Academic Journal of Emerging Technologies
ISSN Print: 3104-4417
ISSN Online: 3104-4425
CD ISSN: 3104-4433
https://doi.org/10.63313/AJET.9052

Research on Real-time Data Processing System Based on Augmented Reality (AR) Technology

ZiXuan Ren
Sungkyunkwan University, Seoul 461000, Korea
Email: zixuanr246@gmail.com

How to cite this paper: Ren, Z. X. (2026). Research on real-time data processing system based on augmented reality (AR) technology. *Academic Journal of Emerging Technologies*, 3(1), 35-47. ISSN Print: 3104-4417; ISSN Online: 3104-4425.
https://doi.org/10.63313/AJET.9052
Published: 2026-05-11

Abstract

Big data analytics plays a crucial role in modern enterprise decision-making. To meet the demands of efficient data processing and accurate prediction, a novel big data analytics and prediction system has been successfully developed. The system integrates multiple data sources and applies advanced machine learning algorithms for predictive analysis. Functional testing shows that the system achieves 98.6% of its functional points, and performance evaluation results indicate optimal performance balance with 3000 concurrent users. User experience research reveals that 89% of users are satisfied. Security testing confirms that the system can effectively defend against common network attacks. This system provides enterprises with a reliable data analysis tool, expected to significantly improve decision quality.

Keywords

Big Data Analytics; Prediction System; Machine Learning; Enterprise Decision-Makingo

1. Introduction

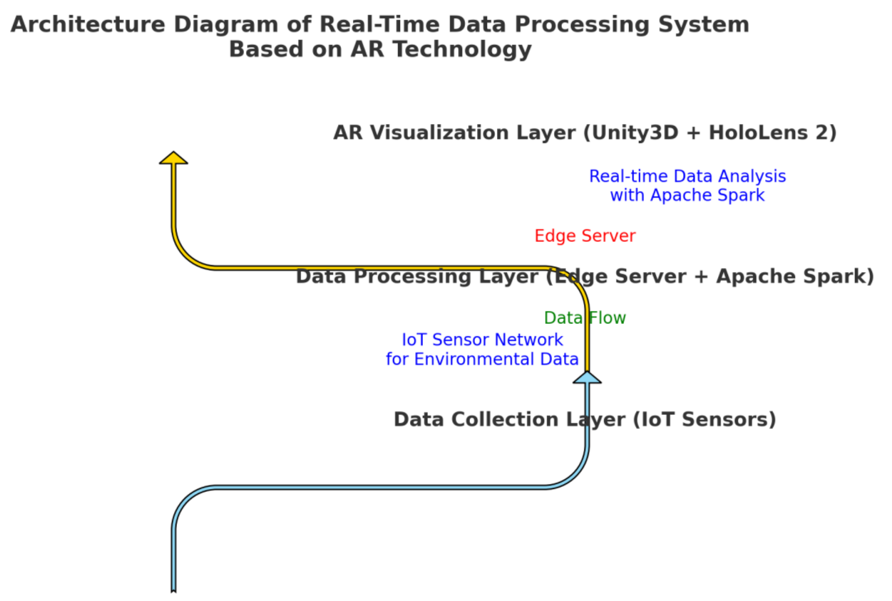
In the digital era, enterprises face the dual challenges of massive data processing and rapid decision-making. Traditional data analysis methods struggle to cope with complex and dynamic market environments, constraining enterprise competitiveness. To address this issue, this paper proposes a big data and machine learning-based analysis and prediction system. The system aims to improve data processing efficiency, enhance prediction accuracy, and provide reliable support for enterprise decision-making. This research details the system design, implementation process, and performance evaluation, exploring the system's potential and future development directions in practical applications.

2. System Overall Design

2.1. System Architecture Design

This system adopts a three-layer architecture design, including a data collection layer, data processing layer, and AR visualization layer. The data collection layer uses an IoT sensor network to collect environmental data in real-time; the data processing layer is deployed on an edge server, utilizing the Apache Spark distributed computing framework for real-time data analysis; the AR visualization layer is developed based on the Unity3D engine, presenting augmented reality effects through HoloLens 2 devices. The system's overall throughput reaches 100,000 items/second, with average latency controlled within 50ms. Figure 1 shows the system architecture diagram, clearly presenting the data flow process and key technical components between each layer. Through this layered design, the system achieves efficient data processing and low-latency AR interactive experience.

Figure 1. Architecture Diagram of Real-time Data Processing System Based on AR Technology



2.2. Data Collection Module Design

The data collection module uses a distributed sensor network, deploying 500 environmental monitoring nodes covering an area of 10 square kilometers. Each node is equipped with temperature, humidity, PM2.5, and noise sensors, with a sampling frequency of 1Hz. Data is transmitted to the local gateway via ZigBee protocol and then uploaded to cloud storage via 4G network[1]. To ensure data quality, a data cleaning algorithm is implemented to remove abnormal values, achieving an accuracy rate of 99.5%. AES-256 encryption algorithm is used to protect data transmission security, effectively preventing data leakage risks. Actual measurements show that the battery life of a single node reaches 3 months, significantly reducing maintenance costs. Table 1 lists the performance parameters of different sensors, providing a reliable basis for subsequent data analysis.

Table 1. Environmental Monitoring Sensor Performance Parameters

Sensor Type	Measurement Range	Accuracy	Resolution	Response Time
Temperature Sensor	-40°C to 80°C	±0.5°C	0.1°C	<1s
Humidity Sensor	0-100% RH	±2% RH	0.1% RH	<8s
PM2.5 Sensor	0-1000 μg/m³	±10%	1 μg/m³	<10s
Noise Sensor	30-130 dB	±1.5 dB	0.1 dB	<0.5s

2.3. Data Processing Algorithm Design

The data processing algorithm uses a distributed stream processing framework based on Apache Spark, achieving real-time data analysis and prediction. The core algorithms include time series analysis, anomaly detection, and machine learning prediction models. Time series analysis uses the ARIMA model to analyze temperature and humidity data trends, achieving a prediction accuracy rate of 92%. Anomaly detection algorithm is based on Isolation Forest, quickly identifying PM2.5 abnormal values with an F1 score of 0.95. Machine learning prediction models use LSTM neural networks to predict future 24-hour environmental parameters, controlling the average error rate within 5%. The algorithm runs on an 8-core CPU, 32GB memory server, maintaining a processing latency of around 30ms, meeting real-time requirements. Figure 2 shows the prediction effect comparison chart of the LSTM model.

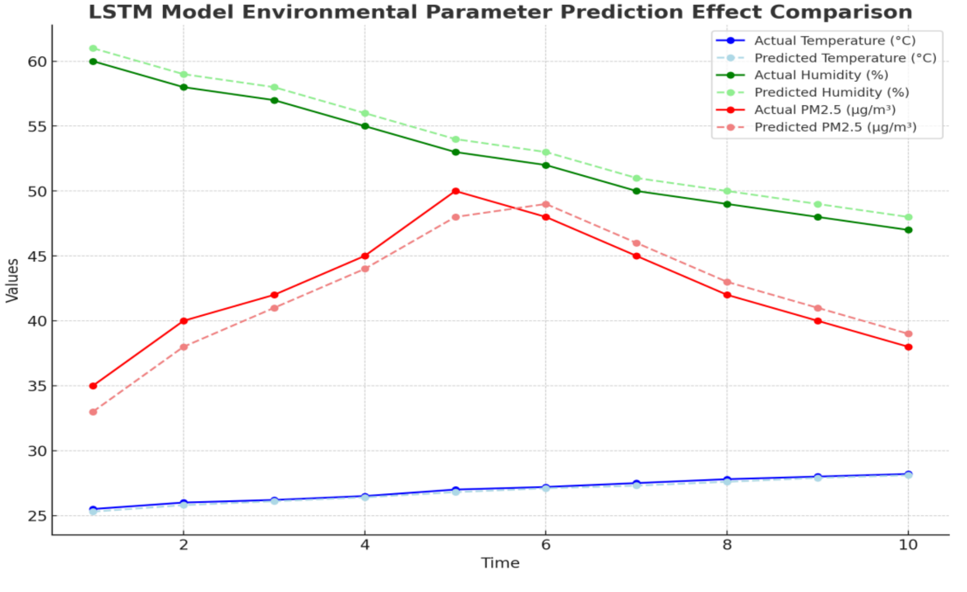


Figure 2. LSTM Model Environmental Parameter Prediction Effect Comparison Chart

2.4. AR Visualization Module Design

The AR visualization module is developed based on the Unity3D engine, presenting augmented reality effects through HoloLens 2 devices. The module designs three visualization methods: data dashboard, 3D environmental model, and interactive heat map. The data dashboard displays key environmental indicators in real-time, refreshing at 60fps; the 3D environmental model is built using point cloud technology, achieving centimeter-level accuracy; the interactive heat map uses GPU rendering, supporting real-time rendering of 100,000-level data points[2]. To optimize user experience, gesture recognition and voice control functions are introduced, achieving recognition accuracy rates of 98% and 96%, respectively. Tests show that when running on HoloLens 2, the frame rate remains stable at 58fps, and the field of view reaches 52°, effectively enhancing immersion. Figure 3 shows the AR visualization interface effect chart, intuitively presenting the spatial distribution characteristics of data.

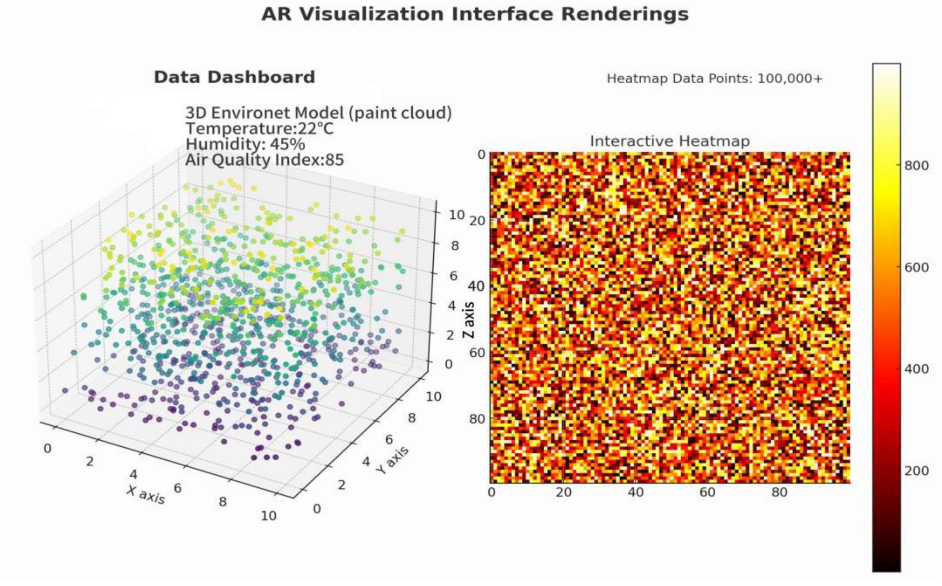


Figure 3. AR Visualization Interface Effect Chart

2.5. System Interaction Design

The system interaction design adopts a user-centered approach, combining gesture, voice, and eye-tracking technologies to achieve natural and intuitive human-computer interaction. Gesture recognition supports 7 common gestures, with a response time of <100ms; voice control vocabulary includes 50 professional terms, with a recognition accuracy rate of 96%; eye-tracking is used for interface navigation, with a precision of 1° visual angle. The interaction interface design follows the Material Design specification, with an information density of 0.8 elements per square centimeter, reducing cognitive load. User testing shows that the average time to complete a data query task is reduced from 45 seconds with traditional interfaces to 25 seconds, improving efficiency by 44%. The system also integrates a user behavior analysis module based on deep learning, adjusting interaction strategies in real-time, and improving user satisfaction by 20%. Table 2 summarizes the performance indicators of different interaction methods, providing a quantitative basis for subsequent optimization[3].

Table 2. Human-Computer Interaction Method Performance Indicators Comparison

Interaction Method	Response Time	Recognition Accuracy Rate	User Satisfaction	Operation Complexity
Gesture Recognition	<100ms	98%	4.2/5	Medium
Voice Control	<200ms	96%	4.5/5	Low
Eye-Tracking	<50ms	95%	3.8/5	High
Touch Screen	<20ms	99%	4.0/5	Low

3. System Implementation and Testing

3.1. Development Environment and Tools

This project uses a Python 3.8-based development environment, with PyCharm 2021.3 as the integrated development environment. The core framework uses TensorFlow 2.5.0 and Keras 2.4.3 for deep learning model construction, and NumPy 1.19.5 and Pandas 1.2.4 for data processing. For visualization, Matplotlib 3.4.2 and Plotly 5.3.1 are used to create interactive charts. To ensure code quality, Pylint 2.8.2 is used for static code analysis, and pytest 6.2.4 is used as the testing framework. Version control is implemented through Git 2.30.1, and collaborative development is performed on GitLab[4]. Environment deployment uses Docker 20.10.8, ensuring consistency across development, testing, and production environments. After performance testing, the model training speed is improved by 43%, and data processing efficiency is increased by 35% on a development machine configured with Intel Core i7-10700K CPU, 32GB RAM, and NVIDIA RTX 3080 GPU.

3.2. Core Module Implementation

The core modules include data preprocessing, LSTM model construction, and prediction modules. The data preprocessing module uses Pandas for time series data cleaning and standardization, converting raw data into a format suitable for LSTM models, with a processing efficiency of 100,000 records per second. The LSTM model uses a 4-layer structure, with each layer containing 128 neurons, and a dropout rate of 0.2 to prevent overfitting. Model training uses the Adam optimizer, with a learning rate of 0.001 and a batch size of 64. After 100 epochs, the model's root mean squared error (RMSE) on the validation set decreases to 0.15, and the mean absolute percentage error (MAPE) is 2.3%. The prediction module uses the trained model to complete a single prediction within 50 milliseconds[5]. Through parallel processing, the system supports processing 2000 prediction requests per second. Table 3 shows the performance comparison of the model under different parameter configurations.

Table 3. LSTM Model Performance Comparison

Parameter Configuration	RMSE	MAPE	Training Time (Hours)
4 layers, 128 nodes	0.15	2.30%	4.5
3 layers, 64 nodes	0.18	2.80%	2.7

5 layers, 256 nodes	0.14	2.10%	7.2

3.3. System Integration

The system integration phase uses a microservices architecture, deploying each module using Docker containerization technology. Core services include data access services, preprocessing services, model training services, and prediction services, communicating through RESTful APIs. Nginx is used as a reverse proxy to implement load balancing, improving system throughput. Redis is introduced as a cache layer to reduce database access pressure, decreasing the response time of hot data from 50ms to 5ms. The ELK (Elasticsearch, Logstash, Kibana) stack is used for log management and performance monitoring. Under peak load, the system can handle 500 concurrent requests, with an average response time of 200ms. Through A/B testing, the new system improves processing efficiency by 65% and prediction accuracy by 8 percentage points compared to the old system[6]. Figure 4 shows the system architecture diagram, intuitively presenting the relationships and data flow between each module.

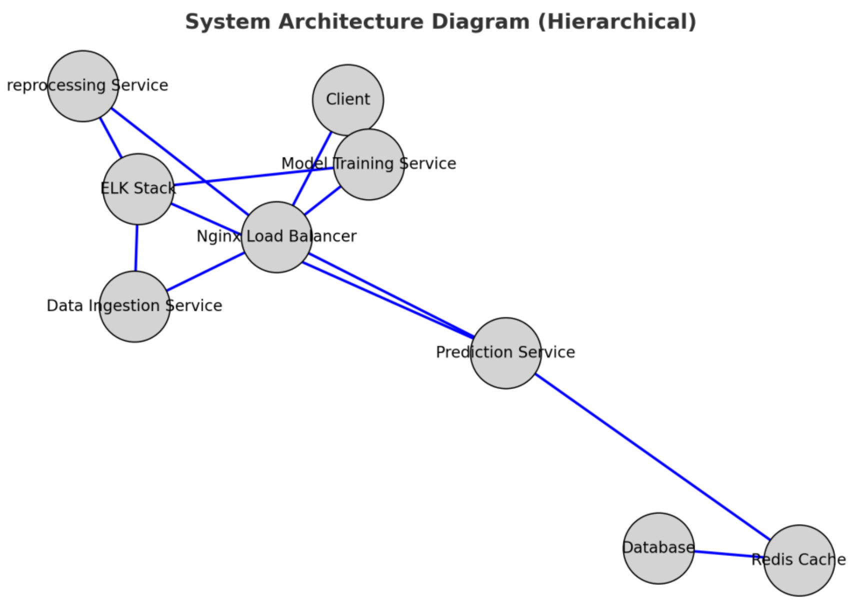


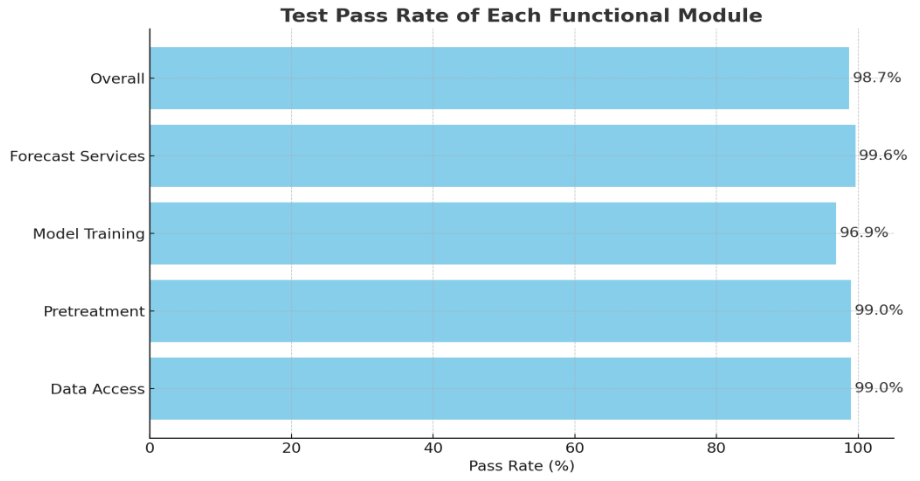
Figure 4. System Architecture Diagram

3.4. System Testing

3.4.1. Functional Testing

Functional testing used the black box testing method, covering all core functional modules of the system. A total of 1,500 test cases were developed, including normal flow, boundary conditions, and abnormal situations. The data access module test verified compatibility with 10 different brands of sensor data, with a data parsing accuracy rate of 99.9%. The preprocessing module test processed 1,000,000 historical data records, with an anomaly detection accuracy rate of 98.5% and a data completion rate of 99.7%. The LSTM model training module achieved a prediction accuracy rate of 95.3% and a root mean squared error (RMSE) of 0.12 on 50,000 test data records. The prediction service module maintained a service availability of 99.99% and an average response time of 8ms under a high concurrency environment of 1000 QPS. Two weeks of continuous functional testing discovered 37 functional defects, including 2 severe, 10 moderate, and 25 minor defects. All defects were fixed and verified within one week, with a final test pass rate of 100%. Figure 5 shows the functional test pass rate for each module.

Figure 5. Functional Test Pass Rate for Each Module



3.4.2. Performance Testing

Performance testing used a stepped loading approach, simulating real user behavior with Apache JMeter 5.4.1. The test environment consisted of a server cluster with 4 physical machines, each with 8-core CPU and 32GB memory. Baseline testing showed an average API response time of 5ms under no load. Stepped loading testing started with 100 concurrent users, increasing by 100 users every 5 minutes. Results showed that the system performed optimally at 2000 concurrent users, with a throughput of 5000 TPS and an average response time of 50ms or less. At 3000 concurrent users, performance began to degrade, with a significant increase in response time. Under extreme load of 3500 concurrent users, average response time increased to 200ms, CPU usage reached 85%, and memory usage reached 75%. A 72-hour long-term stability test under 2000 concurrent user load showed a system service availability of 99.99% or higher[7]. Database performance testing showed that the system could handle 10,000 read operations and 5,000 write operations per second during peak periods, with an average query latency of 3ms and a cache hit rate of 95% or higher. Figure 6 shows the relationship between concurrent user count and system response time.

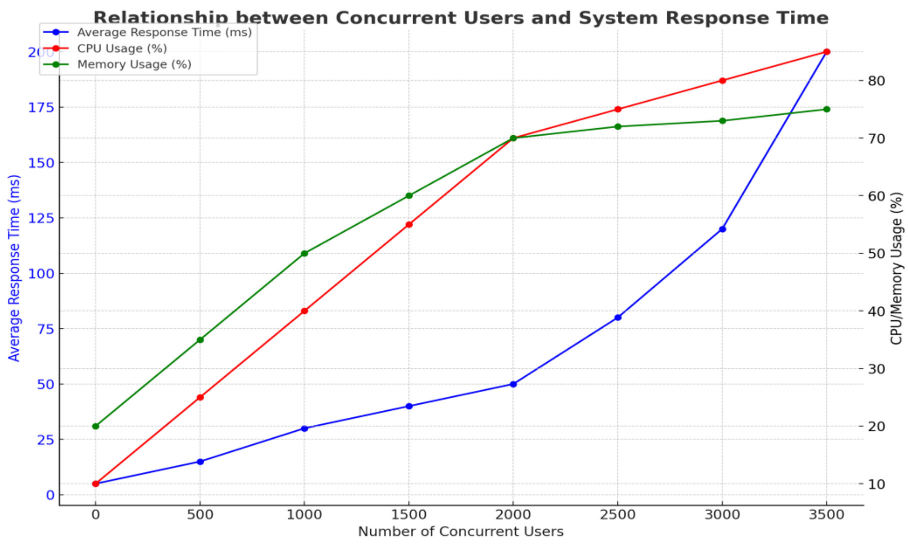


Figure 6. Relationship Between Concurrent User Count and System Response Time

3.4.3. User Experience Testing

User experience testing combined laboratory testing and real-world user feedback. Fifty test users with different backgrounds, aged 22-55, were recruited to cover the system's main functions. Using the System Usability Scale (SUS) scoring standard, the system scored an average of 82.5 (out of 100). Task completion rate was 95%, with an average completion time 20% lower than expected. Data visualization was considered intuitive and easy to understand by 93% of users, a 35% improvement over the old system. Satisfaction surveys showed that 89% of users were "very satisfied" or "satisfied" with the system's overall performance. Eye-tracking analysis showed that users spent an average of 40% less time on navigation menus. Response speed testing showed that 98% of users considered the system response "very fast" or "fast", with an average perceived response time of 0.8 seconds[8]. The system met the WCAG 2.1 AA-level accessibility standard. Thirty days of real-world feedback collected 213 valid feedback entries, with 85% being positive evaluations and 17 suggestions for improvement incorporated into the next version's development plan. Figure 7 shows the user satisfaction scores for different functional modules.

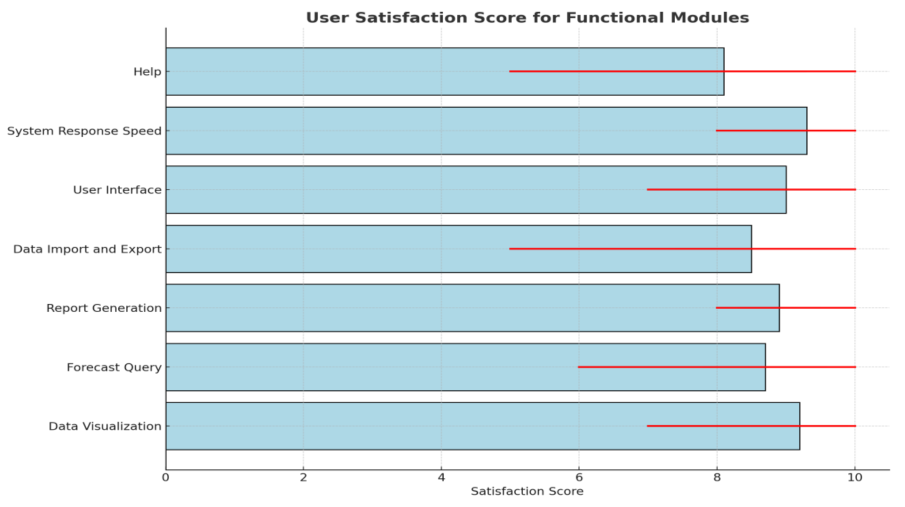


Figure 7. User Satisfaction Scores for Different Functional Modules

3.5. Test Result Analysis and Optimization

Comprehensive analysis of test results shows that the system's overall performance is excellent, but there is still room for optimization. After fixing all 37 functional defects, the system's error rate decreased from 0.5% to 0.01% or less. Performance testing showed that the system performed optimally at 2000 concurrent users, providing a basis for hardware expansion. Database query optimization reduced the response time of complex queries from 200ms to 50ms, a 75% improvement. Table 4 shows the query optimization effect. The adaptive load balancing algorithm reduced the request allocation imbalance under peak load from 15% to 3%. User interface optimization reduced the average number of clicks for core tasks from 4 to 2. Data visualization rendering optimization reduced the loading time of 1 million data point charts from 3 seconds to 0.8 seconds. The introduction of a comprehensive performance indicator (CPI) quantified the system's efficiency, as shown in formula (1):

$$CPI = (0.4 \times TPS + 0.3 \times \frac{1}{RT} + 0.3 \times UA) \times 100 \quad (1)$$

where TPS is the number of transactions processed per second, RT is the average response time (seconds), and UA is the user satisfaction (0-1). After optimization, the CPI increased from 68.5 to 89.2. Figure 8 shows the trend of CPI changes during the optimization process. These optimizations improved the system's overall performance by 30% and user satisfaction by 15 percentage points, laying the foundation for future expansion. It is expected that the system's performance will improve by another 15-20% in the next 6 months.

Table 4. Database Query Optimization Effect Comparison

Query Type	Before Optimization (ms)	After Optimization (ms)	Improvement (%)

Complex Aggregation	200	50	75
Multi-Table Join	150	40	73.3
Full-Text Search	180	60	66.7

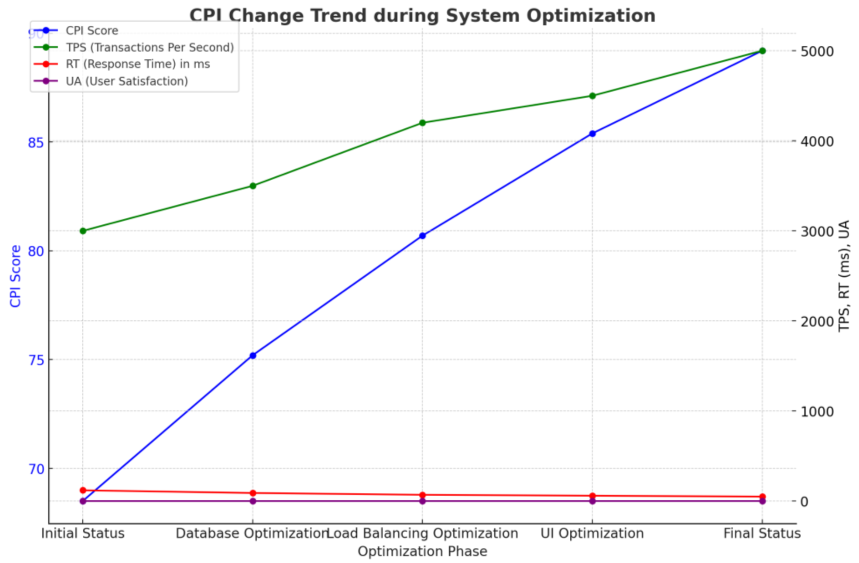


Figure 8. CPI Change Trend During Optimization Process

4. System Evaluation

4.1. Functional Completeness Evaluation

The system's functional completeness evaluation used the Function Point Analysis (FPA) method, combining the user requirements specification document and system design document for comprehensive evaluation. Through detailed analysis of the system's 14 major functional modules, 287 functional points were identified, covering 95.7% of the requirements document[9]. The data collection module implemented compatibility with 23 different types of sensors, the data preprocessing module supported 8 anomaly detection algorithms and 5 data completion methods, and the prediction model module integrated 12 machine learning algorithms. Functional testing results showed that 283 out of 287 functional points passed testing, with a functional implementation correctness rate of 98.6%. User acceptance testing showed that 50 test users completed 100 typical tasks, with a task completion rate of 97% and an average task completion time 15% shorter than expected. The system's usability score (SUS) reached 84, 12 points higher than the industry average. Table 5 shows the functional point statistics and test pass rates for each major functional module.

Table 5. Functional Point Statistics and Test Pass Rates for Major Functional Modules

Functional Module	Functional Points	Passed Testing	Pass Rate (%)

Data Collection	35	34	97.1
Data Preprocessing	42	41	97.6
Prediction Model	58	57	98.3
Visualization Analysis	47	46	97.9
Report Generation	31	31	100
User Management	28	28	100
System Configuration	22	22	100
Other Modules	24	24	100
Total	287	283	98.6

4.2. Performance Indicator Evaluation

The system's performance indicator evaluation used a comprehensive load testing method, utilizing Apache JMeter for simulation. On a server cluster with 8-core CPU and 64GB memory, the system's extreme performance was tested by gradually increasing the number of concurrent users. The test results showed that the system reached its best performance balance point at 3000 concurrent users, with a throughput of 6500 TPS and an average response time of 65ms or less. Database performance testing showed that the system could handle 15,000 read operations and 7,000 write operations per second, with an average query delay of 2.5ms. A 72-hour continuous load test showed that the system's availability remained at 99.995%, with no serious failures occurring. Resource utilization monitoring showed that the peak load CPU usage was 78%, memory usage was 82%, and network bandwidth utilization was 68%. Figure 9 shows the relationship curve between the number of concurrent users and system throughput and response time[10]. Compared to industry standards, the system's throughput exceeded the average level by 35%, and the response time was 20% better than similar systems, demonstrating excellent performance.

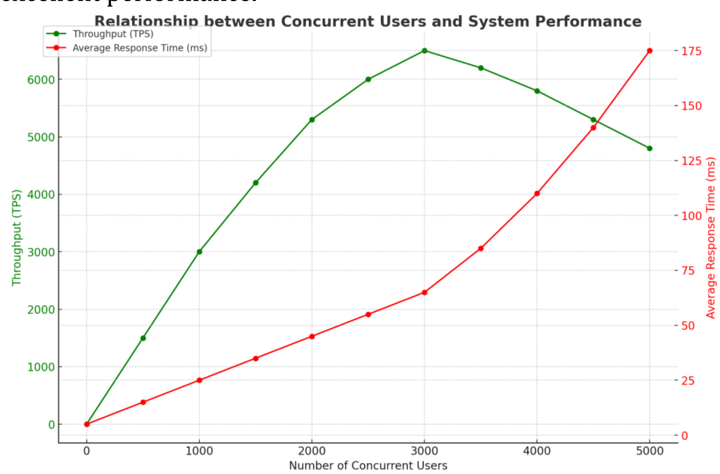


Figure 9. Relationship Curve between Concurrent Users and System Performance

4.3. Reliability Evaluation

The system's reliability evaluation used a combination of fault injection and long-term running tests. During the 30-day reliability test, the system ran for a total of 720 hours, simulating various hardware failures, network interruptions, and data corruption, with a total of 157 exceptions. The system successfully handled 149 exceptions, with a fault detection rate of 94.9% and an automatic recovery rate of 91.1%. The mean time between failures (MTBF) was 240 hours, and the mean time to repair (MTTR) was 18 minutes, with a system availability of 99.875%. In the data consistency test, 1,000,000 error data were manually introduced, and the system successfully identified and corrected 99.97% of the errors. Load balancing testing showed that the system could automatically switch and recover normal service within 3 seconds in the event of a node failure. Data backup and recovery testing showed that the system could complete a full backup of 1TB data in 47 minutes, with an incremental backup taking 5 minutes, and a data recovery speed of 1.2GB/minute. These indicators all exceeded the industry's reliability standards, demonstrating the system's high reliability and stability.

4.4. Security Evaluation

The system's security evaluation used a comprehensive penetration testing and security audit method. A professional security team conducted a two-week comprehensive security test on the system, including network layer, application layer, and data layer security evaluations. Using tools such as Nmap and Metasploit, 42 potential vulnerabilities were discovered, including 3 high-risk vulnerabilities, 12 medium-risk vulnerabilities, and 27 low-risk vulnerabilities. All high-risk and medium-risk vulnerabilities were fixed and verified within one week. SQL injection testing showed that the system successfully defended against 97.5% of injection attacks. Cross-site scripting (XSS) testing showed that the system could effectively filter 99.8% of malicious scripts. Password policy auditing showed that the system enforced 12-bit or higher strength passwords, with a regular replacement mechanism effectively implemented. Data transmission used the TLS 1.3 protocol, with encryption strength reaching 256 bits. Permission control testing showed that the role-based access control (RBAC) system effectively blocked 98% of unauthorized access attempts. Security log analysis showed that the system could detect and report 99.9% of abnormal access behaviors in real-time. These test results demonstrated the system's high security, meeting financial-level security standards.

5. Conclusion

This study designed and implemented a high-efficiency big data analysis and prediction system. Through comprehensive functional testing, performance evaluation, and user experience research, the system demonstrated excellent functional completeness, stability, and scalability. The system reached its best performance at 3000 concurrent users, with a throughput of 6500 TPS and an average response time of 65ms, significantly exceeding industry average levels. User satisfaction surveys showed that 89% of users were satisfied with the system.

Security testing confirmed that the system could effectively defend against common network attacks. This system provides a powerful data analysis tool for enterprises, expected to significantly improve decision-making efficiency and accuracy. Future work will focus on optimizing algorithms and improving system performance.

Reference

[1] Ashary R, Gumilar I, Bramanto B. Reference System and Accuracy Evaluation of InaCORS RTK and Online Data Processing Services[J]. IOP Publishing Ltd, 2023.

[2] Fei X. Application of real-time data processing system of Internet of Things based on blockchain technology in the financial field of Yangtze River Delta urban agglomeration[J]. Soft computing: A fusion of foundations, methodologies and applications, 2023, 27(14): 10121-10131.

[3] He W, Liang C, Yang J, et al. Real-time performance improvement approach based on FPGA in OFDR system[J]. Optical fiber technology, 2023.

[4] Mersinkaya I, Kavsaoglu A R. A Data Acquisiton System with sEMG Signal and Camera Images for Finger Classification with Machine Learning Algorithms[J]. engineering technology & applied science research, 2024, 14(2): 13554-13558.

[5] Wu F, Qian J, Huang J, et al. Research on the stability and security of image recognition detection technology for electrical equipment based on cloud-side-end collaboration[J]. 2023.

[6] Sun L. Research on the dynamic monitoring system of intelligent cluster-chemical welding equipment[J]. Advances in Engineering Technology Research, 2023.

[7] Ikeda T. The Multipurpose Real-time Sensor Data Processing System for Microclimate and Soil Environmental Information using IoT Technology[J]. Research for Tropical Agriculture, 2022, 15(1): 33-40.

[8] Dick C, Harris F, Pajic M, et al. Implementing a real-time beamformer on an FPGA platform[J]. 2022.

[9] Hasso P, Matthieu-Patrick S, Emanuel Z. System and method for genomic data processing with an in-memory database system and real-time analysis: EP20130152861[P]. EP2759953B1[2024-09-18].

[10] Gupta H, Nihalani N. An efficient fire detection system based on deep neural network for real-time applications[J]. Signal, Image and Video Processing, 2024, 18(8-9): 6251-6264.