Scholar Publishing Group
Socio-Economic Statistics Research
https://doi.org/10.38007/SESR.2026.070109
ISSN 2790-2722 Vol. 7, Issue 1: 77-86

Resource Scheduling and Elastic Scaling Optimization of Distributed Advertising Systems in Cloud-Native Environments

Linlin Sun

The Andrew and Erna Viterbi School of Engineering, University of Southern California, Los Angeles, California, 90089, United States

Keywords: Cloud-native advertising system; resource scheduling; elastic scaling; Kubernetes; SLO optimization

Abstract: In advertising platforms, operations such as real-time bidding, creative retrieval, user profile updates, and performance attribution occur frequently, leading to issues such as sudden traffic surges or cascading effects between links. Therefore, effective resource scheduling and elastic scaling for cloud-native governance are crucial. Based on recent research on cloud-native elastic computing and microservice scheduling, this paper constructs a collaborative optimization framework consisting of five parts: demand forecasting, layered scheduling, service-level scaling, node-level scaling, and overload protection, targeting the load characteristics of five stages in the advertising request chain: access, retrieval, ranking, billing, and logging. Methodologically, the paper proposes multi-metric demand forecasting, SLO constraints for replica control, resource fragmentation penalties, and cold start suppression, and provides validation using publicly available industry data and recent research findings. The research reveals that relying solely on CPU thresholds to control reactive scaling can result in peak-hour latency and off-peak idleness. A combined approach of forecasting first, feedback second, and layered scheduling better meets the millisecond-level service requirements of advertising systems. This provides a more engineering-feasible balance between stability, cost, and delivery for cloud-native advertising platforms.

1. Introduction

Distributed advertising systems are typical high-concurrency, latency-sensitive, and cross-chain microservice systems. A single advertising request typically completes traffic access, candidate retrieval, feature assembly, ranking and scoring, budget verification, billing recording, and log transmission within tens of milliseconds; resource congestion at any stage will increase the overall latency. Cloud-native architecture divides all capabilities into independently deployable service units, thereby improving delivery efficiency and observability, but it also introduces new resource configuration challenges such as replica governance, node fragmentation, cold start, cross-service dependencies, and asynchronous message backlog.

In the past three years, the academic and industrial communities have continuously explored issues such as predictive scaling, reinforcement learning scaling, microservice overload control, workflow scheduling, and cloud-native availability migration. Yuan and Liao [1] proposed the idea of using time series for predictive scaling, [2] studied the active scaling mechanism from the perspective of machine learning prediction, [3] systematically summarized the availability and scalability problems existing in the migration process of containerized applications, [4] again explained from the perspective of failure prediction that scaling decision errors are not accidental phenomena, [5] and finally linked entry-level overload control and SLO-oriented microservice collaborative governance. Resource scheduling and elastic scaling have evolved from the original "addition and reduction of a single service replica" to the current "multi-level, strong constraint, and strong feedback" system optimization problem.

However, advertising services have three distinct characteristics that differentiate them from general web services. First, traffic peaks are driven by campaigns and influenced by day-night cycles; prediction errors directly lead to lost bids or wasted resources. Second, ranking, vector retrieval, profile updates, and log calculations utilize CPU, memory, network, and I/O differently, making it impossible to represent the true bottleneck with a single threshold. Third, budget control, frequency control, and risk control backtracking are interconnected; premature scale-down in one service can starve dependent downstream services. Therefore, research on resource scheduling and elastic scaling in cloud-native advertising systems has both theoretical and practical engineering significance.

2. Related Work

From a technical perspective, current research on cloud-native elasticity can be roughly divided into three categories. The first category is threshold-triggered reactive mechanisms, such as HPA, VPA, and Cluster Autoscaler. These mechanisms have the advantage of being simple and having a low deployment threshold, but they lack foresight regarding sudden traffic surges. [6] Fourati et al. [6] argue that the default threshold-triggered method of Kubernetes will cause a lag in resource allocation when the load changes rapidly. The second category is predictive mechanisms, which use time series, deep learning, or hybrid predictive models to predict changes in demand in advance and complete the preparation of replicas and nodes before the peak arrives. Both Yuan and Liao [1] and Shim et al. [2] demonstrate that predictive approaches can address the cold start and queuing phenomena that occur when only reactive scaling is used. The third category is policy learning and joint optimization mechanisms, which incorporate service dependencies, costs, SLOs, and node constraints into the decision-making process. Santos et al. [7] and Tari et al. [8] show that reinforcement learning and multi-layer control are more suitable for complex microservice environments, but training costs, interpretability, and online convergence still need to be taken seriously.

According to the object of resource scheduling, the focus of research has expanded from the original single Pod or single Deployment to multiple layers such as services, nodes, clusters, and businesses. Santos et al. [7] address the problem of dependency propagation in complex containerized applications, demonstrating that simply scaling local bottleneck services cannot guarantee end-to-end performance improvements; Liu et al. [9] propose a resource discovery and dynamic allocation mechanism for workflow scheduling, arguing that proactive scheduling is critical for high-concurrency scenarios. Jeong [10] provide a comprehensive review of autoscaling in cloud-native environments, recommending that future research focus on cross-layer collaboration, energy consumption constraints, and business semantic awareness. The scheduling goal of the advertising platform is no longer "to make CPU utilization higher", but to schedule based on

meeting the requirements of throughput, latency, stability, cost, release, and gray-scale security.

Industry statistics show that cloud-native operations have shifted from automated deployment to refined resource management. According to the 2024 CNCF annual survey, 38% of organizations implemented 80%-100% automated deployments, an increase from 28% in 2023. The average automated deployment rate at the organizational level also increased from 56.5% to 59.2%. This indicates that scaling strategies on advertising platforms must be integrated with continuous delivery systems and cannot be isolated operations. Datadog's report on container and serverless states shows that over 64% of Kubernetes organizations have used HPA, but only about 20% of HPA deployments use custom metrics for scaling. Datadog's cloud cost research also mentions that 83% of container costs are caused by idle resources. This demonstrates that simply "enabling autoscaling" is insufficient for resource optimization; the truly effective methods are metric selection, strategy layering, and closed-loop feedback capabilities.

Table 1. Summary of relevant English research in the past three years

Ref.	Focus	Method	Contribution	Limitation
[1]	Elastic scaling	Time-series forecast	Proactive pod scaling	Prediction drift risk
[4]	Autoscaling failure	Failure prediction	Expose scaling fault patterns	Need rich telemetry
[5]	Microservice overload	Top-down control	Improve goodput under overload	Entry-focused control
[7]	Complex app scaling	Reinforcement learning	Model service dependency	Training overhead
[9]	Workflow allocation	Dynamic resource allocation	Better concurrency handling	Workflow-specific setting

As shown in Table 1, in recent years, the understanding of scaling issues has shifted from simply increasing or decreasing replicas to a comprehensive optimization approach encompassing prediction, failure identification, dependency awareness, and overload management. This approach improves capacity reserves before peak periods, while also addressing service dependencies, entry point throttling, and multi-layered resource constraints, which are increasingly being studied by mainstream research. For advertising systems, these results directly suggest that only by incorporating traffic prediction, link dependencies, and SLO constraints into the decision-making process can we avoid situations where local improvements mask global performance degradation.

As shown in Figure 1, cloud-native organizations are becoming increasingly sophisticated in terms of deployment automation. The proportion of deployments with 80% to 100% automation in 2024 was significantly higher than in 2023, while the proportion of deployments with low to medium automation also decreased. This indicates that resource scheduling and elastic scaling strategies cannot be planned independently of the continuous delivery system. For advertising systems, the faster the deployment speed, the more frequently service instances, configuration parameters, and scaling rules change. Without rollback-capable strategy orchestration and canary testing mechanisms, deployment agility can become operational instability.

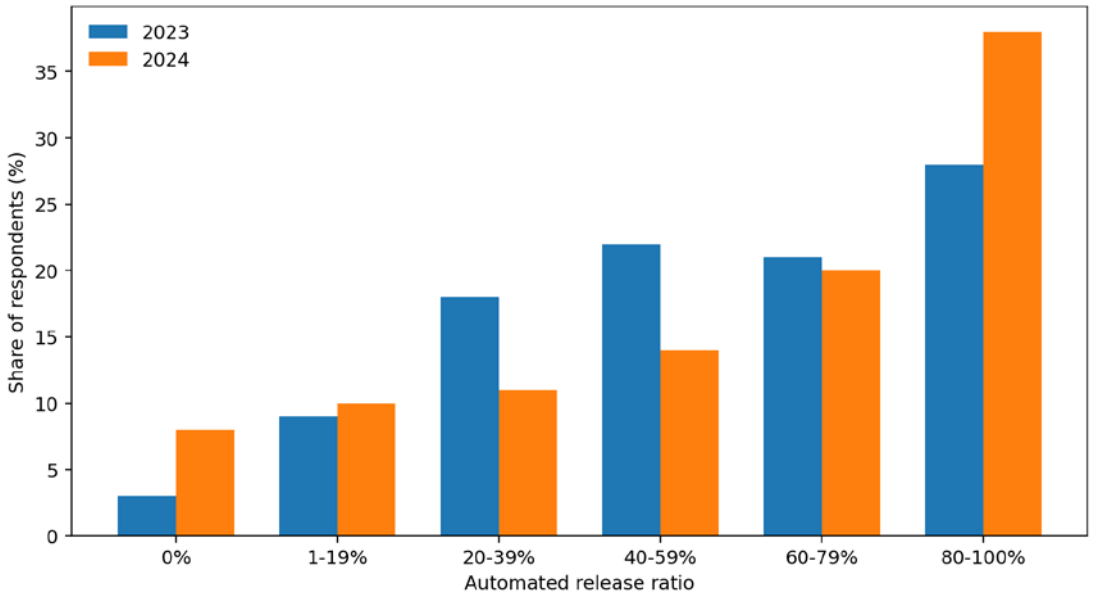

Figure 1. Distribution of automated release rates (redrawn based on CNCF 2024 Annual Survey data)

3. Problem Analysis

Based on the characteristics of advertising business and existing research findings, cloud-native distributed advertising systems currently face the following four structural problems in resource scheduling and elastic scaling.

First, there's the conflict between load forecasting and scaling timing. Advertising traffic is influenced by daily life cycles, as well as holidays, marketing campaigns, budget release methods, platform traffic fluctuations, and other factors. If scaling is delayed too long, requests will queue during recall, ranking, or billing stages, leading to timeouts and failed bids. If scaling is delayed too long, a large number of underutilized replicas and idle nodes will result. For retrieval/ranking services with model loading and cache warm-up, the actual cost of a cold start is much higher than the number of replicas.

Second, the scaling signals are singular and lack business-related data. Currently, many production clusters still use CPU or memory utilization-based scaling, but bottlenecks in advertising systems can manifest in various aspects such as QPS spikes, queue length, P95 latency, remote call failure rate of feature services, Kafka backlog, or CTR/CVR model inference time. If generic metrics continue to be used, a false situation will arise where "monitoring shows sufficient resources, but requests still time out."

Third, there is a mismatch between service-level optimization and cluster-level optimization. On advertising platforms, services such as search, ranking, frequency control, and billing are often handled by different teams, with each replica having its own number and priority, and varying resource request values. As a result, some services receive more resources, but the cluster layer experiences issues such as node fragmentation, frequent contention, and increased cross-availability zone traffic, thereby increasing cloud costs and reducing the overall system stability.

Fourth, there is a lack of closed-loop management of strategies. Many systems are observable, but lack a complete closed loop of prediction, decision-making, execution, verification, and correction. After scaling operations are completed, without reviewing SLO recovery time, scaling benefits, rollback safety windows, false alarm rates, and resource waste rates, the strategy will not

continuously evolve. This is why many organizations, despite using HPA, still experience significant idle costs and capacity mismatches.

4. Proposed Optimization Framework

To address the above issues, this paper presents a collaborative optimization framework for predictive awareness, hierarchical scheduling, elastic scaling, and overload protection in cloud-native advertising systems. This framework does not optimize a single metric in isolation, but instead establishes a verifiable, rollback-capable, and iteratively refinable balance among SLO, cost, stability, and deployment agility.

4.1 Objective Function Design

In advertising systems, resource optimization must consider factors such as instance cost, node cost, latency penalty, and scaling/jitter cost. Therefore, the single-cycle optimization objective can be expressed as:

$$\min J\,t = \alpha C\,pod,t + \beta C\,node,t + \gamma P\,SLO,t + \delta A\,t \tag{1}$$

In Equation (1), C_pod,t represents the resource consumption on the service replica layer, C_node,t represents the capacity occupancy and fragmentation cost of the node layer, P_SLO,t represents the penalty caused by P95 latency, timeout rate, and data loss, and A_t represents the jitter cost caused by frequent scaling. Weighting these four types of costs can prevent the system from focusing solely on high utilization while ignoring latency risks.

4.2 Multi-indicator demand forecasting and capacity pre-planning

Based on the rhythmic and sudden nature of advertising traffic, traffic, queue, and business event signals can be input together into the demand forecasting module to obtain the demand forecast results for the next time window.

$$D\,t+1 = (\hat{q}\,t+1\,,\,\hat{l}\,t+1\,,\,\hat{m}\,t+1\,) = F(q\,t\,,\,l\,t\,,\,m\,t\,,\,e\,t\,) \tag{2}$$

In equation (2), $\hat{q}_{t+1}$ represents the predicted request volume, $\hat{l}_{t+1}$ refers to the predicted queue length, and $\hat{m}_{t+1}$ is the inference burden of the model or the pressure of the feature service. This design draws on the main ideas of predictive scaling, completing the preheating of replicas, loading of cache, and establishment of connections before the actual peak arrives, thereby reducing the latency during cold start.

4.3 Service-level replica scaling

For stateless or weakly stateful services such as recall, sorting, and billing, the required number of replicas can be estimated by using the arrival rate per unit time and the processing capacity of a single replica.

$$R\,t = \lceil\, \lambda\,t\,/\,(\,\mu \cdot u\,^{*}\,)\,\rceil \tag{3}$$

In equation (3), λ_t represents the request arrival rate within the current or prediction window, μ is the stable service rate of a single replica, and u^* is the upper bound of the expected utilization rate. The u^* of the advertising system should not be set too high, otherwise the P95 latency will rise rapidly at the peak edge; generally, a safety margin should be left.

4.4 Comprehensive Scalable Scoring Triggered by Custom Metrics

To overcome the problem of distortion caused by a single CPU metric, a comprehensive scaling score S_t can be created.

$$S\ t = w\ 1\ LAT95\ t + w\ 2\ QDEPTH\ t + w\ 3\ FAIL\ t \quad (4)$$

In equation (4), LAT95_t represents the P95 latency, QDEPTH_t represents the message or request backlog depth, and FAIL_t represents the failure rate of critical downstream calls. The system will only perform capacity expansion when the comprehensive score continuously exceeds the threshold and the minimum observation window continues to exist; in the low load range, the pullback confirmation time should be longer to prevent back-and-forth swings caused by fluctuations in advertising.

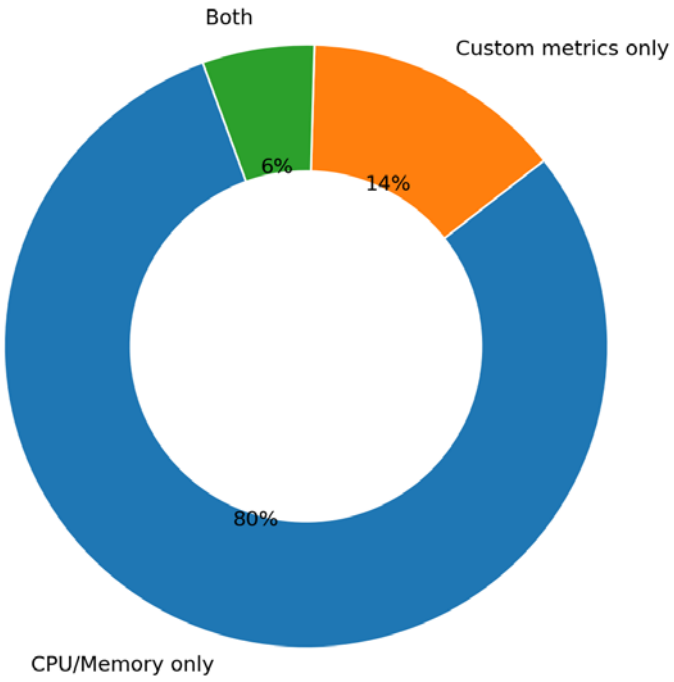

Figure 2. HPA scaling metric usage structure (redrawn based on Datadog State of Containers and Serverless data)

In Figure 2, among deployments that have already implemented HPA, most still rely on standard metrics such as CPU or memory to measure system performance, with a relatively low proportion actually using business-defined metrics such as request rate and queue length. This structure typically reflects a real contradiction in advertising systems: while the platform possesses autoscaling capabilities, it hasn't developed business-semantic-driven autoscaling. Therefore, if advertising platforms want to improve resource response speed during peak bidding periods, they must incorporate business characteristics such as latency, queuing, failure rate, and budget timing into their scaling scores.

4.5 Node-level scheduling and resource fragmentation penalty

In a cluster, it's unacceptable for node fragmentation to prevent theoretically scalable nodes from actually scaling. Node packing quality can be described using a resource fragmentation penalty term.

$$F\ t = \Sigma\ i=1\ N\ (|cpu\ i - \bar{c}pu| + |mem\ i - \bar{m}em|) \quad (5)$$

In equation (5), the larger F_t is, the more unbalanced the resource utilization among nodes becomes, and the more likely it is that one type of resource will be exhausted while another type of resource will be idle. When expanding the advertising platform, first select a scheduling scheme that can improve the CPU and memory packing quality, and reserve bandwidth, cache, and instance slots for high-priority services.

4.6 Hierarchical Collaborative Governance Mechanism

The first layer is called the business forecasting layer. This layer makes demand predictions based on the event calendar, historical QPS, budget release curve, material update speed, and media traffic conditions, thus providing a basis for preparing for capacity expansion before peak periods.

The second layer is the service decision layer. Different SLO weights and scaling windows are applied to access, recall, ranking, billing, and log services. Access and ranking services prioritize latency, while log and offline aggregation services can focus more on cost efficiency.

The third layer is the cluster scheduling layer. This layer mainly completes node pool capacity planning, cross-availability zone balancing, preemption strategies, and affinity and anti-affinity control to ensure that high-priority services have sufficient stable operating space and reduce fragmentation and cross-zone traffic.

The fourth layer is the governance feedback layer. By comparing indicators such as SLO recovery time before and after scaling, idle resource ratio, scaling-up hit rate, and scaling-down security, the thresholds, weights, and prediction models are periodically adjusted to achieve continuous strategy updates.

Table 2 Summary of publicly available metrics related to cloud-native operations

Metric	Value	Period	Source
Automated releases avg.	56.5%	2023	CNCF
Automated releases avg.	59.2%	2024	CNCF
80-100% automated releases	28%	2023	CNCF
80-100% automated releases	38%	2024	CNCF
HPA adoption	64%+	2025	Datadog
HPA with custom metrics	20%	2025	Datadog
Idle-related container cost	83%	2024	Datadog

Table 2 summarizes two main trends in the publicly available metrics. First, the automation level of deployments in cloud-native organizations is continuously increasing, indicating that the platform's operating environment is becoming increasingly dynamic, and resource strategies must be able to adapt to the deployment speed. Second, although HPA usage is already high, the application rate of custom metrics is low, and the idle cost ratio is high, indicating that the problem of "insufficient strategy depth" is more important than whether to use autoscaling. Advertising system resource governance should shift from enabling basic capabilities to improving metric quality and decision-making accuracy.

Table 3. Layered scheduling and scaling mapping for the ad request chain

Stage	Primary signal	Scaling mode	Scheduling priority	Expected effect
Ingress	QPS, LAT95	Predictive + reactive	Highest	Shield burst traffic
Retrieval	CPU, cache miss	Warm-up replicas	High	Cut cold-start delay
Ranking	GPU/CPU, latency	Queue-aware scaling	Highest	Protect bidding SLA
Billing	IO, fail rate	Safe reactive scaling	Medium	Guarantee correctness
Logging	Lag, throughput	Batch-aware scaling	Low	Lower idle cost

Table 3 outlines differentiated governance strategies for the ad request chain. Access, recall, and ranking services are directly related to bidding latency and should employ predictive, queue-aware scaling methods. Billing services must achieve a stable balance between performance and accuracy. Logs and offline services can utilize batch processing and peak/valley smoothing techniques to reduce idle time during off-peak periods. This table indicates that there is no single, uniform scaling threshold across the entire ad chain; phased modeling is the key to ensuring effective engineering.

5. Conclusion

Cloud-native environments provide distributed advertising systems with the foundation for high elasticity, observability, and rapid delivery. However, they also transform resource scheduling and elastic scaling from single-point operation and maintenance issues into systems engineering problems. Based on recent literature and publicly available industry data, this paper analyzes the existing problems in resource governance, such as predictive lag, single signal, local optima, and insufficient closed-loop, starting from the business characteristics of advertising platforms. Finally, it proposes a collaborative optimization framework based on the advertising request chain.

The research results indicate that an optimal scaling strategy for an advertising system should possess three main capabilities: first, the ability to pre-plan capacity based on multiple predictable metrics; second, the ability to incorporate business semantics such as latency, queue size, and failure rate into custom decision-making metrics; and third, the ability to simultaneously limit resource allocation at both the service and cluster layers, thereby avoiding fragmentation and fluctuations. A layered collaborative strategy is more suitable than reactive solutions that rely solely on CPU thresholds for the demands of advertising scenarios characterized by "millisecond-level response, distinct peaks and troughs, complex processes, and cost sensitivity."

Future research can continue in three areas: first, by incorporating advertising business metrics such as bidding revenue, exposure completion rate, and budget consumption rate to establish a more revenue-driven scheduling model; second, by utilizing large-scale model inference and vector retrieval load conditions to expand the elastic control mechanism of GPU homogeneous and heterogeneous computing power; and third, by using canary releases and verification methods in multi-cluster federated environments to test the robustness of the strategy, thereby providing large-scale advertising platforms with more generalized cloud-native resource improvement methods.

References

[1] *Yuan H, Liao S. A Time Series-Based Approach to Elastic Kubernetes Scaling[J]. Electronics, 2024, 13(2): 285. DOI: 10.3390/electronics13020285.*

[2] *Shim S, Dhokariya A, Doshi D, et al. Predictive Auto-scaler for Kubernetes Cloud[C]//2023 IEEE International Systems Conference (SysCon). 2023. Wang, Z. (2026). Exploration of the Application of Resource Circular Economy in the Agricultural Industry Chain.*

[3] *Yin, J. (2026, March). An Efficient Iterative Algorithm for Calibrating Agency MBS Estimation Parameters Based on Bayesian Optimization, Implemented in Python. In 2026 IEEE Madhya Pradesh Section Conference (MPCON) (pp. 1462-1467). IEEE.*

[4] *Zheng, H. (2026, April). Research on Cloud-Edge Collaborative Elastic Computing and Cost Optimization for High-Concurrency Scenarios. In 2026 IEEE 15th International Conference on Communication Systems and Network Technologies (CSNT) (pp. 1489-1496). IEEE.*

[5] *Liu, H. (2025, December). Mlops Model Deployment System for Multi-Cloud Environments and Improvement of Commercial AI Service Availability. In 2025 IEEE International Conference on Communication Networks and Computing (CNC) (pp. 1047-1052). IEEE.*

[6] *Hong, Y. (2025, December). Chain Demand Mutation Identification and Emergency Response Decision-Making Integrated With Social Media Sentiment Analysis. In 2025 IEEE International Conference on Communication Networks and Computing (CNC) (pp. 425-432). IEEE.*

[7] *Chen, M. (2026, March). Design of a Privacy-Oriented AI Compliance Hook System Based on Static Code Analysis. In 2026 IEEE Madhya Pradesh Section Conference (MPCON) (pp. 648-654). IEEE.*

[8] *Lyu, N. (2026, April). Toward Robust AI Agents: A Closed-Loop Task Planning–Execution–Feedback Framework for Open Scenarios. In 2026 IEEE 15th International Conference on Communication Systems and Network Technologies (CSNT) (pp. 1182-1188). IEEE.*

[9] *Wang, Y. (2025, December). Construction of an Intelligent Recommendation System for Personalized Training Plans for Sports Rehabilitation Patients Based on Graph Neural Networks. In 2025 IEEE 1st International Conference on Recent Trends in Computing and Smart Mobility (RCSM) (pp. 1-6). IEEE.*

[10] *Wang, Y. (2025, July). Research on Federated Learning Algorithm Design and Privacy Protection Mechanism of Sports Rehabilitation Data Based on Multimodal Fusion. In International Conference on Frontier Computing (pp. 563-574). Singapore: Springer Nature Singapore.*

[11] *Zhang, Q. (2025, July). Research on Improving the Effectiveness of Programmatic Advertising Using Multi-armed Bandit Algorithms. In International Conference on Frontier Computing (pp. 543-552). Singapore: Springer Nature Singapore.*

[12] *Ding, J. (2025). Exploration of Process Improvement in Automotive Manufacturing Based on Intelligent Production. International Journal of Engineering Advances, 2(2), 17-23.*

[13] *Zhang, K. (2025, December). Research on Cross-Selling Precision Marketing Strategy Based On Xgboost Model and SHAP Explanatory Framework. In 2025 IEEE 1st International Conference on Recent Trends in Computing and Smart Mobility (RCSM) (pp. 1-7). IEEE.*

[14] *Wu, W. (2025, June). Construction and optimization of intelligent gateway software management platform based on jenkins cluster management under cloud edge integration architecture in industrial internet of things. In International Conference on 6G Communications Networking and Signal Processing (pp. 633-645). Singapore: Springer Nature Singapore.*

[15] Wu, Y. (2025, October). Multi-Level Belief Rule Base Modeling Architecture and Intelligent Optimization Technology for Decision Support Systems. In 2025 2nd International Conference on Software, Systems and Information Technology (SSITCON) (pp. 1-8). IEEE.

[16] Liu, X., & Yang, D. (2025, March). LLM Data Strategy: Improving Data Availability and Efficiency. In Doctoral Symposium on Computational Intelligence (pp. 425-437). Singapore: Springer Nature Singapore.

[17] 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.

[18] Zhang, C., Han, J., Zou, Y., Dong, K., Li, Y., Ding, J., & Han, X. (2024, April). Detecting the anomalies in LiDAR pointcloud. In WCX SAE World Congress Experience. SAE Technical Paper.

[19] Ding, J. (2025). Research On CODP Localization Decision Model Of Automotive Supply Chain Based On Delayed Manufacturing Strategy. arXiv preprint arXiv:2511.05899.

[20] Hou, Y. (2026). Exploration of Data Center Cost Optimization Pathways under Multi-generation CPU and GPU Collaborative Architectures. Engineering Advances, 6(1).

[21] Yanchun Wang. (2025) Research on Enhancing ERP System Efficiency through AI in Cross-border Supply Chain Environments. Advances in Computer and Communication, 6(5), 268-273.