International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 17 (2025) pp. 1-11

https://doi.org/10.70917/ijcisim-2025-0315
Article

Real-time data processing optimization for industrial IoT enabled by edge computing

Weibing Li [1,*], Haiyan Chen [1] and Yishan Qi [2]

[1] Academic Affairs Office, Beijing Polytechnic University, Beijing, 100176, China; liweibing0756@163.com (W.L.); chenhaiyan@bpu.edu.cn (H.C.)
[2] Party and government office, Beijing Polytechnic University, Beijing, 100176, China; qiyishan@bpu.edu.cn

Abstract: With the rapid development of industrial Internet of Things (IoT) and real-time data processing technologies, edge computing, as an emerging computing paradigm, has garnered significant attention. This paper addresses the challenges of high bandwidth consumption and latency in traditional cloud computing networks by employing multi-source heterogeneous data collection, cloud-edge collaboration, and low-latency data transmission technologies to optimize real-time data processing efficiency. Through performance tests, transmission error rates, and packet loss rates, the experimental results show that the average response time, throughput, and CPU utilization rate are all below 32%. The overall error rate of edge computing is approximately 0.01133%. When the communication distance reaches 150 meters, the packet loss rate is only 0.041%. Therefore, the edge computing framework architecture in this paper demonstrates excellent communication capabilities, low packet loss rates, and high data transmission security.

Keywords: Industrial Internet of Things; real-time data processing; edge computing; multi-source heterogeneous

1. Introduction

Industrial Internet of Things (IIoT) is a significant trend in the current industrial sector. It connects devices, sensors, controllers, computers, and other related equipment within factories and manufacturing ecosystems, forming a vast network that enables the monitoring, identification, and transmission of data [1-4]. Data processing in IIoT requires efficiency, accuracy, and real-time capabilities, and edge computing technology, as an emerging technology, can play a crucial role in achieving these objectives [5-6].

Data processing is a critical component of industrial IoT, requiring the processing, analysis, and storage of data collected from sensors and other devices [7-8]. Given the massive volume of data involved, traditional cloud computing methods are inefficient for handling big data, consume significant bandwidth resources during data transmission, and fail to adequately meet the real-time requirements of industrial production [9-11]. Therefore, the industrial Internet of Things requires new data processing methods, and edge computing is considered a solution that has matured and continued to develop in recent years [12-13]. Edge computing is a computing model that pushes computation and data storage to the network edge, with its primary characteristic being the processing of data at the location closest to the user to reduce data latency [14-16]. Edge computing also pushes computational resources to the device edge, enabling faster data processing and better meeting the real-time requirements of devices [17-18]. Compared to cloud computing, edge computing offers faster processing speeds and higher security. The application of edge computing in industrial IoT primarily manifests in factory intelligence, human-machine collaboration, workshop automation, smart warehousing, and supply chain management [19-22]. By establishing IoT devices and computing terminals at edge nodes, industrial equipment data can be processed more effectively [23-24]. For example, adding edge computing modules to machinery enables real-time data processing and tracking [25]. In logistics operations, edge computing can be used to track goods, monitor warehouse capacity, and prevent inventory errors, helping industrial enterprises improve production efficiency, reduce production costs, enhance customer experience, and generate new revenue streams [26-29].

Reference [30] introduces the application of industrial cyber-physical systems (ICPS), points out the shortcomings of edge computing (EC), and proposes a real-time transmission optimization scheme aimed at accelerating EC. Simulation results validate the effectiveness of the above scheme, which is conducive to the deployment of EC in ICPS. Reference [31] emphasizes the growing attention on smart factories and proposes an adaptive transmission architecture for industrial IoT (IIoT) based on centralized software-defined networking (SDN) and EC. Through evaluation of this scheme, its effectiveness is revealed, providing a better solution for IIoT data transmission. Literature [32] introduces the application of EC in industry, aiming to promote flexible connectivity, real-time control, and data optimization, while supporting smart applications and ensuring security and privacy. Literature [33] describes the positive impact of advancements in the field of mobile edge computing (MEC) on IIoT and proposes an online algorithm named "Energy-Aware Resource Scheduling" (ERS), which demonstrates superior performance compared to other benchmark solutions. Reference [34] develops a conceptual model aimed at facilitating the realization of Industry 5.0, which is based on edge computing support to simultaneously achieve big data protection and optimization, and validates the model's effective application in industry. Reference [35] highlights the immense potential of Industry 4.0 and its innovative technologies, as well as the challenges faced by enterprises, and proposes developing real-time predictive models using small datasets within an EC environment, validating the effectiveness of the proposed model. Literature [36] leverages the concept of the "information age" to study information freshness in the Industrial Internet of Things (IIoT), with results indicating that the proposed method has superiority and significant improvements over benchmark methods. Literature [37] introduces the integration of wireless power transmission (WPT) and MEC to enable wireless power supply for mobile edge computing, and proposes an integrated architecture for wireless power supply mobile edge computing applicable to IIoT, verifying the feasibility of the architecture through case studies. Reference [38] proposes an EC architecture suitable for IoT-based manufacturing, analyzes the role of EC from aspects such as edge devices and network communication, and achieves proactive maintenance through case studies, providing a reference for the deployment of EC in smart factories. Reference [39] examines the impact of EC on real-time data processing and analysis. Based on research into the basic principles of EC, it compares traditional methods and reveals the advantages of EC in real-time data processing. Literature [40] utilizes unmanned aerial vehicles (UAVs) as edge servers to assist in IIoT data processing. Simulation results indicate that this approach can strictly ensure system stability while reducing energy consumption. Literature [41] describes research progress on EC in IIoT, analyzes the opportunities and challenges of EC in IIoT based on 5G edge communication and edge intelligence, and introduces typical application scenarios of EC in IIoT. Reference [42] notes that few studies have explored the importance of EC in Industry 5.0. To address this gap, it investigates the importance of EC in the Industry 5.0 architecture and examines various technologies that can be used to implement and support this new industrial model. Literature [43] proposes an EC-based sensor anomaly detection algorithm. Addressing data issues in IIoT terminal devices, it employs edge technology to optimize sensor data compression and evaluates the effectiveness of the aforementioned methods. The aforementioned studies systematically introduce EC, elucidate its applications and impacts in IIoT, and emphasize that EC effectively promotes real-time data processing and optimization in IIoT.

This paper adopts a current technological development perspective to explain the basic concepts of the Internet of Things (IoT) and edge computing, and proposes a real-time data processing and optimization method for IoT based on edge computing. A four-layer edge intelligence real-time analysis framework is constructed, comprising an application interface layer, an analysis engine layer, a preprocessing layer, and a data access layer. By employing an improved YOLO algorithm, multi-source heterogeneous data collection technology, edge intelligence real-time analysis technology, and edge-cloud collaborative scheduling technology, this framework effectively addresses issues related to data transmission rates and processing efficiency, enabling rapid target detection and real-time data processing. Performance testing was conducted in a test environment to validate the reliability of the methods proposed in this paper.

2. Overview of Edge Computing Environments

2.1. Analysis of the advantages of edge computing in real-time data processing

Edge computing is a distributed computing architecture that shifts computing and data processing capabilities from centralized cloud environments to edge devices or nodes closer to the data source. Its core principle is to leverage the computational resources of edge devices for rapid processing and response at the point of data generation. Compared to traditional cloud computing, edge computing places greater emphasis on real-time processing, low latency, and localized processing, thereby reducing network transmission pressure while enhancing data processing efficiency and system response speed.

Edge computing technology is based on a distributed network architecture, integrating device nodes, edge gateways, and cloud collaboration to form an integrated solution for data collection, processing, and storage. Its characteristics include ensuring rapid data processing while reducing the load on central nodes, and enhancing system robustness and stability through distributed storage and fault-tolerant mechanisms. In an edge computing environment, local nodes possess basic computing, storage, and analysis capabilities, enabling autonomous operation of certain functions. This distributed and autonomous nature grants edge computing broad applicability and scalability in complex scenarios [44].

The primary advantages of edge computing in real-time data processing are improved data processing speed, reduced network resource consumption, and enhanced data security. Since data processing occurs directly at edge nodes close to the data source, edge computing can significantly reduce data transmission latency from devices to the cloud, enabling millisecond-level real-time responses and meeting the high

real-time requirements of industrial production and electrical systems. Edge computing alleviates network bandwidth pressure, reduces data transmission costs and cloud resource consumption when processing large-scale, multi-source data, and optimizes the overall system's resource utilization. In terms of data security, edge computing enhances system privacy protection by processing sensitive information locally, thereby reducing the risk of data leakage during transmission and centralized storage. The edge computing architecture also supports distributed data synchronization and inter-node collaboration, ensuring the continued operation of basic functions even in the event of node failures or network outages.

2.2. Edge Computing Architecture

The edge computing architecture primarily consists of edge devices, edge servers, and the cloud, which work together in a collaborative manner. Edge devices are primarily used for data generation and collection, while edge servers perform preliminary data processing and analysis near the devices. The cloud provides higher-level computing power and larger storage resources to support more complex data processing and analysis tasks. The three components communicate via efficient protocols to enable smooth data transmission and collaborative processing, thereby constructing a flexible and responsive edge computing architecture.

2.3. Data Characteristics in Edge Computing Environments

Data in edge computing environments is characterized by its temporal nature and high frequency. Real-time data is generated by various sensors and devices, such as temperature sensors and cameras. Data is transmitted via wireless communication (e.g., Wi-Fi, Bluetooth) and wired communication. To ensure data accuracy and integrity, it is necessary to effectively handle outliers and duplicate values, and to ensure the authenticity and integrity of data during transmission, in order to support real-time data analysis in edge computing environments [45].

2.4. Application scenarios of real-time data analy*sis in edge computing*

Real-time data analysis has a wide range of applications in edge computing, including smart cities and industrial IoT. In smart cities, real-time data analysis can optimize intelligent traffic management, environmental monitoring, and urban safety. In industrial IoT, real-time data analysis can be used for equipment health monitoring, production process optimization, and predictive maintenance to improve production efficiency. These application scenarios demonstrate the diversity and importance of real-time data analysis in edge computing environments, providing effective technical support for enhancing urban intelligence and industrial production efficiency [46].

2.5. Real-time data analysis algorithms

Real-time data analysis algorithms are a core component of edge computing devices, specifically designed to operate under limited computational resources. The algorithms must be lightweight and efficient to accommodate the device's constraints. To achieve this goal, researchers will conduct in-depth studies on real-time data analysis algorithms suitable for edge computing environments.

Considering the device constraints, researchers will focus on optimizing algorithm structures and reducing computational complexity. These measures include streamlining algorithms to minimize memory usage and computational overhead, ensuring accurate analysis results while meeting real-time requirements. In resource-constrained edge devices, efficient real-time data analysis algorithms not only meet system performance requirements but also enhance the overall response speed and efficiency of the edge computing system. By conducting in-depth research on these algorithms, researchers can provide innovative and feasible solutions for real-time data analysis in edge computing environments.

3. Edge computing optimization strategies for real-time data processing in industrial IoT

3.1. Multi-source heterogeneous data collection technology

The data access layer enables unified access to multiple data sources. For video analysis scenarios, an improved YOLO algorithm is used for object detection. For equipment monitoring, an anomaly detection algorithm with a sliding time window is employed. The analysis engine supports dynamic model updates and optimizes processing efficiency through compression and quantization techniques [47].

Data collection technology in an edge computing environment distinguishes local traffic based on dedicated APNs to implement a traffic diversion mechanism. High-performance gateways configured on edge nodes enable the integration of multi-source heterogeneous data. In project practice, the system deployed data collection gateways supporting multi-protocol conversion, achieving unified data collection from various types of terminals such as smart security cameras, drone monitoring, and industrial IoT. The data collection architecture employs a distributed message queue as a data buffer layer to ensure smooth processing of sudden large-volume data. For real-time data streams, the system uses a priority-based data scheduling algorithm, with the data priority P calculated as follows:

$$P = \alpha \times RT + \beta \times DU + \gamma \times PR \tag{1}$$

Among these, RT represents real-time requirements, DU denotes data utilization value, PR indicates processing resource utilization rate, and α, β and γ are weighting coefficients. This technology employs a dynamic load balancing mechanism to allocate data collection tasks across different edge nodes, enabling a single node to achieve a data throughput capacity of 20 Gbps. The data collection gateway supports protocol adaptation conversion, including industrial Ethernet and mainstream industrial protocols such as Modbus and OPC UA, ensuring seamless integration and preprocessing of heterogeneous data sources.

3.2. Edge Intelligence Real-time Analysis Technology

Edge intelligence real-time analysis technology is used in projects involving video stream analysis and equipment status monitoring. The edge intelligence real-time analysis framework architecture is shown in Figure 1. The system has constructed a four-layer analysis framework, including the application interface layer, analysis engine layer, preprocessing layer, and data access layer. The application interface layer provides external services through an API gateway. The analysis engine layer is based on a microservice architecture design for a distributed inference engine. The preprocessing layer is responsible for data cleaning and filtering.

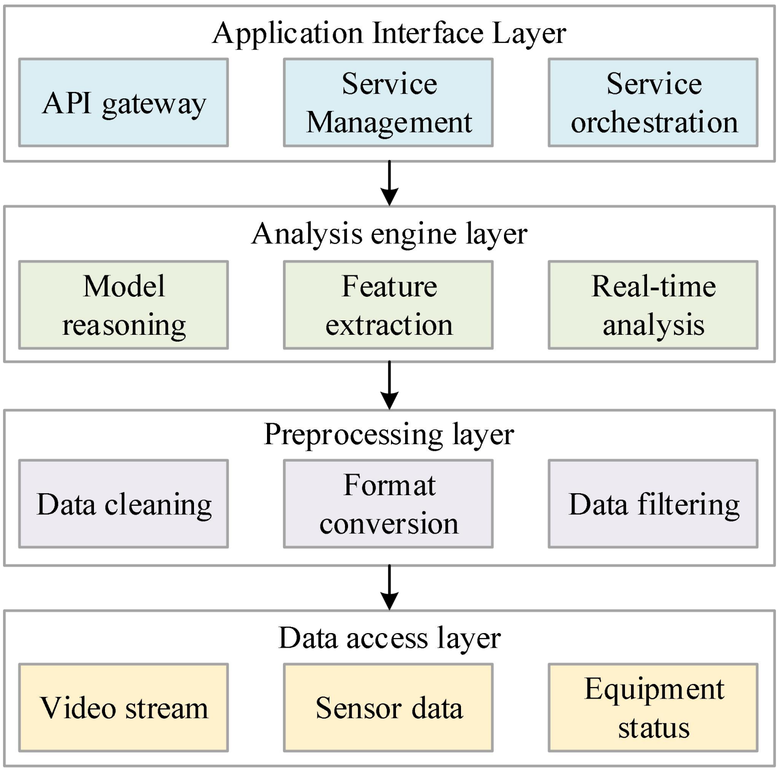

Figure 1. Edge intelligent real-time analysis framework.

3.3. Edge-Cloud Collaborative Scheduling Technology

Edge-cloud collaborative scheduling technology achieves dynamic allocation of computing tasks by constructing a layered resource pool. In project practice, the system uses Docker container technology to achieve elastic scheduling of computing resources and supports collaborative management of edge nodes and cloud resources. The scheduling system adopts a two-layer architecture, with the edge layer responsible for local real-time task processing and the cloud layer responsible for global resource coordination. A task migration evaluation model was designed to address load balancing requirements. The resource scheduling efficiency RE calculation formula is:

$$RE = (WT \times PP + MT \times RP) / (TC + MC) \tag{2}$$

Where WT is the workload transmission volume, PP is the processing performance parameter, MT is the migration time, RP is the resource performance parameter, TC is the transmission cost, and MC is the migration cost. The system implements a threshold-based dynamic scaling mechanism. When the load on the edge node exceeds the preset threshold, task migration or resource expansion is automatically triggered. The collaborative scheduling mechanism classifies and grades tasks to ensure the real-time requirements of critical business, while achieving optimal configuration of computing resources.

3.4. Low-latency data transmission technology

Low-latency data transmission technology is built on 5G networks, achieving end-to-end low-latency data transmission through rational network topology planning and optimized transmission protocols. In the practical application at Zhangjiang Artificial Intelligence Island, the system leverages SDN technology to establish programmable data transmission channels, enabling dynamic optimization of transmission paths. The transmission layer employs an enhanced TCP protocol, with improvements in congestion control algorithms and cache management mechanisms, significantly reducing data transmission latency. The system pre-configures QoS policies for different business types and uses 5G network slicing technology to provide deterministic network services for critical business applications. Intelligent routing gateways are deployed at the network edge, utilizing nearby access and local traffic diversion mechanisms to reduce the number of network hops in data transmission. The transmission system integrates data compression and redundancy elimination technologies to enhance transmission efficiency while ensuring data integrity. For scenarios involving massive small data packet transmission, data aggregation technology is employed to reduce network load, achieving a performance metric of transmission latency below 4.5 milliseconds.

4. Experimentation and performance verification

4.1. Test Environment Setup

To ensure that edge computing technology can operate normally in different environments, testing is an indispensable step. The focus of testing is to verify whether the system can be deployed and run smoothly on servers with different configurations, including different configurations of resources such as CPU, memory, and storage. Additionally, it is necessary to verify the system's performance in different network environments and its ability to handle large data streams. Through testing, the system's performance under high-load conditions, such as response time and processing speed, can be evaluated to ensure that the system can operate stably even under extreme conditions. Key testing items include the system's load capacity, particularly its requirements for CPU and memory, and whether the system can operate stably across different server configurations.

Furthermore, testing of edge computing technology must also focus on its long-term stability and whether the system can handle continuous data streams. This requires testing to not only focus on immediate performance but also observe the system's performance after continuous operation over an extended period. Through such long-term testing, potential issues with the system can be identified and addressed promptly. The test server configurations are shown in Table 1.

In actual testing, automated testing is achieved by deploying CronJobs in a Kubernetes cluster. CronJobs are a resource type in Kubernetes that can execute tasks on a scheduled basis. Through CronJobs, the system's operation in a real-world environment can be simulated, such as simulating continuous data input to verify the system's performance and stability when processing these data streams.

Table 1. Test server configuration table.

Server	IP	Operating system	CPC	memory	Hard disk	Network bandwidth
Server A	139.198.18.137	CentOS 7	3	5GB	500GB	120MB
Server B	139.198.18.138	CentOS 7	3	5GB	500GB	120MB
Server C	139.198.18.139	CentOS 7	3	5GB	500GB	120MB
Server D	172.20.31.3	CentOS 7	5	9GB	100GB	120MB
Server E	172.20.31.4	CentOS 7	3	5GB	500GB	120MB
Server F	172.20.31.5	CentOS 7	3	5GB	500GB	120MB
Server G	192.168.0.4	CentOS 7	2	3GB	200GB	120MB
Server H	192.168.0.9	CentOS 7	1	3GB	200GB	120MB

In edge computing, testing is divided into two main parts: functional verification and performance evaluation. Functional verification aims to confirm the accuracy and reliability of the system's critical operations. Considering that the backend support for edge storage does not provide a frontend user interface, its interaction with edge containers is primarily achieved through coding logic to ensure data persistence. Therefore, this part of the testing primarily relies on writing specific test scripts to simulate various operations via the command-line interface and monitor the logs displayed on the terminal to determine whether the functional modules perform as expected. If errors occur, the relevant exception information will also be exposed through the terminal logs. Functional testing will cover key functional tests such as edge agent components, data distribution and reorganization, resource scheduling, and container drivers. For performance evaluation, considering the dynamic nature of the edge computing environment, testing focuses on verifying the response speed of data persistence when faced with node failures or other unforeseen events. Performance testing will be conducted in a simulated environment, comparing with existing distributed storage solutions such as Sheepdog and Ceph to assess data processing speed and system response time to determine the efficiency of edge computing. The following sections will further detail the testing plans and results for both functional and performance aspects of edge computing.

4.2. Performance Testing

Using the client, design message transmission experiments for the three QoS service quality levels. After each successful connection, immediately send the corresponding QoS level test message. A total of 6,000 messages were sent during the test period. The average time interval from message transmission to reception for each QoS level was calculated, with the results shown in Table 2.

As shown in the table, QoS 0 has the shortest delay because QoS 0 does not require acknowledgment and does not guarantee message delivery, resulting in the shortest time interval. In contrast, messages for QoS 1 and QoS 2 require feedback acknowledgment to ensure message delivery and validity, leading to extended time intervals and increased delay. QoS 2 ensures that each message is received only once, requiring a more complex handshaking mechanism, resulting in the highest delay but also the highest security. The overall message processing delay is less than 1 second, meeting the system communication delay requirements in practical applications.

Table 2. Different qos message delay test table.

Qos level	The average time is (ms)	Standard deviation (ms)
Qos 0	245	11
Qos 1	366	28
Qos 2	552	45

This test simulated 100 users with increasing concurrent requests. To ensure the accuracy of the test results, multiple tests were conducted, and the average value was used as the final test result. The specific test results are shown in Table 3.

Table 3. User concurrent test table.

Concurrent number	100	200	300	400	500	600	700	800	900	1000
Average response time/ms	85	112	124	148	186	225	302	336	370	400
Throughput/TPS	104	168	305	378	487	579	680	762	815	930
Cpve share/%	6.91	8.74	10.56	12.68	14.75	17.46	20.42	22.48	23.46	24.30

As shown in Figure 2, the average response time, throughput, and CPU utilization all increase as the number of concurrent connections increases. However, CPU utilization levels off once it reaches a certain threshold, remaining below 32% and fluctuating within normal ranges. After the number of concurrent connections exceeds 700, the growth rate of average response time flattens out, with response times consistently remaining below 1 second, thereby meeting the concurrent connection requirements for devices in an industrial IoT environment under industrial production conditions. Therefore, the edge computing method designed in this paper demonstrates strong performance and can meet the expected requirements.

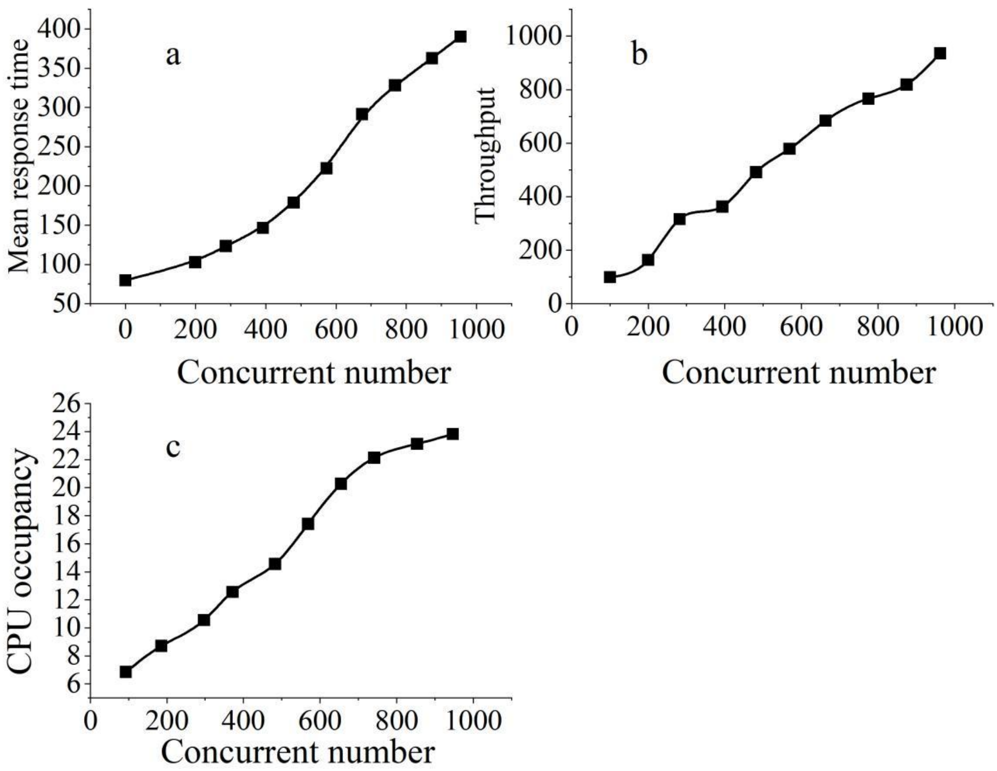

Figure 2. User concurrent test diagram.

4.3. Transmission Error Test

In Ethernet communication, communication quality is affected by transmission errors. The error rate, as a key indicator of data transmission quality, directly impacts the reliability of the transmission channel. Understanding the error rate and transmission time is crucial for evaluating and optimizing Ethernet communication performance. The error rate is typically represented by the bit error rate (BER). In Ethernet communication, signal transmission may be interfered with by factors such as electromagnetic noise and transmission medium loss, which can cause signal distortion or loss, leading to transmission errors. By conducting real-time BER tests through multiple transmissions of different data, the test results are shown in Table 4.

As shown in the table, as the number of bytes increases, the overall transmission time also increases. However, the average BER remains consistently at 0.01133%. This indicates that the communication system can provide reliable data transmission, ensuring high levels of data accuracy and integrity during data processing.

Table 4. Test results.

Test frequency	Send bytes (bit)	Receive bytes (bit)	Average time (ms)	Error rate%
1	1436	1437	0.977	0
2	2059	2060	1.602	0
3	2869	2870	2.211	0
4	3684	3685	2.725	0
5	4478	4478	3.079	0.023
6	5617	5617	3.498	0.02
7	6357	6357	4.09	0.018
8	7484	7484	4.633	0.013
9	8002	8002	5.989	0.028

4.4. Packet Loss Rate Test

By testing the packet loss rate of the system under different communication distances, the packet loss rate detection can verify the stability of the system's data transmission. Through multiple rounds of sending and receiving the same data at different distances to conduct real-time packet loss rate tests, the test results are shown in Table 5.

As shown in the table, when the communication distance is within 100 meters, the number of packets sent and received remains completely consistent, with no packet loss observed. Once the communication distance exceeds 100 meters, a discrepancy arises between the sent and received quantities. Specifically, at a communication distance of 150 meters, the packet loss rate is only 0.041%. Experimental data confirm that the system has excellent communication capabilities, a low packet loss rate, and high data transmission security. In summary, the communication latency in the system described in this paper meets the system's performance requirements.

Table 5. Loss packet rate test.

Test distance (m)	Transmitter (bit)	Receive at (bit)	Loss allowance%
10	2600	2600	0
20	2600	2600	0
30	2600	2600	0
40	2600	2600	0
50	2600	2600	0
60	2600	2600	0
70	2600	2600	0
80	2600	2600	0
90	2600	2600	0
100	2600	2498	0.039
150	2600	2493	0.041

5. Conclusion

Edge computing, as an emerging computing paradigm, has garnered significant attention due to its advantages in low latency, high efficiency, and strong privacy protection when processing large volumes of distributed data. This paper optimizes real-time data processing in industrial IoT systems by integrating improved YOLO algorithms, multi-source heterogeneous data collection techniques, edge intelligence real-time analysis technologies, and edge-cloud collaborative scheduling technologies. Experimental results show that the overall CPU utilization remains below 32% and fluctuates within normal ranges. This meets the system's communication latency requirements in practical applications. The system maintains high levels of data accuracy and integrity throughout the data processing process, thereby fulfilling the system's communication latency demands.

References

1. Sisinni, E., Saifullah, A., Han, S., Jennehag, U., & Gidlund, M. (2018). Industrial internet of things: Challenges, opportunities, and directions. IEEE transactions on industrial informatics, 14(11), 4724-4734.
2. Malik, P. K., Sharma, R., Singh, R., Gehlot, A., Satapathy, S. C., Alnumay, W. S., ... & Nayak, J. (2021). Industrial Internet of Things and its applications in industry 4.0: State of the art. Computer Communications, 166, 125-139.
3. Serror, M., Hack, S., Henze, M., Schuba, M., & Wehrle, K. (2020). Challenges and opportunities in securing the industrial internet of things. IEEE Transactions on Industrial Informatics, 17(5), 2985-2996.
4. Boyes, H., Hallaq, B., Cunningham, J., & Watson, T. (2018). The industrial internet of things (IIoT): An analysis framework. Computers in industry, 101, 1-12.
5. Karmakar, A., Dey, N., Baral, T., Chowdhury, M., & Rehan, M. (2019, March). Industrial internet of things: A review. In 2019 international conference on opto-electronics and applied optics (optronix) (pp. 1-6). IEEE.
6. Nguyen, L. D., & Kortun, A. (2021). Real-time Optimisation for Industrial Internet of Things (IIoT): Overview, Challenges and Opportunities. EAI Endorsed Trans. Ind. Networks Intell. Syst., 7(25), e2.
7. AlSuwaidan, L. (2021). The role of data management in the Industrial Internet of Things. concurrency and computation: Practice and experience, 33(23), e6031.
8. ur Rehman, M. H., Yaqoob, I., Salah, K., Imran, M., Jayaraman, P. P., & Perera, C. (2019). The role of big data analytics in industrial Internet of Things. Future Generation Computer Systems, 99, 247-259.
9. Diène, B., Rodrigues, J. J., Diallo, O., Ndoye, E. H. M., & Korotaev, V. V. (2020). Data management techniques for Internet of Things. Mechanical Systems and Signal Processing, 138, 106564.
10. Babar, M., Khan, F., Iqbal, W., Yahya, A., Arif, F., Tan, Z., & Chuma, J. M. (2018). A secured data management scheme for smart societies in industrial internet of things environment. IEEE Access, 6, 43088-43099.
11. Stergiou, C. L., & Psannis, K. E. (2022). Digital twin intelligent system for industrial internet of things-based big data management and analysis in cloud environments. Virtual Reality & Intelligent Hardware, 4(4), 279-291.

12. Zhao, C., Xiao, C., & Liu, Y. (2021, August). A real-time reconfigurable edge computing system in industrial internet of things based on fpga. In 2021 IEEE 16th Conference on Industrial Electronics and Applications (ICIEA) (pp. 480-485). IEEE.
13. Dou, W., Zhao, X., Yin, X., Wang, H., Luo, Y., & Qi, L. (2020). Edge computing-enabled deep learning for real-time video optimization in IIoT. IEEE Transactions on Industrial Informatics, 17(4), 2842-2851.
14. Sun, Z., Zhang, X., Wang, T., & Wang, Z. (2020). Edge computing in internet of things: a novel sensing-data reconstruction algorithm under intelligent-migratoin stragegy. IEEE Access, 8, 50696-50708.
15. Weibin, S., Yun, L., Yi, D., Yingguo, D., Mingbo, P., & Gang, X. (2019, August). Three-real-time architecture of industrial automation based on edge computing. In 2019 IEEE International conference on smart Internet of Things (SmartIoT) (pp. 372-377). IEEE.
16. Kong, X., Wu, Y., Wang, H., & Xia, F. (2022). Edge computing for internet of everything: A survey. IEEE internet of things journal, 9(23), 23472-23485.
17. Raileanu, S., Borangiu, T., Morariu, O., & Iacob, I. (2018, October). Edge computing in industrial IoT framework for cloud-based manufacturing control. In 2018 22nd International Conference on System Theory, Control and Computing (ICSTCC) (pp. 261-266). IEEE.
18. Ju, X., Gou, R., Xiao, Y., Wang, Z., & Liu, S. (2022). The use of edge computing-based internet of things big data in the design of power intelligent management and control platform. International journal of grid and utility computing, 13(1), 76-86.
19. Galanopoulos, A., Salonidis, T., & Iosifidis, G. (2020). Cooperative edge computing of data analytics for the Internet of Things. IEEE Transactions on Cognitive Communications and Networking, 6(4), 1166-1179.
20. Deng, X., Yin, J., Guan, P., Xiong, N. N., Zhang, L., & Mumtaz, S. (2021). Intelligent delay-aware partial computing task offloading for multiuser industrial Internet of Things through edge computing. IEEE Internet of Things Journal, 10(4), 2954-2966.
21. Zhang, X., Wang, M., Zhu, X., Yan, Z., & Geng, G. (2025). Collaborative Edge-Cloud Data Transfer Optimization for Industrial Internet of Things. IEEE Transactions on Parallel and Distributed Systems.
22. Liang, F., Yu, W., Liu, X., Griffith, D., & Golmie, N. (2020). Toward edge-based deep learning in industrial Internet of Things. IEEE Internet of Things Journal, 7(5), 4329-4341.
23. Quy, N. M., Ngoc, L. A., Ban, N. T., Hau, N. V., & Quy, V. K. (2023). Edge computing for real-time Internet of Things applications: Future internet revolution. Wireless Personal Communications, 132(2), 1423-1452.
24. Cui, L., Xu, C., Yang, S., Huang, J. Z., Li, J., Wang, X., ... & Lu, N. (2018). Joint optimization of energy consumption and latency in mobile edge computing for Internet of Things. IEEE Internet of Things Journal, 6(3), 4791-4803.
25. Kaur, K., Garg, S., Aujla, G. S., Kumar, N., Rodrigues, J. J., & Guizani, M. (2018). Edge computing in the industrial internet of things environment: Software-defined-networks-based edge-cloud interplay. IEEE communications magazine, 56(2), 44-51.
26. Papageorgiou, A., Cheng, B., & Kovacs, E. (2015, November). Real-time data reduction at the network edge of Internet-of-Things systems. In 2015 11th international conference on network and service management (CNSM) (pp. 284-291). IEEE.
27. Yun, D. W., & Lee, W. C. (2021). Intelligent dynamic real-time spectrum resource management for industrial IoT in edge computing. Sensors, 21(23), 7902.
28. Huang, J., Zhou, S., Li, G., & Shen, Q. (2025). Real-time monitoring and optimization methods for user-side energy management based on edge computing. Scientific Reports, 15(1), 24890.
29. Stadnicka, D., Sęp, J., Amadio, R., Mazzei, D., Tyrovolas, M., Stylios, C., ... & Navarro, J. (2022). Industrial needs in the fields of artificial intelligence, Internet of Things and edge computing. Sensors, 22(12), 4501.
30. Peng, Y., Jolfaei, A., Hua, Q., Shang, W. L., & Yu, K. (2022). Real-time transmission optimization for edge computing in industrial cyber-physical systems. IEEE Transactions on Industrial Informatics, 18(12), 9292-9301.
31. Li, X., Li, D., Wan, J., Liu, C., & Imran, M. (2018). Adaptive transmission optimization in SDN-based industrial Internet of Things with edge computing. IEEE Internet of Things Journal, 5(3), 1351-1360.
32. Dai, W., Nishi, H., Vyatkin, V., Huang, V., Shi, Y., & Guan, X. (2019). Industrial edge computing: Enabling embedded intelligence. IEEE Industrial Electronics Magazine, 13(4), 48-56.
33. Wu, H., Lyu, X., & Tian, H. (2019). Online optimization of wireless powered mobile-edge computing for heterogeneous industrial internet of things. IEEE Internet of Things Journal, 6(6), 9880-9892.
34. Bajic, B., Suzic, N., Moraca, S., Stefanović, M., Jovicic, M., & Rikalovic, A. (2023). Edge computing data optimization for smart quality management: Industry 5.0 perspective. Sustainability, 15(7), 6032.
35. Bajic, B., Suzic, N., Simeunovic, N., Moraca, S., & Rikalovic, A. (2020). Real-time data analytics edge computing application for industry 4.0: The mahalanobis-taguchi approach. International Journal of Industrial Engineering and Management, 11(3), 146-156.
36. Li, J., Tang, J., & Liu, Z. (2022). On the data freshness for industrial Internet of Things with mobile-edge computing. IEEE Internet of Things Journal, 9(15), 13542-13554.
37. Wu, H., Tian, H., Nie, G., & Zhao, P. (2020). Wireless powered mobile edge computing for industrial internet of things systems. IEEE Access, 8, 101539-101549.
38. Chen, B., Wan, J., Celesti, A., Li, D., Abbas, H., & Zhang, Q. (2018). Edge computing in IoT-based manufacturing. IEEE Communications Magazine, 56(9), 103-109.
39. Modupe, O. T., Otitoola, A. A., Oladapo, O. J., Abiona, O. O., Oyeniran, O. C., Adewusi, A. O., ... & Obijuru, A. (2024). Reviewing the transformational impact of edge computing on real-time data processing and analytics. Computer Science & IT Research Journal, 5(3), 693-702.

40. Tang, X., Zhang, H., Zhang, R., Zhou, D., Zhang, Y., & Han, Z. (2023). Robust trajectory and offloading for energy-efficient UAV edge computing in industrial internet of things. IEEE Transactions on Industrial Informatics, 20(1), 38-49.
41. Qiu, T., Chi, J., Zhou, X., Ning, Z., Atiquzzaman, M., & Wu, D. O. (2020). Edge computing in industrial internet of things: Architecture, advances and challenges. IEEE communications surveys & tutorials, 22(4), 2462-2488.
42. Sharma, M., Tomar, A., & Hazra, A. (2024). Edge computing for industry 5.0: Fundamental, applications, and research challenges. IEEE Internet of Things Journal, 11(11), 19070-19093.
43. Kong, D., Liu, D., Zhang, L., He, L., Shi, Q., & Ma, X. (2020). Sensor anomaly detection in the industrial internet of things based on edge computing. Turkish Journal of Electrical Engineering and Computer Sciences, 28(1), 331-346.
44. Yixuan Fang,Zhufang Kuang,Haobin Wang,Siyu Lin & Anfeng Liu. (2025). Minimizing energy consumption of collaborative deployment and task offloading in two-tier UAV edge computing networks. Journal of Systems Architecture,167,103511-103511.
45. Yumi Kim,Bokyeong Kim,Taewon Song & Haneul Ko. (2026). Neighbor-aware shared container instance warming framework for serverless edge computing. Future Generation Computer Systems,174,107986-107986.
46. Seoyeon Kim,Jinsung Cho,Jiwoo Shin,Bongjae Kim & Jinman Jung. (2026). Automatic generation of spiking neural networks on neuromorphic computing hardware for IoT edge computing. Future Generation Computer Systems,174,107953-107953.
47. Liang Ma,Qikai Yang & Kaixiang Peng. (2025). A unified representation and fusion framework of multi-source heterogeneous data for fault diagnosis in industrial processes. Advanced Engineering Informatics,67,103539-103539.