Scholar Publishing Group
International Journal of Multimedia Computing
https://doi.org/10.38007/IJMC.2025.060112
ISSN 2789-7168 Vol. 6, Issue 1:126-134

Performance Optimization and Implementation Pathways of Advertising Delivery Systems from a Full-Stack Development Perspective

Taige Zhang

Department of Computer Science, Rice University, Houston, TX 77005

Keywords: Full stack development, Advertising delivery system, Performance optimization, DISM model, Real-time data warehouse

Abstract: With the rapid development of the Internet and digital advertising market, the scale of the global advertising market continues to expand, and the user behavior mode has undergone fundamental changes due to the popularity of mobile devices, which puts forward higher requirements for the real-time, personalized and effect tracking of advertising. Traditional advertising delivery systems face core challenges such as real-time data processing delays, inaccurate capture of user interests, and reliance on outdated information for delivery strategies, leading to decision-making errors and resource waste by advertisers. This study is based on a full stack development perspective, following the entire software engineering lifecycle. Through requirement analysis, the system's functional and non-functional boundaries are clarified, and a layered architecture design is adopted to divide advertising promotion management, advertising push, advertising log management, and real-time data warehouse into four modules; Use class diagrams, flowcharts, and sequence diagrams to complete module modelling and code development, and utilize streaming data processing technologies such as Flink and Kafka to achieve real-time streaming data processing and large screen display of advertising effects. Aiming at the optimization problem of advertising ranking model, a DISM model integrating DSSM dual tower model and FM factorization machine is proposed to effectively solve the shortcomings of DIN model in deep level interest mining and low-level feature interaction learning of user behavior sequences, significantly improving CTR and RPM indicators. The system verifies reliability through functional and non-functional testing, meeting the needs of advertisers for ad creation and push, real-time effect analysis, as well as administrator ad approval and log collection. The real-time data warehouse module assists advertisers in adjusting their advertising strategies in real time. This study forms a complete closed loop of "requirements design implementation optimization testing", providing a performance optimization path from the perspective of full stack development for the efficient and intelligent implementation of advertising delivery systems. In the future, it can further deepen the direction of recall algorithms, coarse ranking models, data indicator improvement, and model optimization.

1. Introduction

With the rapid development of the Internet and digital advertising market, the scale of the global advertising market continues to expand - in 2023, the size of the Internet advertising market will

reach 573.2 billion yuan (a year-on-year increase of 12.66%), and the global mobile advertising spending will exceed 362 billion dollars. The user behavior model has undergone fundamental changes due to the popularity of mobile equipment, which puts forward higher requirements on the real-time, personalized and effect tracking of advertising. However, traditional advertising delivery systems face core challenges such as real-time data processing delays, inaccurate capture of user interests, and reliance on outdated information for delivery strategies, leading to decision-making errors and resource waste by advertisers. To overcome the above bottlenecks, this article focuses on the performance optimization and implementation path of the advertising delivery system from a full stack development perspective. The specific work includes clarifying the functional and non-functional boundaries of the system through requirement analysis, and using a layered architecture design to divide the four major modules of advertising promotion management, advertising push, advertising log management, and real-time data warehouse; Use class diagrams, flowcharts, and sequence diagrams in detailed design to complete module modeling and code development; Aiming at the optimization problem of advertising ranking model, a DISM model integrating DSSM dual tower model and FM factorization machine is proposed to effectively solve the shortcomings of DIN model in user behavior sequence and candidate advertisement deep level interest mining, low-level feature interaction learning, and significantly improve CTR and RPM indicators compared to traditional models; Finally, the reliability of the system is verified through functional and non-functional testing, forming a complete closed loop of "requirements design implementation optimization testing". In terms of structure, the introduction elaborates on the background significance, and the main body focuses on system design and model optimization. Finally, the contribution is verified through testing, providing a performance optimization path from a full stack development perspective for the efficient and intelligent implementation of advertising delivery systems.

2. Correlation theory

2.1. Analysis of Real time Data Warehouse and Flink Stream Processing Framework

Real time data warehouse is a system architecture designed for low latency processing of real-time data streams, typically divided into data storage layer [1] (ODS), dimension layer (DIM), detail data layer (DWD), and summary data layer (DWS). The ODS layer collects advertising business data and log data and stores them in Kafka, providing source data for other layers; The DIM layer processes advertising dimension data (such as region, category, and other attributes) and stores it in HBase as a condition for querying and grouping advertising effectiveness data; DWD layer processes advertising fact table data (such as exposure times, click through times, and other metrics); The DWS layer forms a real-time advertising effect data wide table by associating the DWD layer fact table with the DIM layer dimension table, optimizing query analysis performance through redundant data storage. Flink, as an open-source high-performance real-time stream processing framework, has the core advantages of low latency, high throughput, and fault tolerance. It supports unbounded data stream processing and reads data from multiple sources through Flink Source. After conversion, aggregation, window calculation, and other operations, Flink Sink outputs the data to other databases. The FlinkSQL [2] programming model it provides can simplify real-time streaming data processing, while FlinkCDC data change capture technology can synchronize database change operations (such as insertion, update, and deletion) in real time, enabling real-time data processing and analysis, supporting the real-time and personalized requirements of advertising delivery systems.

2.2. Deep Interest Network Model

Deep interest network [3] is an advertising ranking deep learning model proposed in 2018, which learns the degree of attention users have towards different interests through attention mechanism. The model takes the user's historical behavior sequence as the input [5], calculates the weight according to the correlation between each historical behavior and the current candidate advertisement, and dynamically captures the user's interest bias - such as computers [6], eye shadow disks, basketball and badminton in the user's historical browsing advertisements [7]. The eye shadow disk feature vector is far more important than other items in predicting the click rate of lipstick advertisements. The innovation lies in the introduction of an Activation Unit: by taking the outer product of the user's historical advertisement feature vector and the candidate advertisement feature vector [8], concatenating the candidate advertisement and historical advertisement features, and outputting weights through a fully connected layer [9]. The specific calculation formula is the weighted sum of the user's historical behavior feature vector. In addition, DIN uses Dice activation function instead of traditional PReLU to solve the distribution problem at the correction point of 0. This function adaptively adjusts the correction point value based on the current batch data distribution and smooths the curve near the correction point [10]. In the formula, E (s) is the mean of the batch data and Var (s) is the variance, achieving a more flexible activation effect. The activation unit structure is shown in Figure 1

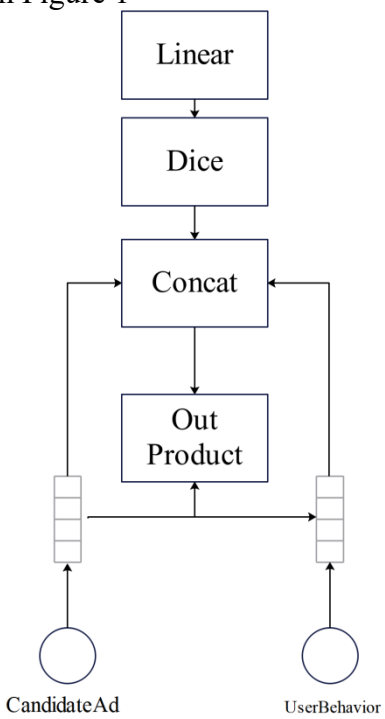

Figure 1. Structure of the fusion model between advertising candidates and user behavior characteristics

The DIN model effectively improves the accuracy of advertising ranking through the above mechanism, becoming a key technology for modeling user interests in advertising delivery systems.

3. Research method

3.1. Framework for Requirement Analysis of Advertising Placement System

Requirement analysis, as the foundation of software design, directly affects system quality. This system revolves around the core goal of advertising placement, and is developed from two aspects:

functional requirements and non-functional requirements. Functional requirements cover four major modules: advertising promotion management, advertising push, log management, and real-time data warehouse. Advertising promotion management supports advertisers to create three-dimensional advertisements through promotion plans, promotion groups, and advertising creativity. It includes sub functions such as advertising information management (including promotion plan/group/creative editing and targeted settings), creative approval management (supporting approval process configuration and task priority processing), traffic anti cheating (integrating abnormal IP detection and token bucket flow limiting mechanism), and advertising billing management (supporting click/display billing, distributed lock protection for concurrent fee accuracy, and dead letter queue processing for abnormal tasks); The advertisement push module includes advertisement retrieval, recall, and sorting, among which the sorting function directly affects the effectiveness of advertising; Log management is responsible for collecting and processing business logs and user behavior logs, providing data sources for real-time data warehouses; The real-time data warehouse module enables real-time processing and decision support of advertising effectiveness data. Non-functional requirements focus on system reliability, real-time performance, and security to ensure efficient and stable operation of the entire advertising delivery process.

3.2. Depth of Advertising Push and Real time Data Warehouse Function Modules

The ad push function takes the ad acquisition request triggered by user access to the media app as input, and achieves precise push through three stages: ad filtering (based on IP blacklist), ad recall (encapsulating user characteristics as recall context, searching for MySQL candidate ads through ElasticSearch after retrieving the ad index), and ad sorting (selecting the top K ads that the user is interested in). The use case diagram shows the entire process interaction logic. Advertising log management intercepts business logic through AOP to generate deduction logs and other types of logs, synchronously stores them on local log servers and real-time data warehouses, and assists administrators in quickly grasping the log overview through preliminary cleaning, partitioning, and aggregation. As the core of the system, the real-time data warehouse adopts the Alibaba OneData methodology to uniformly manage data indicators. The architecture is divided into four layers: ODS (collecting business data and user behavior logs and distributing them to different topics on Kafka), DIM (synchronizing dimension data to HBase and Redis cache, dynamically splitting dimension tables to process changes), DWD (associating business flow and dimension data to form fact wide tables), and DWS (generating advertising revenue, display volume, click through volume and other indicators through Flink operator processing and storing them in Doris database, supporting sub second level queries), realizing real-time storage, processing and visual analysis of advertising effectiveness data, providing data support for advertiser decision-making.

3.3. Overview of Nonfunctional Requirements and Architecture Design for Advertising Placement System

The non-functional requirements of this system are built around four dimensions: real-time, stability, scalability, and usability. Real time visualization of advertising effectiveness indicators is achieved through Kafka memory storage, Flink concurrent processing, HBase/Redis secondary indexing, and Toris sub second queries; Stability is ensured by adopting a one master two slave cluster architecture and HDFS multi replica log backup to ensure high availability of the system; Scalability relies on the factory/strategy/status/observer mode to support dynamic iteration of modules; Ease of use provides professional/fast dual promotion mode to simplify advertiser operations. The system architecture adopts a six layer layered design: the data display layer

implements front-end interaction and RPC request encapsulation; The interface layer is based on Spring Cloud Gateway to complete authentication/routing/load balancing and fuse degradation; The business processing layer (developed by SpringBoot) includes advertising and promotion management (including information management/creative approval/traffic anti cheating/billing submodules, as shown in Figure 2)

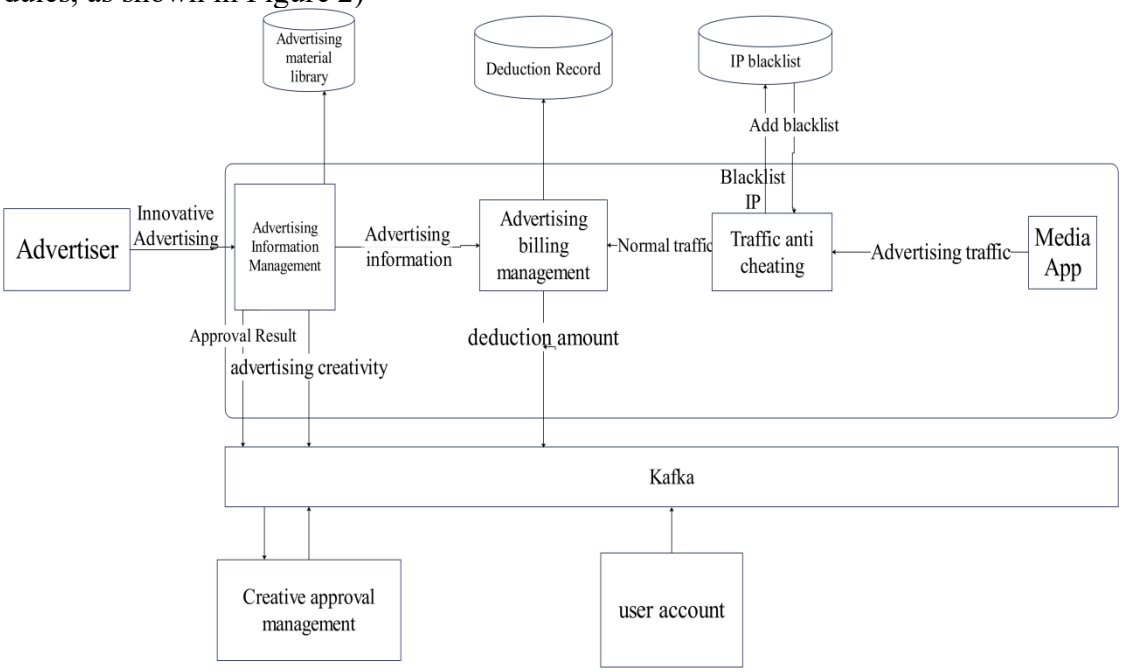

Figure 2. Full process architecture and key modules of the advertising management system

In the ad push module, the recall retrieves ad IDs through Elasticsearch's inverted index, and the sorting is optimized using the DISM model to estimate click through rates; The data collection layer collects logs through Flume and transmits them to the computing layer via Kafka; The computing layer is based on real-time data warehousing to complete stream data processing - the ODS layer (Flume+Maxwell synchronized to Kafka) stores raw logs and incremental data, the DIM layer (HBase+Redis cache) processes dimension data and supports FlinkCDC dynamic dimension table updates, the DWD layer constructs explicit detail tables (such as advertising click through traffic detail tables) through FlinkSQL multi stream associations, and the DWS layer aggregates business indicators through Flink operators and stores them in Doris; The storage layer integrates ElasticSearch/Redis indexes, HBase dimension tables, Kafka source data, and Doris metric library. The log management module adopts AOP to decouple the collection logic, and completes partition aggregation and preliminary statistics after transmission through Kafka. The full chain architecture and module substructure collaborate to support the efficient operation of the entire advertising delivery process, achieving closed-loop management from ad creation, approval, billing to push and effect tracking, meeting the real-time decision-making and precise delivery needs of advertisers.

4. Results and discussion

4.1. Panoramic design of advertising placement system database

In the full stack development of the advertising placement system, performance optimization runs through the entire process design of the advertising promotion management module. Advertising information management achieves dynamic extension of creative types through the

combination of strategy mode and factory mode - AdCreativeFactory factory class uniformly manages the creation of creative subclasses, and AdCreativeService base class supports the extension of graphic/video/text creative types through dynamic binding mechanism, solving the problem of traditional if else judgment of scalability and improving code maintainability. Creative approval management is based on the Activiti workflow engine to build a multi-level approval process. WorkFlowService is responsible for creating and managing process instances, AuthorityService implements approval group permission binding and user permission verification, CreativeApprovalService asynchronously pulls approval tasks through Kafka message queue, combines ThreadPoolExecutor thread pool and PriorityBlockingQueue priority queue to achieve efficient processing of approval tasks, and uses Observer mode to achieve real-time notification of approval results through the Adviser class, optimizing approval efficiency. The traffic anti cheating module integrates abnormal IP detection and flow limiting functions. The BloomFilter class uses a bitmap to store the IP blacklist, calculates the hash value through the MurmurHash algorithm, and dynamically adjusts the storage space by combining the bitmap length and misjudgment rate formula to achieve O (1) level abnormal IP detection efficiency; The RateLimit class uses the token bucket algorithm to implement traffic speed limiting. Overlimited IPs are managed by the IPBlacklistManager multi-level blacklist, combined with the IPWhitelistManager whitelist mechanism to ensure normal traffic flow and reduce system IO overhead. The full module design is achieved through class diagrams, flowcharts, and schematic diagrams (as shown in Figure 3)

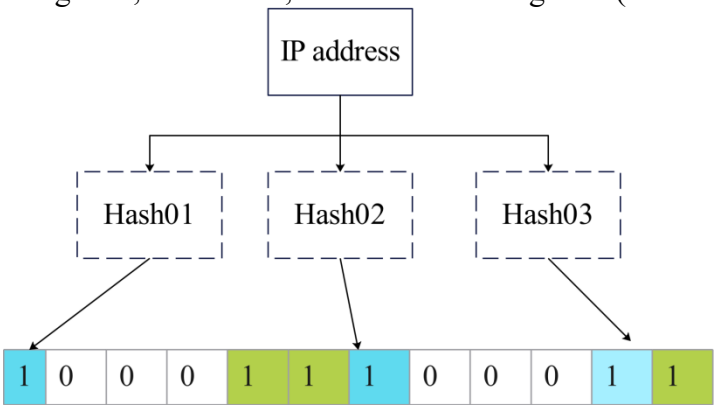

Figure 3. Analysis of the process of mapping IP address hash to bitmap

Visual presentation supports efficient operation throughout the entire lifecycle of advertising creation, approval, billing, delivery, and effect tracking, forming a full stack performance optimization path from design pattern optimization, workflow engine integration to algorithm performance improvement. Performance optimization of advertising delivery system from the perspective of full stack development focuses on collaborative design of traffic anti cheating and billing management: Traffic anti cheating adopts Bloom filter to implement distributed IP blacklist management, stores hash mapping through bitmap and Redis, supports O (1) level abnormal IP detection, and cooperates with token bucket algorithm to achieve hierarchical flow restriction (1 hour for level 1/1 day for level 2/permanent ban for level 3), and manages token bucket objects based on LRU algorithm and LinkedHashMap to reduce memory overhead; Advertising billing builds an order finite state machine through a state mode, supporting multiple state transitions such as pending payment/paid/refunded. It combines Kafka asynchronous queue and Redis distributed lock to ensure idempotent deduction, uses Lua scripts to achieve atomicity of lock operations, and processes failed tasks through a dead letter queue, forming a full link performance improvement path from hash mapping optimization, distributed lock control to state machine management, supporting efficient and stable operation of the entire process of advertising billing, anti-cheating,

and delivery.

4.2. Model experiment

The performance optimization of the advertising delivery system from the perspective of full stack development runs through the design of the entire advertising push chain, achieving full stack performance improvement through model architecture innovation and feature interaction enhancement. The advertising filtering submodule constructs a recall request context based on IP blacklists and user feature queries, achieving early interception of abnormal requests; The recall submodule relies on Elasticsearch inverted indexing and Redis cache acceleration, combined with user targeting (such as city, gender) and ad space features (such as page type) to construct efficient recall expressions and quickly filter candidate ad collections; The sorting submodule adopts the Deep Interest Similarity Model (DISM) and introduces the DSSM dual tower structure optimization attention mechanism based on the DIN model (calculating the inner product similarity between user historical advertising towers and candidate advertising towers+heat penalty to improve long tail exposure). It also integrates FM model to learn second-order feature interaction (such as "sci-fi movie male" crossover), and finally achieves click through rate prediction through Sigmoid function fusion output - AUC reached 0.7399 on the MovieLens dataset (3.22% higher than DIN, RelaImpr 7.43% higher), and AUC reached 0.7495 on the Amazon (Electron) dataset (1.61% higher than DIN), RelaImpr 5.59%）， The performance is superior to models such as LR, FM, DNN, Wide&Deep, DeepFM, XGBDeepFM, DIN, DRIN, IARM, etc. As shown in Table 1

Table 1. Performance comparison of different models on MovieLens and Amazon (Electro) datasets

model	MovieLens AUC	MovieLens RelaImpr	Amazon (Electro) AUC	Amazon (Electro) RelaImpr
LR	0.7163	-3.13%	0.7233	-5.50%
FM	0.7201	-1.43%	0.7303	-2.53%
DNN	0.7233	0.00%	0.7363	0.00%
Wide&Deep	0.7283	2.23%	0.7383	0.84%
DeepFM	0.7303	3.13%	0.7397	1.44%
XGBDeepFM	0.7315	3.67%	0.7418	2.33%
DIN	0.7327	4.21%	0.7457	3.98%
DRIN	0.7351	5.28%	0.7462	4.19%
IARM	0.7356	5.51%	0.7473	4.66%
DISM	0.7399	7.43%	0.7495	5.59%

The full module collaborative design supports the full chain optimization from request filtering, candidate calling back to precise sorting, forming a full stack performance improvement path of "data index optimization (Elasticsearch/Redis acceleration)+model architecture innovation (DSSM/FM enhanced feature interaction)+feature cross enhancement (second-order feature learning)", ensuring the real-time and effective advertising placement in high concurrency scenarios.

4.3. Effect analysis

Software testing verifies system functionality and non-functional requirements by constructing test cases. The testing environment is built on a Hadoop distributed cluster and integrated Flume 1.9.0、Kafka 3.0.0、Maxwell 1.29.2、MySQL 5.7.16、HBase 2.4.11、Redis 6.0.8、Flink 1.17.1、Doris 1.2.4.1 Waiting for middleware and database. Functional testing covers advertising promotion management (such as ad plan creation, abnormal IP filtering, graphic and creative generation, etc.), ad push (normal/abnormal IP request processing, feature matching, and ad sorting), ad log

management (click/display/billing log collection and analysis, and synchronization to real-time data warehouse), and real-time data warehouse (real-time data processing and visualization of user clicks/ad orders/dimension table modifications, etc.), all test cases have been validated. In terms of non-functional testing, performance testing verifies the timeliness of ad creation (responded within 3 seconds), ad request processing (completed within 2 seconds), and effect data generation (completed within 1 second); Reliability testing verifies Kafka's ability to read data from previous offsets even after restart, Hadoop node failure, or system normal operation and election of a new primary server after restart. At present, all test cases have passed, known defects have been fixed, and the system meets the requirements of functional correctness, performance stability, and reliability.

5. Conclusion

The design and implementation of the advertising placement system strictly follows the software engineering lifecycle, completing core processes such as requirement analysis, preliminary design, detailed design, and system testing. The system is divided into four modules: advertising promotion management, advertising push, advertising log management, and real-time data warehouse. The real-time data warehouse module adopts streaming data processing technologies such as Flink and Kafka to process real-time streaming data of advertising effectiveness and display it in real time through a large screen, helping advertisers adjust their advertising strategies in real time; In response to the problems of inaccurate calculation of user deep level interest weights and insufficient interaction learning of low-level features in the DIN model, the model structure is optimized to construct user historical advertising towers and candidate advertising towers to calculate deep level relevance. The FM model is introduced to mine second-order interaction features, and the DISM model is proposed to significantly improve the advertising ranking performance. The system discovers and fixes defects through functional and non-functional testing, improves reliability, and meets the needs of advertisers for ad creation and push, real-time effect analysis, as well as administrator ad approval and log collection. Future work prospects focus on performance optimization and implementation path deepening: introducing a recall algorithm based on graph neural networks, constructing a graph structure through entity nodes and connection relationships such as advertisements and users to improve recall accuracy and efficiency; Introducing a coarse ranking model to quickly screen advertisement candidate sets to reduce the burden of subsequent ranking calculations; Improve real-time data warehouse data indicators, add dimensions such as advertising retention curve analysis and access time distribution; Optimize rule-based anomaly traffic detection, utilize real-time data warehouse to analyses anomaly data and improve rules such as retaining anomaly curves, abnormal click through rate distribution, etc; Introducing Transformer to model user behavior sequences for complete user behavior sequence scenarios, and using advanced algorithms to model advertising context information to learn user context aware interests, further optimizing the performance of DISM algorithm. This study forms a complete closed loop of "requirements design implementation optimization testing" from the perspective of full stack development, providing performance optimization and implementation path support for the efficient and intelligent implementation of advertising delivery systems.

References

[1] Ghadimi M, Baghayi N, Shateri A.DataBay: A Unified Platform for Automating Data Warehouse Management, Real-Time Data Processing, and Ensuring Data Quality and Monitoring[J]. 2025.DOI:10.36227/techrxiv.173834980.04708005/v1.

[2] *Fan X, Lu J. Enterprise Level Data Warehouse System Based on Hive in Big Data Environment[J].Procedia Computer Science, 2024, 243:67-75.*
[3] *Zhang T, Jin X, Bai S, et al. Smart Public Transportation Sensing: Enhancing Perception and Data Management for Efficient and Safety Operations[J].Sensors (14248220), 2023, 23(22).DOI:10.3390/s23229228.*
[4] *Chang, Chen-Wei. "AI-Driven Privacy Audit Automation and Data Provenance Tracking in Large-Scale Systems." (2025).*
[5] *Li W. Building a Credit Risk Data Management and Analysis System for Financial Markets Based on Blockchain Data Storage and Encryption Technology[C]//2025 3rd International Conference on Data Science and Network Security (ICDSNS). IEEE, 2025: 1-7.*
[6] *Huang, J. (2025). Balance Model of Resource Management and Customer Service Availability in Cloud Computing Platform. Economics and Management Innovation, 2(4), 39-45.*
[7] *Zhang K. Research on the Application of Homomorphic Encryption-Based Machine Learning Privacy Protection Technology in Precision Marketing[C]//2025 3rd International Conference on Data Science and Network Security (ICDSNS). IEEE, 2025: 1-6.*
[8] *Tang X, Wu X, Bao W. Intelligent Prediction-Inventory-Scheduling Closed-Loop Nearshore Supply Chain Decision System[J]. Advances in Management and Intelligent Technologies, 2025, 1(4).*
[9] *Pinto R C, Tavares A R. PReLU: Yet Another Single-Layer Solution to the XOR Problem[J]. 2024.*
[10] *Xu, H. (2025). Optimization of Packaging Procurement and Supplier Strategy in Global Supply Chain. European Journal of Business, Economics & Management, 1(3), 111-117.*
[11] *Pan, H. (2025). Development and Optimization of Social Network Systems on Machine Learning. European Journal of AI, Computing & Informatics, 1(2), 73-79.*
[12] *Wu X, Bao W. Research on the Design of a Blockchain Logistics Information Platform Based on Reputation Proof Consensus Algorithm[J]. Procedia Computer Science, 2025, 262: 973-981.*