visualization and Decision Support Analysis Based on Tableau. Socio-Economic Statistics Research (2026), Vol. 7, Issue 1: 10-18*
- [44] Xinran Tu. *Resource Allocation Optimization and Cost Saving Analysis Based on Data Mining. International Journal of Business Management and Economics and Trade (2026), Vol. 7, Issue 1: 1-9*
- [45] Wang, C. (2026). *Research on the Control of Uncertainty Risks in Investment Decision-making by Financial Modeling.*