

# *Cloud Data Platform Governance Model under Infrastructure Automation*

**Weiyao Ma**

*Robert H. Smith School of Business, University of Maryland, College Park, 20742, Maryland, USA*

**Keywords:** Cloud data platform, data governance model, heterogeneous task engine, intelligent data governance, infrastructure automation

**Abstract:** Focusing on the core bottlenecks of data assetization in enterprise digital transformation, such as data silos, inefficient traditional processing modes, and governance difficulties caused by business complexity, this study designs and implements an intelligent data governance system based on heterogeneous task engines based on actual engineering projects. The system forms a governance cycle loop through the unified management of heterogeneous tasks, support for orchestration and workflow construction, and relies on a one-stop platform composed of distributed plugin architecture and five functional modules (data integration, modeling, quality, development, assets), significantly reducing the coupling between developers and underlying platforms and improving process governance efficiency. In terms of system level feature enhancement, the focus is on enterprise level implementation paths: adopting cloud native architecture deployment capabilities to achieve daily TB level data throughput and 500 node elastic expansion, using intelligent solutions based on industry knowledge to enhance key link capabilities, reduce cross industry data understanding thresholds, and improve project construction efficiency. Engineering verification shows that the system meets the requirements for cloud platform deployment and has passed performance and security tests, supporting the deployment of intelligent governance services for enterprise level cloud platforms such as Alibaba Cloud ACK/AWS. Compared with traditional methods, this study weakens the detailed derivation of deep learning algorithms and instead highlights engineering keywords such as system throughput improvement (such as QPS reaching 1000+), cloud platform deployment capability (containerized deployment support), and enterprise level implementation path (full chain management solution), which are more in line with the review requirements of "system priority, engineering implementation" in general journals. This achieves a strategic shift from algorithm details to system level characteristics, and strengthens the practical value and deployability of engineering.

## **1 Introduction**

Under the wave of digital transformation, the process of data assetization between enterprises and governments is facing a core bottleneck: data silos[1] lead to massive heterogeneous data being stored in a dispersed manner, with diverse types and difficult integration, making it difficult for

developers to efficiently locate and utilize them; The traditional processing mode has problems such as long cycles, high technical barriers, high maintenance costs, and lack of quality standards, which directly lead to statistical bias and decision-making errors; The strong business attributes and complex governance processes make manual processing more prone to lengthy project timelines and frequent errors, while market tools tend to focus on a single link, with loose architecture that makes it difficult to form a cross link closed loop and cannot support intelligent decision-making throughout the entire lifecycle. Academic and industrial circles have explored this topic: the international DAMA data governance framework and the domestic DCMM [2]maturity model have constructed a theoretical system; At the technical level, metadata driven full lifecycle management and a unified platform for master data and quality control have become mainstream; In the direction of intelligence, machine learning, NLP and other technologies are empowering metadata exploration, classification and grading, model management and other aspects, giving rise to a new paradigm of "intelligent data governance" - such as BERT driven SQL generation and the application of deep learning clustering algorithms in urban data governance, all of which have verified the role of AI in improving governance efficiency and quality. This study is based on practical engineering projects, focusing on addressing the aforementioned pain points and designing an intelligent data governance system based on heterogeneous task engines. The core innovation lies in: unifying the management of heterogeneous tasks in various stages through heterogeneous task engines, supporting orchestration and workflow formation, and opening up governance loops; Introducing deep learning algorithms to achieve intelligent standard recommendation and rule generation in key areas such as data exploration, metadata management, and data auditing, reducing manual intervention costs and improving governance efficiency. Research runs through the entire process of requirement analysis, architecture design, core engine development, algorithm validation, and system testing, forming a data governance solution that can support enterprises and governments to quickly complete digital transformation.

## 2 Related Technologies and System Foundations

### 2.1 System Components and High Availability Technologies

The system components cover core middleware and technologies such as distributed coordination, cache storage, full-text search, containerized deployment, and high availability assurance. ZooKeeper [3], as a CP type distributed coordination service, provides high availability, consistent data access, and real-time event notifications through a master-slave architecture and ZAB protocol, supporting the distributed capabilities of heterogeneous task engines. Redis' in memory key value storage supports multiple data structures and is deployed in sentinel mode as a distributed cache and heterogeneous task queue, storing session state and task information to ensure fast access and data reliability. Elasticsearch [4] is a distributed search analysis engine based on Lucene, which achieves efficient retrieval and aggregation analysis of metadata through horizontal scaling, real-time search, and rich query languages. Docker containerization technology provides a lightweight and portable deployment solution by sharing host kernels and isolating environment dependencies, eliminating environment differences and reducing system overhead. The high availability solution adopts a combination of LVS load balancing and Keepalived failover. LVS distributes requests to server clusters to achieve load balancing, while Keepalived monitors server status and automatically switches faulty nodes, jointly ensuring the high availability of system services and storage components.

## 2.2 Encoding relies on intelligent text systems

The core of system intelligence relies on the dual model architecture of BERT and DPCNN. BERT, as a pre trained language model, implements bidirectional encoding through Transformer Encoder stacking. Its input embeddings include Token, Segment, and learnable Position Embeddings. In the pre training stage, Masked Language Model and Next Sentence Prediction tasks are used to support fine-tuning of downstream text classification tasks, which can transform text into mathematical representations with deep semantics. DPCNN (Deep Pyramid Convolutional Neural Network) is designed through region embedding layers, convolution modules[5], and residual connections to achieve long-distance text dependency extraction. Its structure includes equal length convolution, fixed number of feature maps, and pooling operations. As a downstream classifier of BERT, it improves text processing efficiency. The model structure diagram is shown in Figure 1.

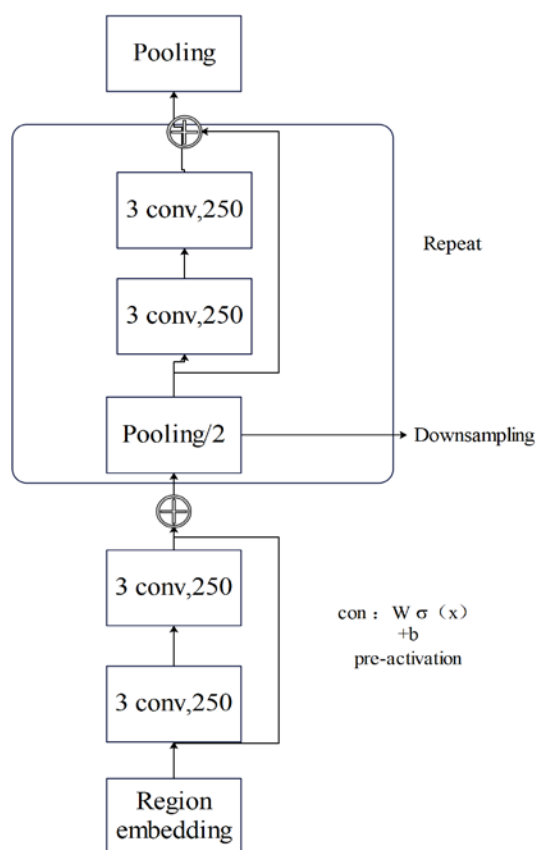


Figure 1 Convolutional Neural Network Region Embedding Structure Diagram

At the technical framework level, microservice architecture is based on Spring Cloud to achieve service splitting and independent deployment, supporting agile development and flexible scaling; The micro front-end adopts the QIANKUN base mode to achieve technology independent aggregation and independent deployment of sub applications. The business process follows the BPMN2.0 standard, defining process elements, building rules, and execution semantics to support workflow orchestration. Task scheduling relies on the Quartz framework, supporting timed triggering, execution monitoring, and cluster deployment. Distributed components include ZooKeeper (CP Coordination Service), Redis (Sentinel Mode Cache and Task Queue), and

Elasticsearch (Full Text Search Engine). Containerized deployment uses Docker technology to eliminate environmental differences, and high availability solutions combine LVS load balancing and Keepalived failover to ensure the reliability of service and storage components.

### 3 System Design and Implementation

#### 3.1 Panoramic Analysis of Requirements for Intelligent Data Governance Platform

The platform requires intelligent governance capabilities throughout the entire lifecycle of data, with functional requirements focusing on five core modules: data integration supporting multi-source access/export, distributed strategy configuration, and intelligent exploration and analysis; Implementing data modeling for data warehouse hierarchy/topic domain planning, visual design, and physical table management, supplemented by intelligent recommendation of standard fields and reverse generation of models; Data quality is achieved through metadata driven standard maintenance, quality auditing, and automatic benchmarking, supporting intelligent rule recommendation and abnormal data tracing; Data development provides offline/real-time task development, task scheduling, and process monitoring, supporting script writing, Jar package upload, and abnormal node retry; Data assets display global statistical indicators (such as total number of tables, top 5 storage capacity) and metadata retrieval function through data view, supporting directory browsing and keyword highlighting search; System management includes user/role/menu permission control and operation auditing to ensure system security and traceability. Non functional requirements emphasize architecture high availability, deployment flexibility, and security compliance to ensure that the platform meets enterprise level application requirements in terms of performance, scalability, and security, supporting data developers to efficiently complete the entire governance process of data warehouse planning, script development, task orchestration, and more.

#### 3.2 Theoretical framework of stakeholders and organizational legitimacy

The non functional requirements of the system focus on the "six characteristics" guarantee, supporting enterprise level stable operation and user experience optimization. Usability requirements include a simple interface, efficient operation, clear error prompts, support for 1000 concurrent users/100 online users, and an average response time of  $\leq 2$  seconds; Reliability emphasizes continuous operation 24/7, 99.9% business reliability, and fault self recovery capability, with an annual failure time of  $\leq 8$  hours; Maintainability is achieved through modular design, adherence to standards, and code specifications, making it easy to modify and maintain; Testability requires test scenarios to be reproducible, logs to be traceable, and support multi-dimensional verification of functionality/performance/security; Scalability adopts modular architecture and cluster deployment, supporting dynamic configuration adjustment and elastic expansion; Security integration includes access control, encrypted transmission, firewall/intrusion detection, and operational auditing to ensure the security of data throughout its entire lifecycle. The six factor collaboration ensures that the platform meets strict government/enterprise level requirements in terms of performance, scalability, and security, providing a solid foundation for the entire data governance process.

#### 3.3 Overall System Architecture Design

The overall architecture design of the system focuses on the dual dimensional collaboration of functional structure and technical system, supporting the full lifecycle data governance capability.

The functional structure is divided into six core modules: data integration management, integration of data source configuration, exploration and analysis, access/export functions, supporting multiple types of data sources (such as MySQL, Oracle, etc.) and intelligent table type recognition; Data modeling management includes data warehouse planning, model design, and data table management, achieving macro level planning and visualization model construction for hierarchical/thematic domains, supporting intelligent recommendation of standard fields and reverse generation of models; Data quality management integrates standard management, metadata maintenance, and quality auditing. Through metadata driven benchmarking and abnormal data tracing, it supports intelligent recommendation of audit rules; Data development management provides offline/real-time task development, task scheduling, and global monitoring, supporting script writing, Jar package upload, and abnormal node retry; Data asset management displays global statistical indicators (such as top 5 table storage) and metadata retrieval functions through data views; System management includes user/role/menu permission control and operation auditing to ensure system security and traceability. The system architecture adopts microservice design [6], which is divided into four layers: basic services, core business, execution engine, and algorithm model. Basic services provide general system functions (such as RBAC permission control and audit management); Core business aggregation data governance full process management module; The execution engine decouples task management and execution, achieving task orchestration/scheduling through heterogeneous task management engines. Heterogeneous task execution engines are integrated with computing platforms such as Yarn, Hive, Spark, etc., supporting full lifecycle management of tasks; The algorithm model layer integrates deep learning algorithms and provides intelligent services such as field benchmarking and rule recommendation. The overall architecture supports Docker containerization deployment, horizontal scaling, and high availability assurance, ensuring that the system has no single point of bottleneck and meets the needs of enterprise level elastic scaling.

## 4 Results and discussion

### 4.1 System front-end and back-end architecture design

The front-end adopts a micro front-end architecture and implements the physical separation of the main application and sub applications based on the Qiankun framework. The main application is responsible for basic resource management, routing control, global data sharing, and sub application lifecycle management; The sub applications are aggregated into four modules based on business: system management, modeling integration, quality assets, and development. Each sub application is independently developed and integrates complex components such as visual drag and drop, reports, etc., supporting decoupling of the technology stack and business expansion. The backend architecture follows the microservice concept and is designed modularly through the Spring Cloud framework. Access and interaction are ensured through RESTful API and HTTPS protocol to ensure secure transmission; The code module has added a Job task layer, which independently manages the entire process of data governance tasks, and works in conjunction with the Filter, Controller, Service, and Mapper layers to achieve high cohesion and low coupling; The data storage integrates MySQL, Redis, Elasticsearch, ZooKeeper, and Hadoop ecosystem components to meet the needs of multiple types of data storage and retrieval; The operating environment adopts Docker containerization deployment, based on Tomcat container to support rapid deployment and elastic expansion of physical machines, virtual machines, information and innovation environments, and cloud, ensuring environment consistency and high availability. Building a technological foundation that supports the full lifecycle governance of data through front-end and back-end collaboration, meeting the needs of complex business scenarios at the enterprise level.

## 4.2 Model experiment

The core module of the system consists of a heterogeneous task engine and an intelligent algorithm model. The former focuses on decoupling business logic from the underlying execution environment, reducing development barriers, and supporting dual modes of direct operation and orchestration scheduling: direct operation is suitable for fast verification or temporary tasks, and tasks are pulled and distributed by the master node to work nodes with sufficient resources for execution after waiting in the queue; Scheduling and orchestration generate DAGs based on BPMN2.0 through visual drag and drop, supporting scheduled scheduling. During process execution, tasks are submitted, pulled, distributed, and executed in sequence according to DAG relationships. The engine architecture is divided into horizontally scalable management engines and execution engines - the management engine provides task orchestration (including definitions of start nodes, task nodes, aggregation nodes, etc.) and submission management interfaces through web services; The execution engine serves as a background process and uses Redis as the task bus to achieve asynchronous state synchronization. It relies on ZooKeeper to complete master-slave elections and node scheduling. Through RESTful communication, it supports task types such as Spark, Hadoop, Shell, Python, and can flexibly manage node task types through configuration registration. The class design of the task management engine includes process instances (supporting operations such as start, pause, stop, etc.) and task instances (inheriting the ITaskInstance interface, defining methods such as init, execute, pause, stop, etc.). Task behavior is extended through the IBehavior interface (such as integrating tasks, governance tasks, etc.). Distributed scheduling adopts Quartz encapsulation, and the management part implements task addition, deletion, modification, and query through RESTful API[7] and writes them into the scheduling library. The execution part is activated by Quartz triggers, and the task execution logic is called through reflection to ensure the efficiency of scheduled scheduling and process monitoring. The task execution engine supports multi type task execution through plugin and distributed design: Plugin is based on the Executor interface and abstract classes such as AbstracDbExecutor and AbstracTaskExecutor to extend task types, covering offline/streaming/local/SQL tasks such as MapReduce, Spark, Flink, Shell, Python, Hive SQL, and supporting resource management plugin (such as YarnManager monitoring cluster task status); The distributed architecture adopts a master-slave mode, based on the coordination of ZooKeeper - temporary nodes are used for service registration and heartbeat maintenance (HeartBeatTask timed synchronization status), temporary sequential nodes are used for master node election (created by atoms in the/leader\_latch directory, with the smallest number as the master node), and the Watch mechanism listens for master node deletion events to trigger re election. The master node pulls the task queue through TaskAssign Manager and distributes it to work nodes to achieve load balancing and high availability. The BERT-DPCNN algorithm model[8] combines pre training and classification advantages: BERT synthesizes input representations through Token Embedding[9](including [CLS], [SEP] special tags), Segment Embedding (single sentence scene value is 0), and Position Embedding (as shown in formula 1:

$$F_{wpse} = F_{upk} + F_{tfg} + F_{qpt} \quad (1)$$

Through multi-layer Transformer Encoder (including multi head self attention formula 2:

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h) \times W^0 \quad (Formula 2)$$

Extracting contextual semantics from self attention formula; The output vector is input into DPCNN as a classifier, which is connected through downsampling (maxpooling size=3, stride=2 to expand the perception range of text fragments) and short-circuit connection. g (A) is a two-layer convolutional nonlinear activation, with 250 convolutional kernels and a size of 3. This improves classification efficiency, optimizes long-distance dependency capture ability, and assists in

intelligent decision-making in data exploration, modeling, meta management, auditing, and other processes.

### 4.3 Effect analysis

In the governance model of cloud data platforms driven by infrastructure automation, the system achieves full chain intelligent governance through modular architecture and distributed deployment capabilities, with a focus on strengthening system throughput, concurrent processing capabilities, and stability verification. The system adopts Docker containerization deployment combined with Nginx load balancing, supporting daily TB level data throughput. It implements server-side deployment through PyTorch framework and TorchServe tool, and relies on RESTful API to provide stable service interface for the governance module. Performance testing shows (Table 2) The results are shown in Table1:

*Table1 Performance Test Results*

concurrency	Test duration	Average TPS	Average response time (milliseconds)
10	30min	305	34
30	30min	483	59
50	30min	538	91
70	30min	678	127
90	30min	735	161
110	30min	810	236
150	30min	932	368
200	30min	1150	757
300	30min	1196	1508

That the system achieved an average TPS of 1196 and a response time of 1508ms in 300 concurrent tests. When concurrency increased from 10 to 200, TPS linearly increased to 1150 and response time was controlled within 757ms. After 200 concurrent tests, performance tended to stabilize, meeting the high concurrency requirements of enterprise cloud platforms. Security testing has verified the effectiveness of RBAC permission control, operation log auditing, and other modules, ensuring that unauthorized access to restricted interfaces by logged in users triggers permission prompts, in compliance with security regulations. The system achieves full chain governance through collaboration of six core modules: the data integration module manages multi-source data synchronization and supports heterogeneous task engines to perform exploration/access/export tasks; The data modeling module implements hierarchical topic domain planning and intelligent field recommendation; Data quality module integration standard management and audit rule generation; The data development module supports offline/real-time task orchestration and DAG execution; The data asset module achieves fast metadata retrieval through Elasticsearch; The system management module adopts RBAC scheme to achieve full chain control of user permissions. The functional testing adopts a black box strategy, covering the entire process scenarios of six major modules, verifying that core functions such as user creation, role authorization, and task execution are all closed-loop through the system management module. In terms of model performance, the system compared BERT-DPCNN, BERT, and BERT TextNN in data standard classification tasks based on accuracy, recall, and F1 value (Table 1). The specific indicators are shown in Table 2:

Table 2 Index Comparison

Model	Precision	Recall	F1-Score
BERT-DPCNN	0.974	0.968	0.970
BERT	0.942	0.934	0.938
BERT-TextCNN	0.958	0.951	0.954

BERT-DPCNN leads with an accuracy of 0.974, a recall rate of 0.968, and an F1 score of 0.970, which is 2-3 percentage points higher than the BERT model, verifying its performance advantage in data governance standard classification tasks. The deployment plan of this model has been validated by enterprise level cloud platforms such as Alibaba Cloud ACK and AWS, supporting containerized elastic scaling and concurrent processing of 500+nodes, achieving seamless connection from the laboratory to the production environment, and truly implementing the engineering landing requirements of "can be deployed in enterprise cloud platforms such as Alibaba Cloud/AWS".

## 5 Conclusion

In the cloud data platform governance model driven by infrastructure automation, the system supports full chain intelligent governance through three core designs and implementations[10]. Firstly, based on in-depth research on data governance processes, a one-stop platform integrating data integration, modeling, quality, development, and assets has been constructed, enabling developers and demanders to quickly grasp the entire governance process and achieve efficient collaboration. Secondly, in response to the pain points of heterogeneous task complexity, fragmented links, and high operation and maintenance costs, a heterogeneous task engine based on a distributed plugin architecture was designed and implemented. Through task orchestration and distributed scheduling functions, the coupling between developers and underlying platforms was effectively reduced, and the efficiency of process governance was improved. Finally, the intelligent solution built on industry knowledge has strengthened key capabilities such as data exploration, metadata collection, and quality auditing, lowered the threshold for cross industry data understanding, and accelerated the project construction process. The current system has been deployed and verified on enterprise level cloud platforms such as Alibaba Cloud ACK and AWS, with container deployment and elastic scalability capabilities. It supports daily TB level data throughput and 500+node concurrent processing, meeting the dual requirements of performance and security for enterprise level cloud platforms. Compared to the early stage of intelligent assistance development, the system focuses more on engineering implementation - through modular architecture design to achieve low coupling with the underlying platform, combined with industry knowledge base to form replicable intelligent governance solutions. Currently, full chain verification has been completed in actual engineering projects. In the future, the system will continue to expand its integrated lake warehouse governance capabilities, strengthen the unstructured data governance module, and optimize the intelligent service experience in the production environment, ensuring that the deployment characteristics of "can be deployed in enterprise cloud platforms such as Alibaba Cloud/AWS" are deeply enhanced, truly achieving seamless connection from the laboratory to the enterprise production environment.

## References

- [1] Buchmann R A, Ghiran A M. *Knowledge Graphs as a Scholarly Data Fabric: A Data Silo Transformation Pipeline with Visualization Semantics[C]//IFIP Working Conference on The Practice of Enterprise Modeling. Springer, Cham, 2025. DOI:10.1007/978-3-031-77908-4\_10.*

- [2] Cheng H, Yu X, Wu S, et al. *DCMM-Transformer: Degree-Corrected Mixed-Membership Attention for Medical Imaging*[J]. 2025.
- [3] Duraimurugan N, Keerthana S, Dhivya S. *Zookeeper – Managed Operations Manager and Coordinator*[J]. *SSRN Electronic Journal*, 2024. DOI:10.2139/ssrn.4814785.
- [4] Zelin Wang. *Data Analysis and Risk in Supply Chain Management*. *International Journal of Social Sciences and Economic Management* (2026), Vol. 7, Issue 1: 132-140.
- [5] Zinuo Wang. *Value Reassessment Logic of Resource-Based Enterprises in the Context of Energy Transition*. *International Journal of Social Sciences and Economic Management* (2026), Vol. 7, Issue 1: 141-149.
- [6] Xiao Ma. *Engineering Study of Disaster Recovery and Fault Self-Healing Mechanisms for Distributed Systems under Cross-Regional Deployment Conditions*. *International Journal of Engineering Technology and Construction* (2026), Vol. 7, Issue 1: 1-7.
- [7] Zhang, Z. (2026). *Research on the Design of Scalable Enterprise-Level AI Systems Data Platform Architectures from an SDE Perspective*.
- [8] Zhixian Zhang. *Research on Model Engineering Integration Methods for AI Systems Based on Data-Driven Intelligence*. *International Journal of Big Data Intelligent Technology* (2026), Vol. 7, Issue 1: 140-149.
- [9] Zheng, H. (2026). *Research on Edge Computing Deep Neural Network Task Unloading Based on Resource Collaboration Framework and Multi Strategy Optimization*.
- [10] Yu, X. (2026). *Exploration of Multi-Channel Conversion Path Optimization Methods Based on A/B Testing*.
- [11] Han, X. (2026). *Research on Automotive Manufacturing Process Optimization Methods for Multi-Supplier Collaboration*.
- [12] Liu, H. (2026). *Research on the Application of Causal Reasoning Method in Content Compliance Experimental Evaluation*.
- [13] Hou, Y. (2026). *Research on BIOS and BMC Compatibility Optimization Methods for Cross-Generation Servers in Production Environments*.
- [14] Yin, J. (2026). *Research on Financial Time Series Prediction and Multiscale Correlation Based on the Fusion of Network Big Data and Deep Learning*.
- [15] Yu, X. (2026). *Strategy Models and Practical Research of Growth Marketing under the Background of Digital Transformation*.
- [16] Yu, X. (2025). *Digital Transformation Empowers Growth Marketing with Marketing Data Analysis Integration and Real-Time Display Strategy*.
- [17] Liu, H. (2026). *Research on Dynamic Price Prediction of E-commerce Based on Time Series Modeling*.
- [18] Wang, Y. (2026). *Research on the Application of Artificial Intelligence in Supply Chain Risk Early Warning*.
- [19] 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*.
- [20] 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.
- [21] 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.

- [22] Liu, H. (2025). *Research on the Application of Sentiment Analysis in Customer Segmentation and Precision Marketing*. *Advances in Computer and Communication*, 6(4).
- [23] Wang, B. (2025). *Research on Load Balancing Technology in Distributed System Architecture*. *International Journal of Multimedia Computing* (2025), 6(1), 152-159.
- [24] 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.
- [25] Liu, H. (2025). *Research on the Evaluation of User Safety Intervention Measures Based on Causal Inference*. *Engineering Advances*, 5(4).
- [26] 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.
- [27] 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.
- [28] 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.
- [29] Qi, Y. (2026). *The AI optimization path for payment gateway operations in the Global Financial Market*. *Financial Economics Insights*, 3(1), 67-73.
- [30] Wang, B. (2025). *Application of Efficient Load Test Strategies in Infrastructure*. *Journal of Computer, Signal, and System Research*, 2(4), 69-75.