ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

Real-Time Data Processing and Security Protection Mechanisms for IoT Devices in an Intelligent Edge Computing Architecture

Bingfu Hu[1] *and Xuewei Liu*[2,*]

[1] Department of Information Engineeringt, Weifang Engineering Vocational College, Weifang 262500, Shandong, China
[2] Science and Technology Division, Weifang Engineering Vocational College, Weifang 262500, Shandong, China

SUMMARY: *This paper addresses an intelligent edge computing structure for handling the rapid processing of data and security of IoT devices. This paper seeks to analyze the performance of edge computing integrated with real-time data acquisition and preprocessing algorithms that build on processing and security tasks, scheduling and balancing computing load to offer multi-level security, and evaluate the applicable combined security regarding multiple situations. Enhancing computing productivity and constructing additional security systems for edge computing lead to rapid results from data processing. Furthermore, edge computing with security systems and the protection of individuals and data offer a relentless and safe solution for large scale IoT systems.*

KEYWORDS: *Intelligent edge computing; Internet of Things; Real-time data processing; Security protection; Privacy protection*

1 Introduction

The growing number of IoT devices translates to a higher volume of data that must be handled with greater surgical precision and security in real time. Classic processing models that lean on a cloud-based approach are weak in low latency, high volume, and high privacy demand scenarios. The intelligent edge approach remedies these shortcomings by deploying edge nodes close to the devices to perform local data processing and real-time analysis and enact a seamless edge-cloud collaboration. The paper offers a systematic analysis of data collection and storage, edge real-time processing algorithms, real-time load balancing, task scheduling, and security. Through a practical analysis of each of the listed components, the author seeks to verify practical application in security, effectiveness, and optimization, and offers a theoretical underpinning from an edge-based IoT processor viewpoint.

2 Related Work

2.1 IoT Device Data Processing Technologies

The constant stream of data produced by IoT devices is voluminous, diverse, and of variable quality, thus creating challenges for data processing technologies. This process can be divided into three different operations: data collection, data preprocessing, and data processing. The

*qzlxw1234@gmail.com
https://doi.org/10.65102/is2026975
Published: 30 April, 2026

collection of data is like this:

$$D_t = \{d_1, d_2, ..., d_n\} \tag{1}$$

where D_t represents the n data points collected at time t ,and d_i denotes the data from a single sensor.Data preprocessing primarily includes anomaly detection, denoising, and normalization, with the normalization process expressed by the formula:

$$d_i^{morm} = \frac{d_i - d_{\min}}{d_{\max} - d_{\min}} \tag{2}$$

where $d_{\min}$ and $d_{\max}$ represent the minimum and maximum values in the dataset, respectively, and d_i^{norm} denotes the normalized data. In the edge analysis phase, the focus is primarily on the data that has been subjected to some preprocessing steps. The data is pre-processed, and the aggregation of the data at the edge is executed. This is useful for narrowing the gap between the processing capacity of the cloud and the demand for processing. Also, transmission delays are alleviated. All of the above methods can dramatically boost your IoT systems to provide the edge of smart analysis and decision-making.

2.2 Edge Computing Architecture and Applications

Between IoT devices and edge nodes, local processing is achieved through edge nodes. This is done by processing data that has already been captured IoT devices. This is both beneficial to the user and the business.Let the data stream generated by an IoT device be denoted as $D = \{d_1, d_2, ..., d_n\}$ The processing time for the data at the edge node can be expressed as:

$$T_{edge} = \frac{\sum_{i=1}^{n} C_i}{R_e} \tag{3}$$

where C_i represents the computational load of the iii-th data item, and R_e represents the computational capacity of the edge node. If the task requires uploading to the cloud for further analysis, the total latency can be expressed as:

$$T_{total} = T_{edge} + T_{trans} + T_{cloud} \tag{4}$$

where T_{trans} represents the data transmission delay, and T_{cloud} represents the cloud processing time.Edge computing architectures typically adopt a three-tier distributed design: the IoT device layer handles data collection, the edge node layer performs real-time processing and preliminary analysis, and the cloud layer handles complex computations and historical data storage. Through edge-cloud collaborative processing, not only can immediate responses to data streams be achieved, but resource scheduling and load balancing can also be optimized, thereby improving the efficiency and security of the entire IoT system (see Figure 1: Hierarchical Edge Computing Architecture).

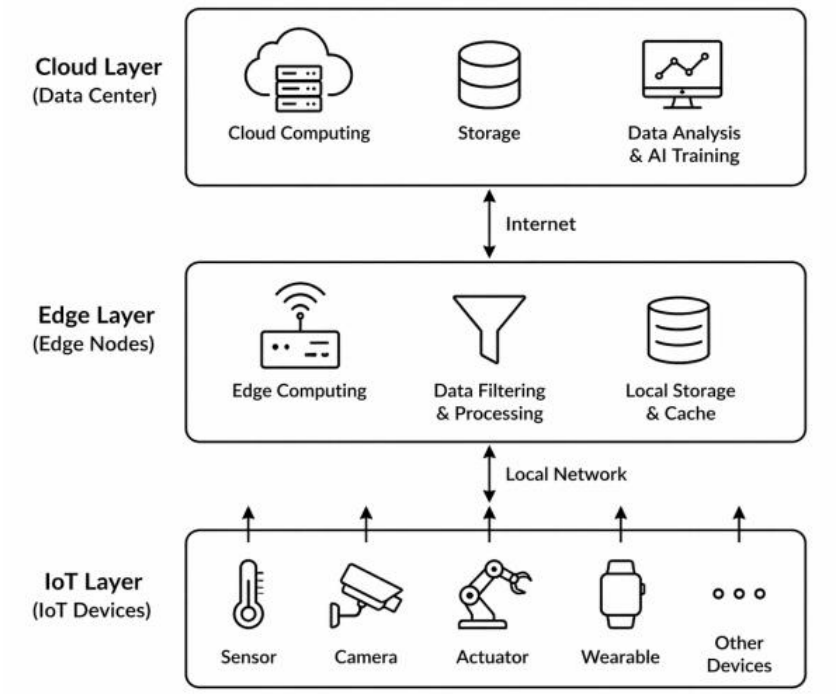

Figure 1: Hierarchical Edge Computing Architecture

2.3 Research Progress on IoT Security Protection Mechanisms

The growth of smart devices brought on the need for secure software to implement across technology that protects devices as boundaries to data related to devices and data security. Control mechanisms for devices and security software(SS) include intrusion detection as well as use of measures that afford privacy. Protection of data security through the use of Secure Sockets Start Layer (SSL) along with Protocols for Layered Adjustments (VPN) are just some of the accessible resources that secure data. Proprietary access measures focus on the use of assignment and role related measures to protect access and control privileges of those devices. Anomalous behavior and intrusion detection represent the evolving challenges of security. Many use machine and deep learning to propose and implement real time detection of the security measures that would provide the highest diagram. Privacy solutions are improved through the use of differential privacy and data processing; automated enforcement of security makes it possible to provide fast access to valuable information. Protection of privacy has challenges and those include malleable and shared privacy and the elusive balance between collaboration and related security in multiple node systems.IoT systems will use intelligent analysis, dynamic policy adjustment, and layered protection for safe and dependable large scale IoT systems.

2.4 Shortcomings and Challenges of Existing Research

While significant improvements have been made in recent years in terms of IoT data processing and edge computing, several drawbacks and issues remain. For instance, the real-time data processing algorithms available today face severe performance bottlenecks in large-scale IoT applications. Edge nodes are characterized by limited computational resources, and as data traffic and processing demand rise, a corresponding increase in latency is observed, which negatively impacts the responsiveness of the system. Further, collaboration among edge nodes and coordination between the edges and the cloud are still inadequate, as resource allocation and scheduling towards balancing the workload is still an unsolved problem, which in turn

causes resource wastage and overload. Although encryption, access control, intrusion detection, and privacy protection technologies have evolved and developed, issues of high resource consumption, poor adaptability, and collaboration complexity across nodes remain. Therefore, the issues of the edge computing environment, with widely distributed and exposed nodes prone to direct and physical attacks, and the urgent need for a timely and effective defense, are still extremely challenging. In addition, there remain issues pertaining to the dilemma between data loss prevention and information sharing, which can lead to significant data loss. Furthermore, many of the studies conducted in this area are application-centric. Therefore, the challenges involving diversified IoT systems and meeting sophisticated demands to a large extent remain. Future studies must focus on the strategies and approaches for implementation of light and effective algorithms, collaboration, and multi-layer security systems, and dabble in the equilibrium between privacy preservation and information sharing. Therefore, to a large extent, system performance and security can be assured.

3 System Architecture Design

3.1 Overview of the Intelligent Edge Computing Architecture

The intelligent edge computing architecture includes many features. Data processing is done in a decentralized, multi-tiered architecture made up of three layers. The first portion is a cloud tier, followed by an edge node compression layer, and finally a layer that houses computing and analytical capabilities. The design architecture handles processing of data in real time, data storage (both historical and current), and compression and aggregation. Balancing collaborative processing between edge and cloud resources in real time is also available. Robustness and performance is seen when edge node hosting Artificial Intelligence (AI) analytics modules engage in real time predictive system analytics and decision making. Load balancing and task scheduling is also a safe, high-performance way to protect data processing. Incorporating adaptive analytical capabilities enhances the system and augments the performance of the architecture. The architecture balances resource and data processing with security protection. The performance of the architecture augments large scale system design, refactoring for adaptive, real time, unitary user driven predictive analyses.

3.2 Design of the Data Acquisition and Preprocessing Module

In the intelligent edge computing architecture, the data acquisition and pre - processing module is a key component, making sure of the quality of IoT data and the processing efficiency of edge nodes (refer to Figure 2 Data Acquisition and Preprocessing Flowchart). Let the raw data sequence collected by IoT devices be $X = \{x_1, x_2, ..., x_n\}$. The data normalization process can be expressed as:

$$\hat{x}_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{5}$$

where $\hat{x}_i$ is the normalized data, $x_{\min}$ and $x_{\max}$ are the minimum and maximum values of the sequence, respectively. After normalization, the data can undergo sliding window feature extraction, and the window average can be expressed as:

$$\overline{x}_t = \frac{1}{w} \sum_{i=1-w+1}^{t} x_i \tag{6}$$

where $\bar{x}_t$ is the window average at time t ,and w is the sliding window length. The module workflow includes data acquisition, cleaning, normalization, feature extraction, compression, and caching, ensuring that edge nodes can respond quickly to real-time processing tasks and providing a high-quality data foundation for subsequent task scheduling and security analysis.

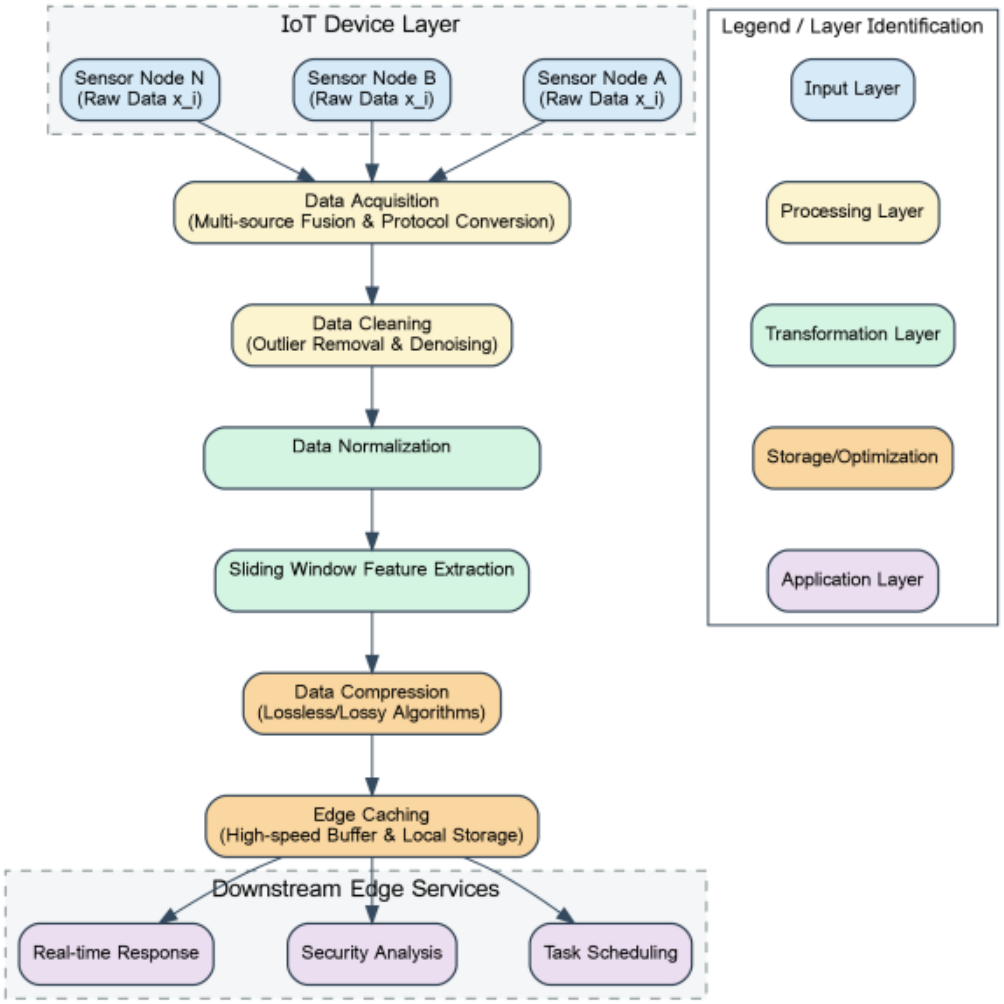

Figure 2: Data Acquisition and Preprocessing Flowchart,

3.3 Collaboration Mechanism Between Edge Computing Nodes and the Cloud

The collaboration mechanism between edge computing nodes and the cloud achieves efficient data processing and system performance optimization through reasonable task allocation. Let the task set be $T = \{T_1, T_2, ...T_m\}$, where each task T_i can be executed at an edge node or in the cloud. The processing latency at an edge node can be expressed as:

$$L_{edge}(T_i) = \frac{C_i}{R_e} + T_{queue} \tag{7}$$

where C_i represents the computational load of the task, R_e represents the computational capacity of the edge node, and T_{queue} represents the queue waiting time. If the task is selected for processing in the cloud, the total delay is:

$$L_{total}\left(T_i\right) = L_{edge}\left(T_i\right) + T_{trans} + \frac{C_i}{R_c} \tag{8}$$

where T_{trans} represents data transmission delay, and R_c represents cloud computing capacity.

The coordination mechanism achieves load balancing and resource optimization between the edge and the cloud by means of dynamic task allocation. It places emphasis on delay - sensitive tasks at edge nodes and transmits computation - intensive or historical data analysis tasks to the cloud, so as to ensure the system's real - time responsiveness and processing efficiency.

3.4 Security Protection Module Design

The security protection module is a basic and mainly essential component in the intelligent edge computing architecture, which has an important influence on the safe operation of data produced by IoT devices, edge node and whole system (as shown in Figure 3). In this module, it follows a multi-level and distributed security model which secures up to the IoT device layer, edge node layer, and cloud layer which in turn gives end-to-end whole ecosystem protection.To ensure that the data stays intact, confidential, and free from interception or tampering en route to its destination, the end-to-end encryption methods as well as secure communication protocols are written into the software. Role-based access control (RBAC) or attribute-based access control (ABAC) policies allow restrictions for device and user permissions, mitigates unwanted operations.Finally, the module also combines sophisticated intrusion detection mechanisms and abnormal behavior analysis using rule-based engines and AI-driven models to continuously monitor real-time data streams. This helps in identifying anomalous functioning, security loopholes or cyberattacks on time. As for privacy protection, models bolstered by means of differential privacy, data anonymization and secure encrypted computation help to ensure that sensitive user information is protected in the processing of all data at both edge and cloud levels.Its modular and layered design ensures that each security function can be independently upgraded or tuned without disrupting other processes, while still working seamlessly with real-time data processing. In general, this new security protection module helps build a smart and multi-level end-to-end security framework, which will provide secured operations (resilience), high throughput (scalability) and integrity (robustness) to the large-scale IoT systems while strengthening everyone’s trust in the system performance and data security.

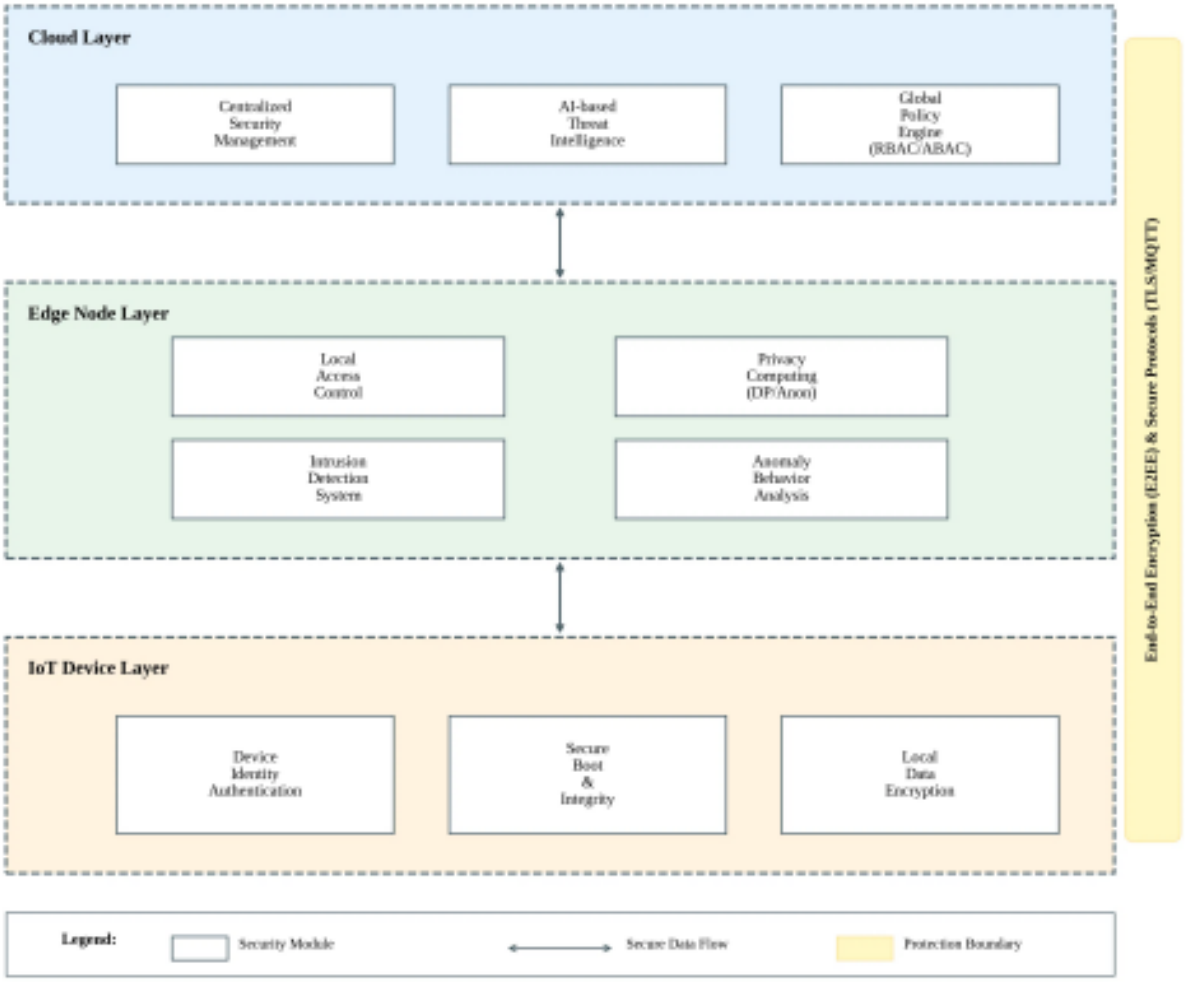

Figure 3: Multi-layer Security Protection Architecture

4 Real-time Data Processing Mechanism

4.1 Data Flow Models and Analysis of Real-Time Processing Requirements

In intelligent edge computing architectures, data transmission within IoT systems mainly follows two primary modes: continuous data streams and event-driven data streams. Continuous streams refer to the steady transmission of sensor data and telemetry information, while event-driven streams are generated by sporadic triggering events, including system alarms and environmental condition changes. Such data flows are typically featured by high transmission frequency, stringent low-latency demands and sudden traffic fluctuations, posing considerable challenges to real-time data processing.In order to achieve efficient stream management, it is of great importance to establish a well - defined and standardized data flow model. This model clearly specifies the full - fledged workflow of data collection, transmission, pre - treatment, edge - side analysis, and cloud - end processing, together with the corresponding latency constraints and performance indicators. IoT data streams span a large variety of data formats, including structured sensor measurements, semi - structured log files, and unstructured multimedia content like video and audio streams. Each data category makes unique demands on real - time processing and scheduling mechanisms. Key requirements for real-time

processing include the ability to control latency and maintain throughput, accuracy, and integrity. Such tasks include sensor feedback or emergency notifications and are best processed at the edge nodes. Computationally intensive analyses and tasks such as historical data aggregation and deep learning are routed to the cloud. A data flow model allows the design of edge-oriented real-time processing algorithms, efficient task scheduling, and resource allocation. When combined with responsive real-time processing at the edge nodes, this ensures high-performance large scale IoT systems even in highly dynamic and turbulent environments.

4.2 Design of Real-Time Processing Algorithms for Edge Computing

Within the intelligent edge computing architecture, real-time data processing algorithms are designed to efficiently analyze continuous data streams generated by IoT devices while ensuring low latency and high throughput. Let the data stream received by the edge node be $D = \{d_1, d_2, ..., d_n\}$ .The processing time for each data item at the edge node can be expressed as:

$$T_i = \frac{C_i}{R_e} \tag{9}$$

where T_i is the time required to process the th data item (i), C_i is the computational load of the data, and R_e is the computational capacity of the edge node. To improve real-time performance, a sliding window aggregation algorithm is commonly used to perform statistical computations on the data stream within the window w:

$$d_t = \frac{1}{w} \sum_{i=1-w+1}^{t} d_i \tag{10}$$

where $\bar{d}_t$ is the window average at time t , and w is the window length. Real-time edge computing algorithms typically integrate lightweight AI inference, anomaly detection, and data compression strategies. This approach ensures low-latency processing for critical tasks while reducing the computational burden on the cloud, enabling collaborative optimization between the edge and the cloud to enhance the overall processing efficiency and stability of the IoT system.

4.3 Task Scheduling and Load Balancing Strategies

In edge computing environments, to ensure real-time processing capabilities and system stability, it is necessary to design efficient task scheduling and load balancing strategies. Let the set of edge nodes be $N = \{n_1, n_2, ..., n_m\}$ and the set of tasks be $T = \{T_1, T_2, ..., T_k\}$. The total load on node n_j can be expressed as:

$$L_j = \sum_{i \in T_j} \frac{C_i}{R_j} \tag{11}$$

where L_j is the total load of node n_j , C_i is the computational effort of task T_i ,and R_j is the computational capacity of node n_j . To achieve load balancing, a minimum-load-first allocation strategy can be adopted, assigning tasks to the node with the lowest load:

$$n^* = \arg\min_{n_j \in N} L_j \tag{12}$$

where n^* is the optimal node selected to execute the task. This strategy, combined with dynamic task allocation and edge-cloud collaboration mechanisms, improves resource utilization and processing throughput in multi-node systems while ensuring the real-time performance of latency-sensitive tasks, thereby optimizing the overall processing performance of IoT data streams in edge computing environments.

5 Security Protection Mechanisms

5.1 Data Transmission Security Strategy

IoT data faces various security threats during transmission between devices, edge nodes, and the cloud, such as eavesdropping, tampering, and replay attacks. To ensure data integrity and confidentiality, an end-to-end encryption strategy must be adopted. Let the original data be D; the encrypted data is represented as:

$$E(D) = Enc(D, K) \tag{13}$$

where $Enc(\cdot)$ is the encryption function, K is the symmetric or asymmetric key, and $E(D)$ is the encrypted data. During transmission, data integrity can be verified using a Message Authentication Code (MAC), calculated as follows:

$$MAC = H(D \,||\, K_h) \tag{14}$$

where $H(\cdot)$ is the hash function, K_h is the MAC key, and $||$ denotes the concatenation operation. By verifying whether the MAC value at the receiving end matches that of the sending end, it is possible to detect whether the data has been tampered with during transmission. Additionally, by combining encrypted tunnels (such as TLS/SSL) with lightweight encryption algorithms, security can be ensured while reducing the computational burden on edge nodes.This comprehensive strategy supports the secure transmission of real-time data streams, enabling low-latency, highly reliable data communication, and providing a solid foundation for the security of IoT systems.

5.2 Edge Node Access Control and Authentication

As a fundamental component of IoT data processing and preliminary analysis, the security of edge nodes plays a critical role in ensuring the reliability and integrity of the entire intelligent edge computing system. Edge nodes serve as intermediaries between IoT devices and cloud resources, making them prime targets for potential security breaches. Therefore, robust access control and authentication mechanisms are essential to prevent unauthorized devices and users from accessing sensitive edge resources, and to guarantee the secure execution of both data processing and computational tasks. The most common access control strategies are Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC), which focus on control from opposite ends of the spectrum: centralized versus decentralizing control. RBAC facilitates the centralization of user and device privileges by assigning permissions to predefined user roles, whereas ABAC offers the decentralization of control and greater flexibility, especially in complex scenarios common to the Internet of Things (IoT), by adapting

end-user permissions to multiple conditions that can be simplified to four categories: environmental (i.e., the situation or problem) context, user and device properties, and situational attributes. The importance of authentication mechanisms is to ensure that the user and the end stations are legitimate, and these three (multi-factor authentication, digital certificates, and several secure key exchange protocols) approaches guarantee that communication between the nodes, devices, and users is secure and trusted. The use of several other advanced solutions, including blockchain and other distributed identity management systems, can provide secure, transparent, and resilient decentralized access systems. RBAC and ABAC solutions decouple from real-time data, enabling them to be enforced at the operation level without slowing the multi-node operation. Combined, these mechanisms enable edge nodes to provide controlled access to disparate systems securely, reliably, and at scale. The modular nature assures that the greater complexity of the system, the more focus the secure operation of the system itself.

5.3 Intrusion Detection and Anomaly Behavior Analysis

Analyzing anomalous behavior and intrusion detection are crucial for protecting IoT systems in edge computing architectures (see Figure 6). This module is built for monitoring and analyzing the behavior of edge nodes and IoT devices. Its capabilities allow for the early identification of attacks and the detection of anomalous traffic and anomalous behavior of devices. Detection methods are executed using either rule-based detection techniques or intelligent analysis techniques based on machine learning or deep learning. Rule-based techniques identify known attacks using defined thresholds and policies. Intelligent analysis methods are focused on detecting unknown attacks or anomalous events by using behavioral characteristics and pattern recognition technologies. In distributed edge environments, intrusion detection typically combines local analysis at edge nodes with centralized analysis in the cloud, resulting in a collaborative response mechanism. Anomalous events are logged, categorized, and alerts are triggered, while supporting policy adjustment and rapid protection. Through this mechanism, IoT systems can achieve efficient monitoring of various intrusion behaviors and anomalous events while ensuring real-time data processing performance, therebyproviding reliable security assurance for collaboration between edge nodes and the cloud.

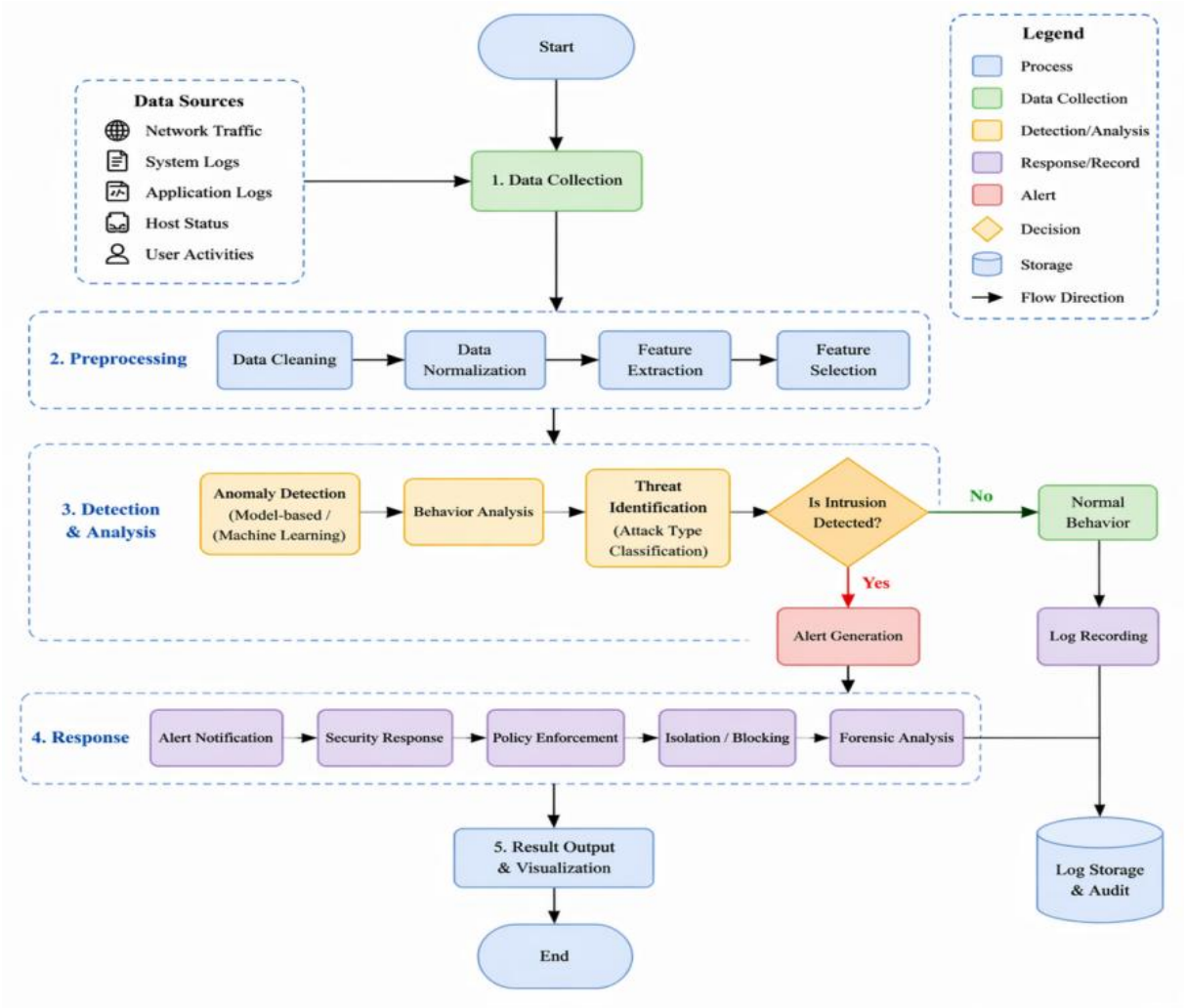

Figure 4: Intrusion Detection Process,

5.4 Application of Privacy Protection and Encryption Technologies

Privacy protection and encryption technologies are crucial means of ensuring data security. Differential privacy is a commonly used technique whose core principle involves adding noise to query results to ensure that the privacy of individual records is not compromised. Let the query function be ` $f(D)$ `; the result after adding Laplace noise is:

$$f'(D) = f(D) + Lap\left(\frac{\Delta f}{\in}\right) \tag{15}$$

where Δf is the global sensitivity of the sensitive function, $\in$ is the privacy budget, and $Lap(\cdot)$ represents theLaplacenoise.Additionally, homomorphic encryption (Homomorphic Encryption) technology can be employed during data transmission and storage to enable computations on ciphertext without decryption.Let the ciphertext be $E(m)$; the additive homomorphic property can be expressed as:

$$E(m_1) \oplus E(m_2) = E(m_1 + m_2) \tag{16}$$

where $\oplus$ denotes the addition operation on ciphertext, and m_1, m_2 is the plaintext data. By combining differential privacy with homomorphic encryption, edge nodes can perform real-time data analysis and model training while protecting user privacy, effectively safeguarding sensitive information in IoT systems and supporting edge-cloud collaborative computing.

6 Experimental Results and Analysis

6.1 Experimental Environment Configuration

To validate the effectiveness of real-time IoT data processing and security protection mechanisms under the intelligent edge computing architecture, this study established a comprehensive experimental environment. On the hardware side, the setup includes multiple edge computing nodes and IoT devices. The nodes provide the option of high-performance CPUs, as well as support for high-speed data processing and AI inferences with caching. Local storage allows for the processing of data closer to the node. IoT devices allow for all types of data collection through the use of sensors, smart terminals, and various monitoring devices. Within the Cloud servers, resources of advanced high-performance computing and distributed storage are used for the most complex computing, Global Coordination, and data analysis. From a software perspective, edge nodes are used to run real-time data processing middleware and lightweight AI inference to support the rapid analysis of data. For the efficient transfer of data, the IoT devices use a range of standard communication protocols (like MQTT, CoAP, and HTTP/HTTPS) to connect to the edge nodes. The experiment included security module integration to test varied scenarios and the combination of communication encryption, access control, intrusion detection, privacy protection, and the application system in real-world scenarios. The edge computing experiment allows for real-time processing and provides a means of task evaluation, while security and protection functionality can be assessed to validate the experimental results.

6.2 Design of Comparative Experiments

This study implemented a series of comparative tests to assess the performance of real-time IoT data processing and the mechanisms for protection systems. These tests were designed to assess the performance of the proposed Edge Computing solution with respect to the traditional Cloud-based data processing systems. The main indicators considered were processing delays, throughput, efficiency of task scheduling, protection mechanisms, and several other aspects that pertained to the effectiveness of the system. The experiments aimed to model the IoT systems as closely as possible with layers of data collection from a large number of IoT devices, real-time data processing at the Edge, and analysis of activities to be performed in the Cloud. The experiments involved processing data in different ways. One model was using a completely Cloud-based system, which required all data to be sent to the Cloud for processing. The other model was the Intelligent Edge Computing system, where the processing of all hyper-time tasks was implemented at the Edge and only the highly complex, long-term, or highly computational tasks were transferred to the Cloud for processing. The experiments also assessed the performance of the real-time processing system and the mechanisms that were designed to protect data during transmission and processing, including end-to-end encryption, access and intrusion protection systems, and privacy protection.

The results gave an in depth answer to the question of whether the proposed architecture of edge computing could optimize security, integrate real-time processing and preserve data integrity. In general self explanatory, these comparative experiments provide concrete evidence

to substantiate the benefit of edge computing and offer improvements to the processing and security of large scale IoT computing.

6.3 Analysis of Experimental Results

In order to assess the real-time processing ability of intelligent edge computing in IoT systems, this experiment specifically compares edge computing and cloud computing latency and throughput for different types of tasks (see Table 1).

Table 1: Comparison of Real-Time Performance Between Edge Computing and Cloud Computing

Task Type	Data Volume (MB)	Cloud Computing Latency (ms)	Edge Computing Latency (ms)	Cloud Throughput (MB/s)	Edge Throughput (MB/s)
Temperaure Sensor Stream	50	120	45	5.2	12.8
Video surveillance streams	200	450	150	3.1	7.5
Device log stream	100	210	80	4.5	10.2
Mixed data streams	150	320	110	4.0	9.1

As shown in Table 1, edge computing significantly outperforms cloud computing in terms of processing latency. For example, the latency of video surveillance streams was reduced from 450 ms in cloud computing to 150 ms at the edge, a decrease of approximately 66%, while the throughput of edge nodes increased by a factor of 2.4. This indicates that the edge computing architecture can significantly enhance real-time task processing capabilities, with particularly notable low-latency processing effects for high-frequency data streams.

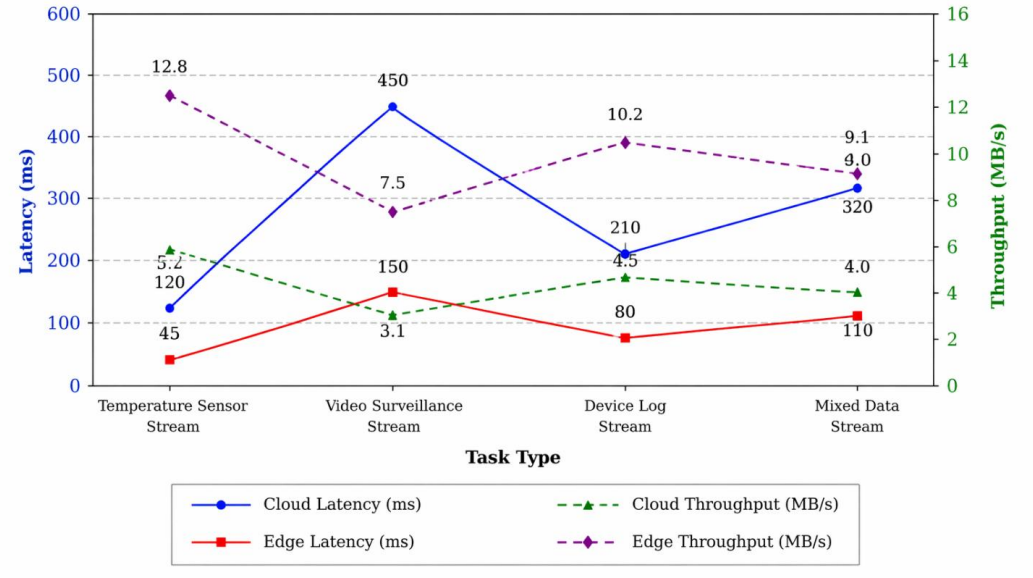

Figure 5: "Real-time Performance Comparison of Edge Computing and Cloud Computing"

Tofurther validate the effectiveness of security protection, this experiment conducted a comparative analysis of intrusion detection, anomaly detection, and data privacy protection (see Table 2).

Table 2: Security Protection and Anomaly Detection Performance

Security Metrics	Cloud Computing (%)	Edge Computing (%)
Intrusion Detection Success Rate	88	94
Anomaly Detection Rate	85	92
Data Encryption Transmission Success Rate	99	99
User privacy leakage rate	4	1

As shown in Table 2, edge computing outperforms cloud computing in intrusion detection and anomaly detection, with success rates increased by 6% and 7%, respectively. At the same time, local data processing reduces the user privacy leakage rate from 4% to 1%. This clearly demonstrates that edge computing has significant advantages in ensuring data security and privacy protection, providing an efficient and secure operating environment for IoT systems.

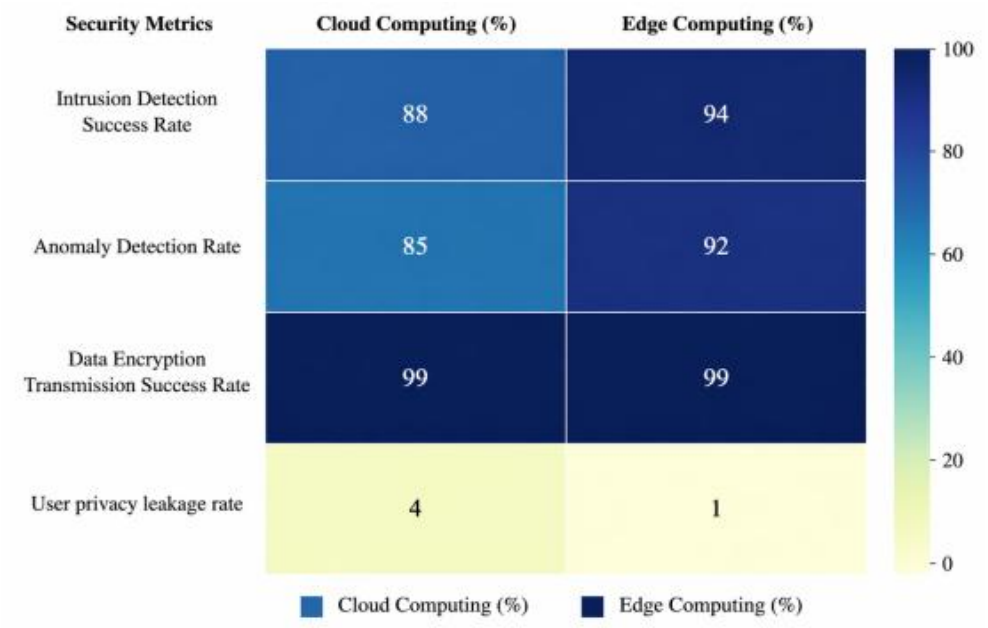

Figure 6: Security Protection and Anomaly Detection Performance on White Background

6.4 Model Performance Evaluation

To obtain primary metrics for the research question of processing speed, cloud resource usage, and processing ability of the intelligent edge computing model for different nodes, the latency, CPU and memory usage, and data throughput of edge nodes were measured (Table 1.) in this experiment.

Table 3: Evaluation of Model Real-Time Processing and Resource Utilization

Node Type	Number of Tasks	Average Processing Latency (ms)	CPU Utilization (%)	Memory Utilization (%)	Data Throughput (MB/s)
Edge Node A	50	42	68	55	11.5
Edge Node B	80	55	74	62	9.8
Edge Node C	60	48	70	58	10.3
Edge Node D	100	65	80	67	8.9

Table 3 illustrates low latency across various nodes and task loads. Processing 50 tasks indicates Edge Node A has a latency of only 42 ms with CPU and memory utilization of 68% and 55%, respectively, showing good resource management. The latency only increases to 65 ms with 100 tasks. With this, the model demonstrates strong real-time processing and efficient resource management to high-load IoT systems.

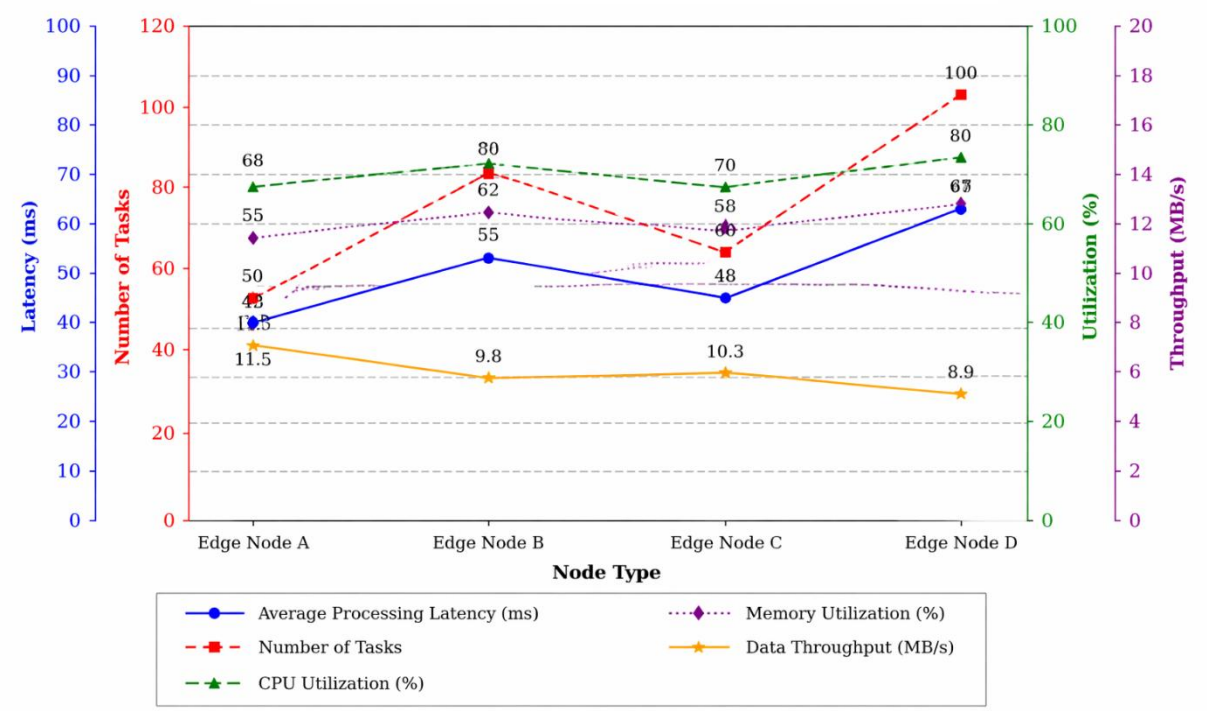

Figure 7: Evaluation of Model Real-Time Processing and Resource Utilization

To test the model's specific performance, this experiment assessed different real time protection, such as intrusion detection and anomaly identification, protection against data breaches, data encryption, and privacy protection (see Table 2).

Table 4: Evaluation of Model Security Performance

Security Metrics	Edge Node (%)	Cloud (%)
Intrusion Detection Accuracy	95	90
Anomaly Detection Rate	92	87
Data Encryption Transmission Success Rate	99	99
User privacy leakage rate	1	3
Malicious Access Blocking Rate	96	91

Table 4 indicates that edge nodes achieve a 5% increase in accuracy for intrusion and anomaly detection and malicious access blocking compared to cloud-based systems. User privacy leakage rates have also improved to 1%. These results validate that edge nodes are effective in security systems for IoT networks.

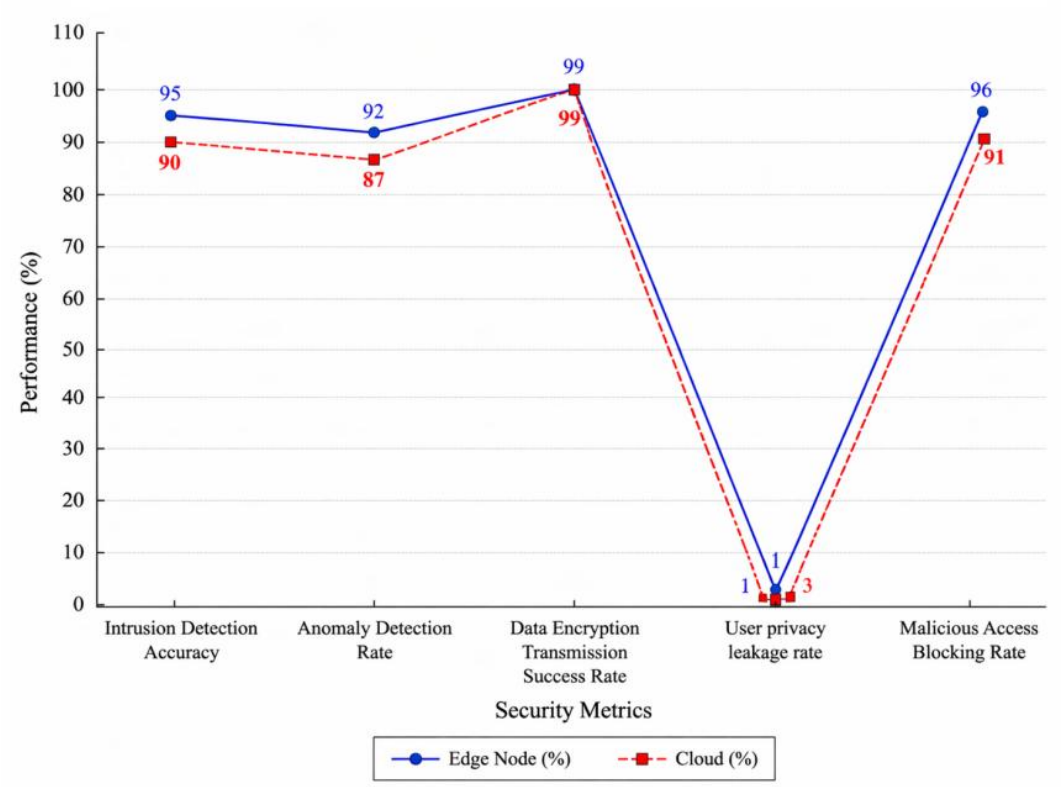

Figure 8: Evaluation of Model Security Performance

7 Conclusion

Real-time data processing and multi-layered security in IoT systems can be achieved through intelligent edge computing. The design makes use of layered edge computing architecture and makes collaboration and processing efficient between edge computing and clouds making an application of task scheduling and load balancing intelligent edge computing. The design offers privacy and security through altered privacy and detection of intrusion and encryption technology. Security and privacy of the IoT systems are improved. The architecture is stable, and the design offers high reliability and processing high-load tasks amid numerous complex security conditions. The intelligent edge provides cross-domain collaboration. The design makes use of an adaptive and dynamic security strategy in large-scale IoT systems to meet real-world challenges and complex security conditions.

References

[1] Bablu T A, Rashid M T. Edge computing and its impact on real-time data processing for IoT-driven applications[J]. Journal of Advanced Computing Systems, 2025, 5(1): 26-43.

[2] Ishrat M, Khan W, Shaikh A A, et al. Intelligent wireless networks: Edge computing, sensors, real-time computing, security, and emerging applications[J]. Deep Learning

Approaches in Intelligent Wireless Networking, 2026: 199-223.

[3] Quy N M, Ngoc L A, Ban N T, et al. Edge Computing for Real-Time Internet of Things Applications: The Future Internet Revolution[J]. Wireless Personal Communications, 2023, 132(2): 1423-1452.

[4] Bargavi M, Muhammed H, Harish P S, et al. Edge Computing and AI for Real-time Analytics in Smart Devices[J]. Asian Journal of Basic Science & Research, 2025, 7(2): 01-09.

[5] Zhukabayeva T, Zholshiyeva L, Karabayev N, et al. Cybersecurity solutions for industrial Internet of Things–edge computing integration: Challenges, threats, and future directions[J]. Sensors, 2025, 25(1): 213.

[6] Rupanetti D, Kaabouch N. Combining edge computing-assisted Internet of Things security with artificial intelligence: Applications, challenges, and opportunities[J]. Applied Sciences, 2024, 14(16): 7104.

[7] Fazeldehkordi E, Grønli T M. A survey of security architectures for edge computing-based IoT[J]. IoT, 2022, 3(3): 332-365.

[8] Modupe O T, Otitoola A A, Oladapo O J, et al. Reviewing the transformational impact of edge computing on real-time data processing and analytics[J]. Computer Science & IT Research Journal, 2024, 5(3): 693-702.

[9] Hossain M E, Tarafder M T R, Ahmed N, et al. Integrating AI with edge computing and cloud services for real-time data processing and decision making[J]. International Journal of Multidisciplinary Sciences and Arts, 2023, 2(4): 252-261.

[10] Ficili I, Giacobbe M, Tricomi G, et al. From sensors to data intelligence: Leveraging IoT, cloud, and edge computing with AI[J]. Sensors, 2025, 25(6): 1763.

[11] Islam U, Alatawi M N, Alqazzaz A, et al. A hybrid fog-edge computing architecture for real-time health monitoring in IoMT systems with optimized latency and threat resilience[J]. Scientific Reports, 2025, 15(1): 25655.

[12] Burhan M, Alam H, Arsalan A, et al. A comprehensive survey on the cooperation of fog computing paradigm-based IoT applications: layered architecture, real-time security issues, and solutions[J]. IEEE Access, 2023, 11: 73303-73329.

[13] Choudhary S, Vijitha S, Bhavani D D, et al. Edge AI: Deploying Artificial Intelligence Models on Edge Devices for Real-Time Analytics [C]//ITM Web of Conferences. EDP Sciences, 2025, 76: 01009.

[14] Deng X, Chen B, Chen X, et al. A trusted edge computing system based on intelligent risk detection for smart IoT[J]. IEEE Transactions on Industrial Informatics, 2023, 20(2): 1445-1454.

[15] Li W, Chen H, Qi Y. Real-time data processing optimization for industrial IoT enabled by edge computing[J]. International Journal of Computer Information Systems and Industrial Management Applications, 2025, 17: 11-11.

[16] Irshad R R, Hussain S, Hussain I, et al. An intelligent Buffalo-based secure edge-enabled computing platform for heterogeneous IoT networks in smart cities[J]. IEEE Access, 2023, 11: 69282-69294.

[17] Hartmann M, Hashmi U S, Imran A. Edge computing in smart healthcare systems: Review, challenges, and research directions[J]. Transactions on Emerging Telecommunications Technologies, 2022, 33(3): e3710.

[18] Shahzad A. EDGE COMPUTING: ENHANCING REAL-TIME DATA PROCESSING[J]. Electronic Research Journal of Engineering, Computer and Applied Sciences, 2025, 7(1): 75-84.

[19] Lv Z, Qiao L, Verma S, et al. AI-enabled IoT-edge data analytics for connected living[J]. ACM Transactions on Internet Technology, 2021, 21(4): 1-20.

[20] Zhang X, Cao Z, Dong W. Overview of edge computing in the agricultural Internet of Things: Key technologies, applications, challenges[J]. IEEE Access, 2020, 8: 141748-141761.