International Journal for Housing Science and Its Applications
Publish August 10, 2025. Volume 46, Issue 4 Pages 2997-3010

https://doi.org/10.70517/ijhsa464249

# Research on Intelligent Manufacturing Data Collection and Transmission Optimization Methods Based on Cloud-Edge Collaboration

**Haijun Zhou**[1,*]
[1] BEIJING POLYTECHNIC UNIVERSITY, Beijing, 100176, China
Corresponding authors: (e-mail: 17701326530@163.com).

**Abstract** This paper proposes an optimised data acquisition and transmission method based on cloud-edge collaboration. By constructing a real-time data processing framework that integrates CNN algorithms, optimising the protocol conversion mechanism for heterogeneous networks, and designing a data interaction control system with business prioritisation and dynamic bandwidth allocation capabilities, the processing efficiency and transmission reliability of manufacturing data are significantly improved. In metal coating thickness detection, the CNN-based cloud-edge collaborative fusion algorithm achieved a fusion result of 61.1146 μm (reference value: 61.11 μm), with a relative error of only 0.0088%, outperforming the arithmetic mean method (0.1353%) and the evidence theory method (0.0362%). The fusion process took 0.12 ms, representing an over 80% speedup compared to traditional methods. In the 10Mb candy packaging recognition task, the cloud-edge collaborative model demonstrated comprehensive performance leadership, with a latency of only 4.36 seconds, which is 51.6% of the fog computing FC's 8.45 seconds and 26.3% of the local computing LC's 16.57 seconds. The energy consumption of the cloud computing CC algorithm is 329.41 J, which is 49.1% more energy-efficient than FC's 646.61 J and 78.6% lower than LC's 1540.72 J. The reliability task success rate is 95.11%, significantly higher than FC's 83.23% and LC's 65.26%. This study validated the significant advantages of the cloud-edge collaboration architecture in terms of data real-time performance, energy efficiency, and reliability, providing an effective solution for optimising intelligent manufacturing systems.

**Index Terms** cloud-edge collaboration, intelligent manufacturing, data acquisition and transmission, CNN data fusion, heterogeneous network optimisation

## I. Introduction

Smart manufacturing is an emerging manufacturing method that integrates traditional manufacturing technology with intelligent technology to serve the entire product life cycle. Relying on technologies such as sensing, testing, numerical control, robotics, computers, networks, and artificial intelligence, smart manufacturing can perceive, analyse, reason, make decisions, and control during the manufacturing process, thereby achieving dynamic response to product demand [1], [2]. Currently, smart manufacturing technology has become a global trend in the development of the manufacturing industry. In China, smart manufacturing has been elevated to a national strategic level, serving as the primary focus of the 'Made in China 2025' initiative and one of the emerging engineering disciplines prioritised for development in China's higher education institutions [3], [4].

The core architecture of smart manufacturing generally encompasses the equipment layer, perception layer, execution layer, and decision-making layer. The perception layer typically includes various sensors, PLCs (programmable logic controllers), industrial control computers, etc. Its core function is data collection, which involves real-time collection and necessary pre-processing of status data from relevant equipment in the smart workshop (including warehousing, logistics, personnel, products, and environmental factors), and then transmitting the data to a server (or cloud) database via wired networks, Wi-Fi, 4G/5G, etc., for subsequent analysis, inference, and decision-making [5]-[8].

Data collection and transmission are critical components of smart manufacturing. By ensuring efficient and high-quality production lines, they enable rapid, precise, and reliable data collection and processing, providing real-time, accurate data support for corporate management decisions [9], [10]. The real-time data collection functionality in intelligent manufacturing systems has transformed the traditional distributed numerical control (DNC) system's approach to data transmission and management within equipment. By integrating information as nodes into enterprise information management systems, data is accurately and promptly transmitted, analysed, and stored,

facilitating information exchange and collaboration between management and operational levels [11]. The Manufacturing Data Collection and Status Management (MDC) system is an extension of the traditional DNC system, based on lean manufacturing principles, to manage real-time data [12]. By adopting a real-time data acquisition system, the smart manufacturing system achieves networking of production equipment, establishing comprehensive data exchange at the workshop production site. This enables the collection, transmission, and analysis of equipment status, workshop conditions, and production data, maximising the fulfilment of production management needs in smart factories, achieving big data storage for production management, and enabling cloud computing functionality [13]-[15]. This provides technical support for smart manufacturing workflows and serves as the foundation for smart manufacturing.

Data collection provides the information foundation for smart manufacturing and serves as a critical core module of smart manufacturing systems. Given the diversity of production equipment and corresponding data types in smart workshops, the primary challenges in implementing data collection and transmission lie in developing a unified data collection solution and data transmission protocol applicable to all devices, as well as the reduced efficiency of traditional centralized cloud computing architectures [16]. Guo et al. [17] addressed the data collection and storage issues of CNC machine tools in enterprise manufacturing workshops by utilising the MTConnect protocol and the FOCAS protocol-based device information model, and constructed a prototype system for CNC machine tool data collection and monitoring. Currently, research on optimising data collection and transmission for smart manufacturing is in its early stages. In addition to addressing protocol requirements, improving the computational capabilities of centralised cloud computing is equally important.

Cloud-edge collaboration is an architectural model that combines cloud computing with edge computing. Cloud computing centres are responsible for processing large-scale data and complex tasks, while edge computing nodes are located close to data sources or user terminals and can respond quickly to local tasks [18], [19]. This makes data processing more flexible and efficient, optimises resource allocation, improves overall system performance, and provides better services for various application scenarios [20], [21]. Additionally, cloud-edge collaboration enhances network stability and reliability, reduces data transmission latency, and helps improve data security and privacy protection [22], [23]. In the era of IoT artificial intelligence, Lu et al. [24] designed a cloud-edge-terminal collaborative data collection scheme based on swarm intelligence, which uses a particle swarm optimisation algorithm to match workers with sensing tasks. Shi et al. [25] applied cloud-edge collaboration in industrial IoT to jointly optimise sensor data sampling rates and edge server pre-processing modes. The data sampling rate determines data accuracy, and improving the sampling rate helps to enhance data transmission efficiency. Wu et al. [26] proposed an efficient industrial data transmission scheme combining deep learning and cloud-edge collaboration, which saves communication bandwidth while ensuring data transmission accuracy. Cloud-edge collaboration organically integrates the powerful computing power of the cloud with the real-time response capabilities of the edge. Hu et al. [27] pointed out that under cloud-edge collaboration, combining edge computing and cloud computing can meet real-time intensive data computing demands, thereby optimising data quality control. Yang et al. [28] aimed to optimise data transmission, storage, and analysis for condition monitoring in industrial IoT. They employed cloud-edge collaboration to create a cloud-edge collaborative condition monitoring platform for intelligent manufacturing systems supported by microservices. This platform effectively optimises the computational and real-time diagnostic capabilities of the cloud layer.

The article first explores the core methods of data fusion using deep learning technology in a cloud-edge collaboration environment. Specifically, it investigates data fusion methods for intelligent manufacturing systems based on CNN algorithms. This method leverages the powerful feature extraction and non-linear modelling capabilities of CNNs, deployed at the edge or in the cloud, to perform feature-level or decision-level fusion of heterogeneous raw data from different devices and sensors. The aim is to reduce redundancy, extract key information, improve data quality, provide more precise input for upper-layer intelligent applications, and simultaneously reduce the amount of data uploaded to the cloud. Additionally, the paper delves into the foundational technologies and optimisations for data collection and transmission in intelligent manufacturing scenarios. In terms of production data collection methods, the paper analyses collection strategies for different devices and protocols. The focus is on the analysis and implementation of OPC UA client functionality. Addressing the common issue of heterogeneous device protocols in industrial environments, the paper provides a detailed explanation of a client implementation scheme based on the OPC UA unified architecture. This scheme aims to establish standardised data access interfaces, abstract underlying device differences, and achieve efficient, secure, and interoperable data collection. Finally, in response to the complexity of industrial field network environments, the study focused on optimising data collection and transmission in heterogeneous networks. At the heterogeneous network protocol conversion level, it explored how to achieve efficient and reliable conversion between different industrial network protocols and upper-layer communication protocols, addressing the ‘information silo’ issue and ensuring smooth

data flow between the cloud and edge. In terms of data transmission network routing optimisation, the research examined how to dynamically select the optimal transmission path or strategy based on real-time network status, data priority, and other factors in complex cloud-edge network topologies. This aims to minimise transmission latency, improve bandwidth utilisation, and ensure the reliability and real-time nature of critical data, thereby optimising the data transmission performance of the entire cloud-edge collaborative system.

## II. Intelligent manufacturing data collection, transmission, and integration optimisation under a cloud-edge collaboration architecture

### *II. A. Data fusion in intelligent manufacturing systems based on CNN algorithms under cloud-edge collaboration*

The intelligent manufacturing system cloud-edge collaborative scheduling solution is an advanced production management model that aims to achieve the coordinated development of business models and manufacturing models, thereby improving production efficiency and quality control levels. The overall process of the intelligent manufacturing system cloud-edge collaborative scheduling solution is shown in Figure 1. Through the collaborative work of three key platforms—the artificial intelligence platform, edge computing platform, and intelligent factory platform—this solution achieves real-time collaboration between the cloud platform and industrial field equipment.

The artificial intelligence platform is the core of the entire solution, comprising a cloud platform scheduling algorithm library, distributed devices, a fault diagnosis system, and data analysis and processing modules. The cloud platform scheduling algorithm library generates corresponding scheduling algorithm tables based on market demand and production plans, guiding distributed devices in production. The fault diagnosis system monitors device operating status in real time and promptly addresses any issues detected. The data analysis and processing module analyses and models data from the production process to support the optimisation of production workflows. The edge computing platform serves as a bridge between the cloud platform and the smart factory platform, comprising edge computing solutions, edge computing systems, and edge computing software. These solutions and systems can convert device protocols, collect, store, and analyse data, and perform online simulation and real-time control operations. Through the edge computing platform, the smart factory platform can better collaborate with the cloud platform to achieve real-time data transmission and control.

The smart factory platform is the foundation of the entire solution, comprising smart factory solutions, smart factory systems, and smart factory software. These solutions and systems enable comprehensive digital management of the production process, including design, simulation, production, and testing. At the same time, the smart factory platform can connect to distributed devices via IoT technology to monitor and control them in real time.

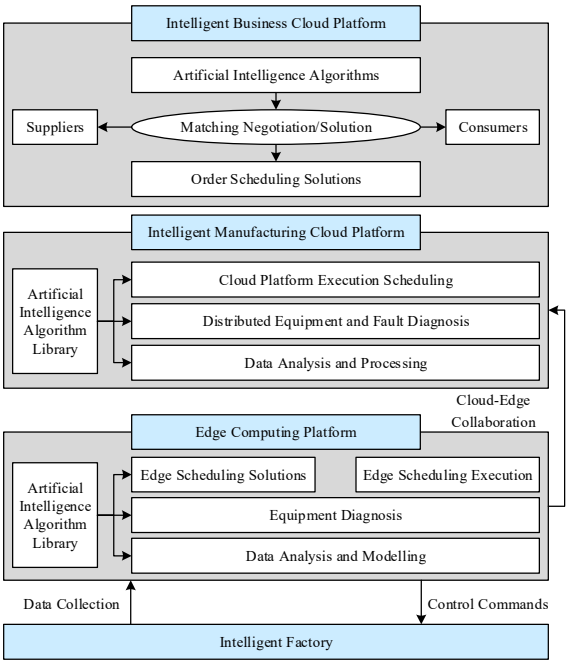


Figure 1: Cloud-edge collaborative scheduling of intelligent manufacturing systems

### *II. B.Data Collection and Transmission in Smart Manufacturing*

#### II. B. 1) Production Data Collection Methods

The data collection requirements under the aforementioned scheme demand high frequency and high accuracy, thereby placing stringent demands on real-time performance during the data collection process. OPC UA is a vendor-neutral, platform-independent architecture that provides extensibility features for relevant platforms, enabling communication between industrial equipment PLCs and upper-level computers in accordance with OPC standards. Therefore, based on the OPC UA Server, a data transmission module is developed, and device-to-device data communication is achieved using the TwinCAT software provided by the production line equipment manufacturer. During the data collection process for the production line, the data to be collected is categorised into the following types:

(1) For data acquisition of PLC-controlled equipment used in assembly lines, data acquisition interfaces are generally connected to BECKHOFF PLCs, and data in the PLCs is acquired and processed using TwinCAT.

(2) For data collection from non-PLC-controlled equipment on the production line, if the equipment supplier can provide a data interface that is compatible with the PLC, the PLC can also be used to transmit the collected data. If the equipment cannot provide the relevant data interface, sensors can be installed on the equipment to measure the relevant status data and complete data collection.

(3) Small peripheral equipment on the production line is outside the control range of the PLC and only provides equipment operating status or fault information. Therefore, it needs to enter the PLC control system through the IO interface, and then be uniformly provided and opened by the PLC to Twin CAT.

(4) For data from other equipment on the production line, redundant IO points for PLCs provided by equipment suppliers can be uniformly collected through PLCs and opened to Twin CAT.

#### II. B. 2) OPC UA Client Function Analysis and Setup

The primary function of the OPC UA client is to establish a connection with the OPC UA server on the PLC and read relevant data, making it a critical component in the instance verification process. Since the OPC UA specification provides a unified address space, service model, and security model, it enables all data, alarms, events, and historical information to be unified within the address space of the OPC UA server, thereby providing a unified interface to the outside world. The functional architecture of the OPC UA server is shown in Figure 2.

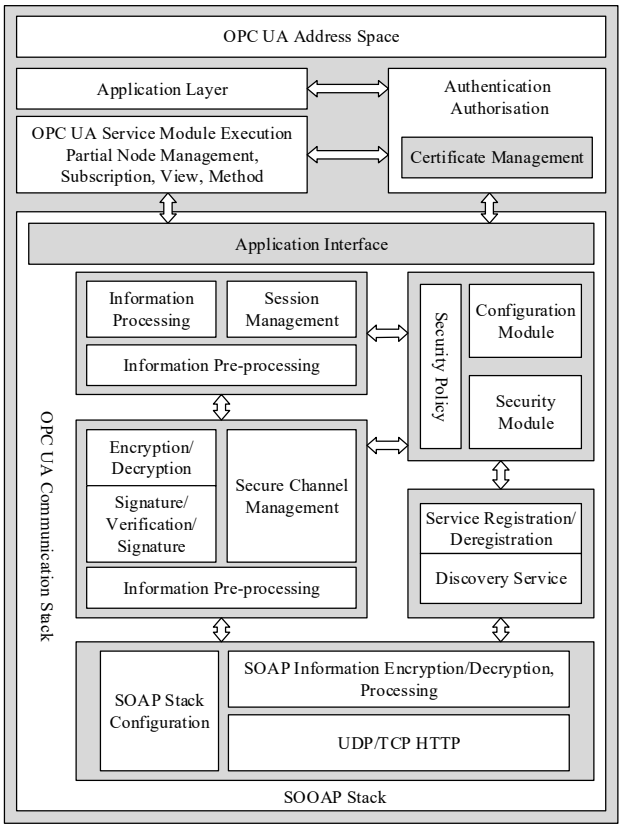


Figure 2: The functional architecture of the OPC UA server

### *II. C.Heterogeneous Network Data Collection and Transmission Optimisation*

#### II. C. 1) Heterogeneous Network Protocol Conversion

To optimise the data collection process at the manufacturing resource layer, we propose a heterogeneous network protocol conversion strategy that deploys edge intelligent gateways at the edge layer to collect and convert underlying data. The specific execution process is shown in Figure 3. When the underlying device begins data collection, the data node sends a communication request to the intelligent manufacturing gateway. After authentication, if the node has not been registered, the gateway performs registration and records the node's connection attributes, IP address, and port number. If the node is already registered, the gateway directly reads the relevant information and establishes the connection. When processing business flow inputs, the gateway prioritises tasks based on the current tasks being executed by the edge intelligent gateway. Considering the characteristics of manufacturing environment information, this paper categorises data transmission based on latency requirements into three priority levels: real-time transmission priority, high priority, and general priority. After priority determination, pending data enters the corresponding priority queue. If the queue is full, it is transferred to the next priority queue. When all general priority queues are full, data processing operations are transferred to the second gateway.

When data enters the processing area from the priority queue, the edge intelligent gateway first performs local protocol conversion and data packet search based on the data format. If the cache area contains the corresponding protocol conversion data processing method, it will be directly adopted; if not, network configuration will be performed to obtain the required protocol conversion scheme from the cloud database. Since OpenFlow flow table matching during data transmission can only recognise Ethernet protocol formats, unified conversion is performed when handling heterogeneous network protocols. This involves converting the original multi-protocol data (Modbus, Ethernet, TCP/IP, etc.) into Ethernet protocol format data. This can be done in three steps: packet disassembly and parsing, data alignment, and encapsulation and packaging. When de-packing and parsing data, header information must be extracted according to the data composition rules of different protocols. Since the data composition of different protocols varies, data alignment is required for identical components across different protocols after parsing, following a unified set of rules. Finally, the processed data is encapsulated and packaged, with Ethernet addresses, IP addresses, port numbers, and other information required for OpenFlow flow table matching encapsulated into the header, and the data is uploaded.

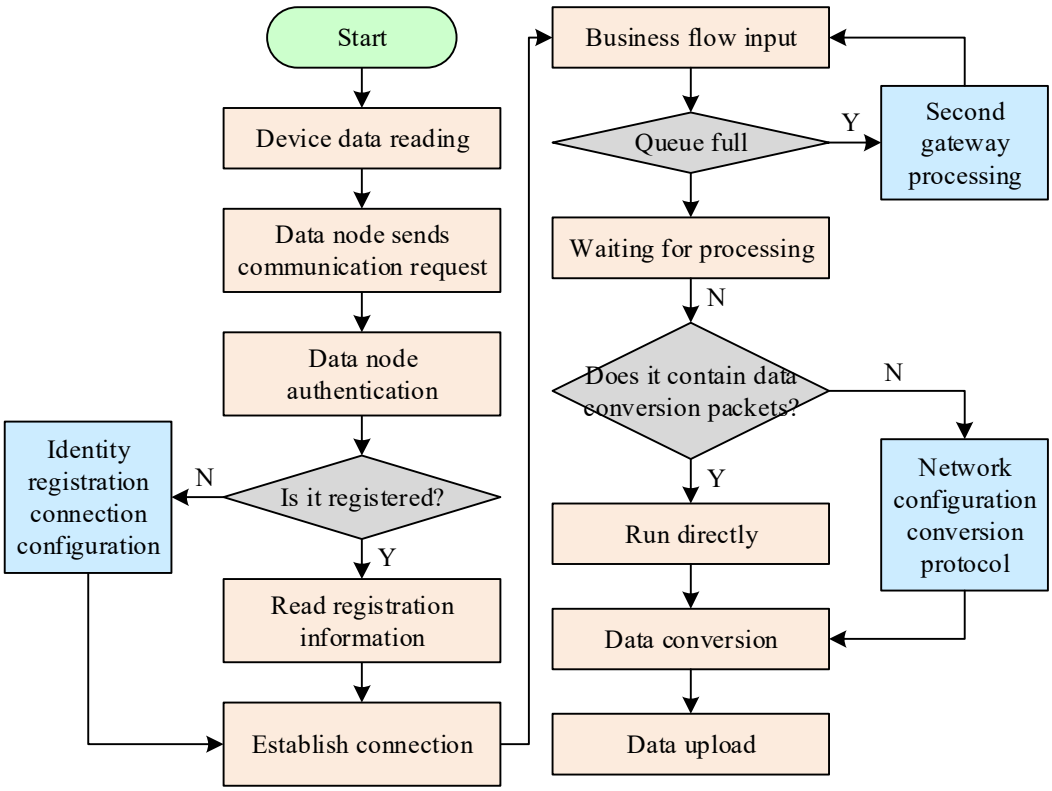


Figure 3: The process of heterogeneous network protocol conversion

**II. C. 2) Data Transmission Network Routing Optimisation**

Figure 4 shows the SDN architecture used in this paper. The data transmission network topology of this architecture is complex, which results in more diverse data transmission routes. For example, when manufacturing resource R1 serves as the source device and R8 as the destination device for data transmission, the data transmission path between edge intelligent gateways E1 and E2 can be selected as E1-S1-S4-S3-E2, E1-S1-S2-S3-E2, or E1-S1-S5-S4-S3-E2; The selection of transmission paths has a critical impact on data transmission efficiency. If the widely used Dijkstra algorithm is employed to select paths based on the shortest path, although the path selection process is simple and computationally fast, it suffers from drawbacks such as low network utilisation and excessive local network load. To optimise transmission routing, this paper adopts a path selection scheme based on genetic algorithms, combined with dynamic updates to the switch transmission flow table based on the current state of the transmission network.

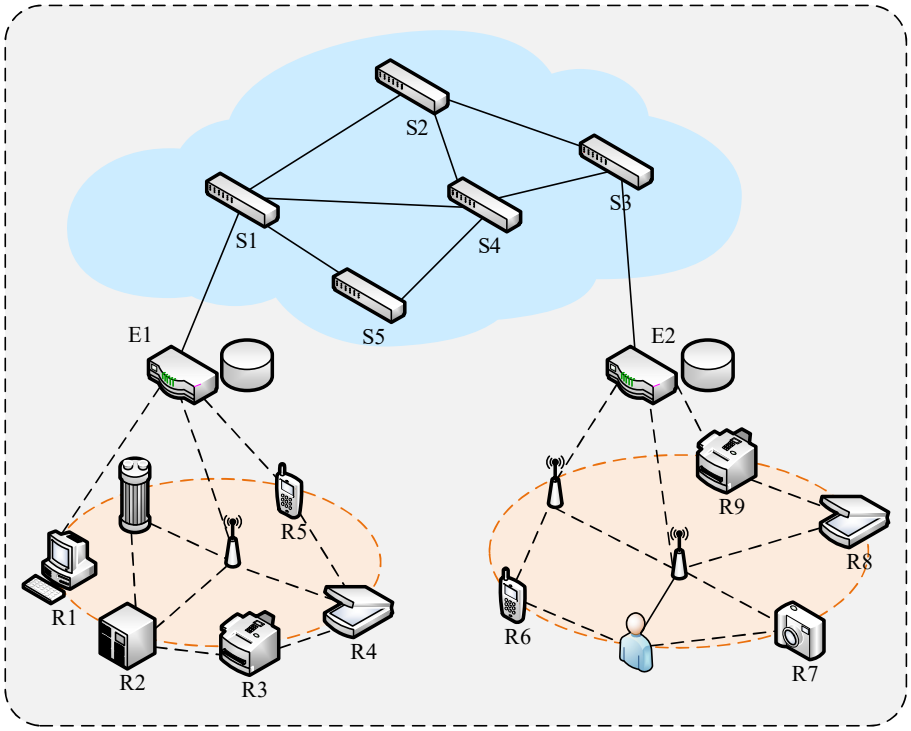


Figure 4: Schematic diagram of data transmission network topology

Based on the network topology structure shown in Figure 4, the set of switches is represented as $S_A$, a single switch device is represented as $S_i$, and the set of $n$ reachable links between edge intelligent gateways $E_i$ and $E_j$ is represented as $L_{ij} = \{l_1, l_2, \cdots, l_n\}$. Based on the real-time status of the transmission network, we use the four factors of available bandwidth $(W_{ij})$, bandwidth utilisation rate $(R_{ij})$, latency $(T_{ij})$, and packet loss rate $(D_{ij})$ between switches $S_i$ and $S_j$ as individual selection factors in the subsequent genetic algorithm. Among these, the bandwidth utilisation rate:

$$R_{ij} = \frac{\max W_{ij} - W_{ij}}{\max W_{ij}} \tag{1}$$

The total delay of link $l_{ij}$ can be expressed by summing the delays of each part:

$$T_{ij} = T_{i0} + T_{op} + \cdots + T_{qj} \tag{2}$$

The total packet loss rate of the link is defined as:

$$D_{ij} = 1 - (1 - D_{i0}) * (1 - D_{op}) * \cdots * (1 - D_{qj}) \tag{3}$$

Combining the above routing selection reference factors, the fitness function of link $l_k$ can be defined as:

$$f_k = \alpha W_k - \beta R_k - \gamma T_k - \delta D_k \tag{4}$$

Among these, $W_k$ represents the available bandwidth constraint in link $l_k$, $W_k = min\{W_{io}, W_{op}, \cdots, W_{qj}\}$ denotes the segment with the minimum available bandwidth in the link between switches $S_i$ and $S_j$, $R_k$ indicates the bandwidth utilisation rate of that segment, $T_k$ signifies the total delay of link $l_k$, and $D_k$ represents the total packet loss rate of the link. $(\alpha, \beta, \gamma, \delta)$ represents the corresponding impact factor coefficients. The types of information transmitted in a manufacturing environment vary, and services can be categorised into three major types based on QoS (Quality of Service): control class $(S_{con})$, monitoring class $(S_{mon})$, and management decision-making class $(S_{man})$. The impact factor coefficients for different types of data are divided into intervals according to QoS requirements, making path selection more suitable for the specific application scenario.

After determining the optimisation indicators, a genetic algorithm is used to solve the routing optimisation scheme. Combined with the topological structure, the process of solving the optimised path between edge intelligent gateways E1 and E2 is mainly as follows:

(1) Select the initial population. The number of initial populations can be determined based on the complexity of the specific topological structure. Since the topological structure in this section is relatively simple, the number of initial populations can be set to 3. The initial population is represented as $\{l_1, l_2, l_3\}$. Among them, $l_1 = (S1, S2, S3)$, $l_2 = (S1, S2, S4, S3)$, and $l_3 = (S1, S5, S4, S3)$.

(2) Sample coding: Path binary coding can be achieved by numbering the switches. In this section's example, there are relatively few switch samples, so we can use four-digit binary coding for path coding. For paths with fewer hops, we can fill in the gaps using the final switch coding to facilitate subsequent operations.

$$\begin{aligned} &0001 \quad 0010 \quad 0011 \quad 0011 \rightarrow l_1 = (S1, S2, S3, S3) \\ &0001 \quad 0010 \quad 0100 \quad 0011 \rightarrow l_2 = (S1, S2, S4, S3) \\ &0001 \quad 0101 \quad 0100 \quad 0011 \rightarrow l_3 = (S1, S5, S4, S3) \end{aligned} \tag{5}$$

(3) Crossover: By performing single-point crossover on the encodings of paths $l_1$ and $l_3$, new sample individuals $l_4$ and $l_5$ are generated:

$$\begin{aligned} &0001 \quad 0101 \quad 0011 \quad 0011 \rightarrow l_4 = (S1, S5, S3, S3) \\ &\qquad \uparrow\uparrow\uparrow\uparrow \\ &\qquad \downarrow\downarrow\downarrow\downarrow \\ &0001 \quad 0010 \quad 0100 \quad 0011 \rightarrow l_5 = (S1, S2, S4, S3) \end{aligned} \tag{6}$$

(4) Mutation: Randomly mutate the position coding of individuals, changing 1 to 0 and 0 to 1 at that position to generate new individuals. $l_6$:

$$\begin{aligned} &0001 \quad 0010 \quad 0011 \quad 0011 \rightarrow l_1 = (S1, S2, S3, S3) \\ &\qquad \uparrow \quad \uparrow \\ &\qquad \uparrow \quad \uparrow \\ &0001 \quad 0011 \quad 0010 \quad 0011 \rightarrow l_6 = (S1, S3, S2, S3) \end{aligned} \tag{7}$$

(5) Next-generation population selection: After crossover and mutation, the first-generation population $\{l_1, l_2, l_3\}$ produces new individuals $\{l_4, l_5, l_6\}$. The fitness function proposed earlier is used as the criterion for individual selection. The survival rate of individuals is defined as:

$$P_i = f_i \Big/ \sum_i f_i \tag{8}$$

Retain individuals with high survival rates and form new groups.

(6) Repeat the above process of cross-breeding, mutation, and selection on new sample populations until the

convergence condition is met.

Through multiple comparisons and screenings using genetic algorithms, network optimisation routing schemes that meet the conditions can be generated. The SDN controller generates and distributes transmission flow tables based on the existing optimised schemes. After entering the switch, data packets are matched and subsequently transmitted according to the grouping rules stored in the flow tables, thereby achieving transmission path optimisation based on network status.

## III. Multi-dimensional experimental verification of intelligent manufacturing data collection and transmission based on cloud-edge collaboration

In order to comprehensively evaluate the effectiveness and superiority of the aforementioned theoretical models, algorithms, and system designs, this chapter will conduct systematic experimental verification and analysis. First, we will perform instance verification and comparative analysis of the core data fusion algorithms. Second, we will conduct simulation tests on the designed cloud-edge collaborative data interaction control system. Finally, in typical smart manufacturing scenarios, we will compare and analyse the comprehensive performance of the cloud-edge collaborative model with local computing and fog computing models.

### *III. A. Real-time data fusion*

#### III. A. 1) Analysis of the results of sample data fusion from random sampling of instances

To validate the effectiveness of the proposed real-time fusion method based on the CNN algorithm under cloud-edge collaboration, experiments were conducted using metalised coating thickness data from a randomly selected sample. The reference thickness for this sample is 61.11 μm. The measured thickness data for the eight metallised coatings are 61.68, 61.61, 61.29, 60.91, 60.82, 61.42, 61.08, and 61.53, respectively.

First, check the consistency of the data. After calculation, the interquartile range Dd is 0.36. After selecting the $\beta$ value of 1.0 for comparison, it can be seen that there are no error values affecting the consistency of the data in the eight data points. Normalise these eight measured values. Next, the normalised measurement values are used as the identification framework in evidence theory. Through calculation, the similarity matrix S between the sensor values and the identification target can be obtained. Let A represent the vector composed of ai. Through calculation, a set of mutual support coefficients wq based on the first value is obtained. Further calculations yield the probability distribution values for each sensor measurement, the generated evidence ei, and the final combined evidence E, as shown in Table 1. Finally, the final result of this data fusion is calculated as $Dm = 61.1146\mu m$ .

Table 1: Probability distribution values and combined evidence E

| E | Probability distribution value | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | $m(D_1)$ | $m(D_2)$ | $m(D_3)$ | $m(D_4)$ | $m(D_5)$ | $m(D_6)$ | $m(D_7)$ | $m(D_8)$ |
| $e_1$ | 0.1299 | 0.1223 | 0.2202 | 0.2174 | 0.0867 | 0.1833 | 0.1116 | 0.1181 |
| $e_2$ | 0.0640 | 0.1713 | 0.1007 | 0.0959 | 0.1061 | 0.1836 | 0.1144 | 0.0782 |
| $e_3$ | 0.0802 | 0.1777 | 0.2018 | 0.2376 | 0.2182 | 0.1093 | 0.1826 | 0.1424 |
| $e_4$ | 0.1617 | 0.2228 | 0.1391 | 0.2026 | 0.1620 | 0.1207 | 0.1935 | 0.0824 |
| $e_5$ | 0.1788 | 0.1729 | 0.1857 | 0.1291 | 0.2122 | 0.0768 | 0.1906 | 0.1380 |
| $e_6$ | 0.1547 | 0.2173 | 0.2043 | 0.1550 | 0.1726 | 0.2201 | 0.1791 | 0.1016 |
| $e_7$ | 0.1048 | 0.2306 | 0.1844 | 0.1409 | 0.1833 | 0.0657 | 0.0644 | 0.0938 |
| $e_8$ | 0.1380 | 0.1617 | 0.1098 | 0.0999 | 0.2016 | 0.1196 | 0.1346 | 0.0724 |
| E | 0.1265 | 0.1846 | 0.1683 | 0.1598 | 0.1678 | 0.1349 | 0.1464 | 0.1078 |

#### III. A. 2) Comparative analysis of different fusion algorithms

To validate the superiority of the CNN data fusion algorithm under cloud-edge collaboration, a comparative analysis was conducted with arithmetic mean and evidence-based fusion algorithms. The results of the various algorithms are compared in Table 2 and Figure 5.

Table 2: The comparison results of various integration methods

| | CNN | Arithmetic average | Evidence theory |
|---|---|---|---|
| Final result | 61.1146 | 61.1928 | 61.1321 |
| Relative result | 0.0054 | 0.0828 | 0.0221 |

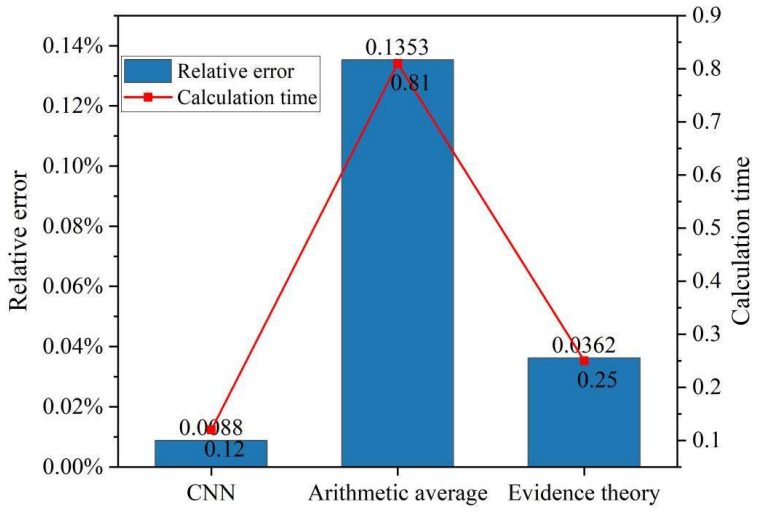


Figure 5: The comparison results of various integration methods

The fusion result of the CNN algorithm is 61.1146 μm, which is closest to the reference value of 61.11 μm, with a relative error of only 0.0088%, significantly better than the arithmetic mean method's 0.1353% and the evidence theory method's 0.0362%. In terms of computational efficiency, the CNN algorithm takes only 0.12 ms, far below the 0.81 ms of the arithmetic mean method and the 0.25 ms of the evidence theory method. This indicates that the CNN algorithm has significant advantages in both accuracy and real-time performance, making it particularly suitable for high-real-time requirements in cloud-edge collaboration scenarios.

### *III. B. Simulation and verification of data interaction control systems based on cloud-edge collaboration*

The effective transmission and control of data is another key aspect of implementing a cloud-edge collaboration architecture. Therefore, the rest of this chapter will focus on verifying the performance of the designed cloud-edge collaboration data interaction control system in terms of ensuring data transmission reliability and service quality. Simulation tests and analyses were conducted on the designed service differentiation module and queue management module.

#### III. B. 1) Test Environment Setup

The configuration of the computer used in this experiment is as follows: Intel Core i7-10875H @2.3GHz processor, 16GB memory, Win10 operating system. We deployed an Ubuntu 16.04 virtual machine in VMware Workstation and built a simulation environment based on Mininet and Floodlight controllers in the virtual machine.

The experiment uses Mininet to establish a custom topology and connect to the Floodlight controller to simulate the network system architecture based on CNN.

#### III. B. 2) Testing of Distinct Service Modules

When testing the service modules in different zones, QoS policies were configured on the switch for flows 1, 2, and 3, and three 30-second data streams were generated using the Iperf network performance testing tool. When the host sent different business flows to another host, the iperf command was used to measure the bandwidth of each link simultaneously. The results are shown in Figure 6.

Business classification can be divided into three types: EF: alarm signals, feedback control; AF: interface data, cloud-based policy distribution, state estimation, fault diagnosis; BE: device status monitoring, basic production data. Flow1 belongs to the EF type of business in the model, flow2 belongs to the AF type, and flow3 belongs to the BE type. It can be seen that when the bandwidth is limited to 10M, after the QoS policy is matched with the corresponding flow, different business flows entering different queues are allocated the configured bandwidth. Flow1, which is the EF type business flow, is allocated the most bandwidth, while flow3 is allocated the least bandwidth. This indicates that the service differentiation module has indeed achieved the basic functionality of business classification.

#### III. B. 3) Testing the queue management module

To test the effectiveness of the queue management module, each queue was assigned the same bandwidth during queue configuration, and each queue was given a different priority. The priorities P of the three business flows flow1,

flow2, and flow3 are 1, 2, and 3, respectively.

When generating data streams using Iperf, the flow1, flow2, and flow3 business streams are generated sequentially at 10-second intervals and continue for 30 seconds. When different types of business enter the corresponding priority queues, the experimental results are shown in Figure 7.

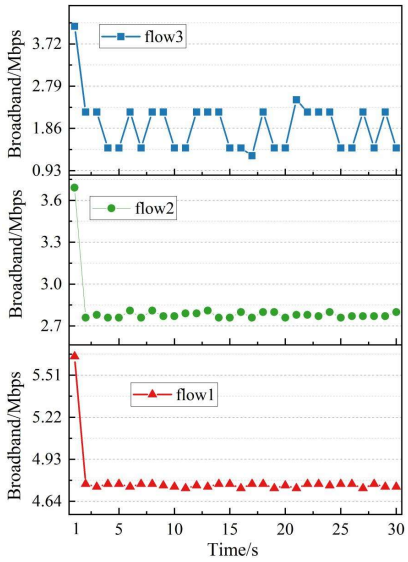


Figure 6: Line chart showing different business bandwidth allocations

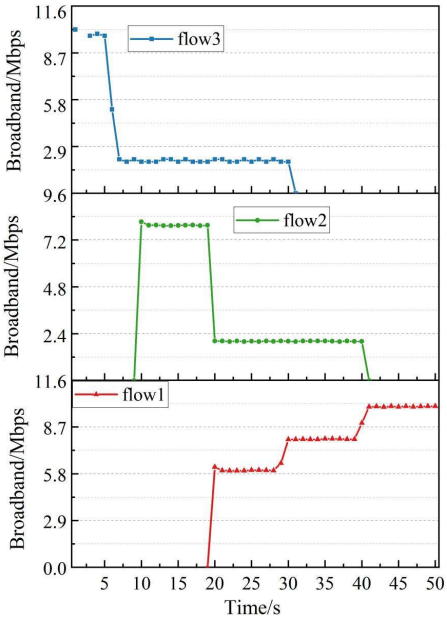


Figure 7: Bar chart showing bandwidth allocation for different priority queues

Between 0 and 10 seconds, only flow3 (BE-class business flow) is active, so it occupies the entire 10M bandwidth; Between 10 and 20 seconds, the flow2 business flow of the AF class appears, and the bandwidth allocated to the BE-type business flow with the lowest priority quickly becomes the minimum guaranteed bandwidth; Between 20 and 30 seconds, all three business flows coexist, with the EF-class business flow allocated the most bandwidth, fluctuating between 5.8 Mbps, while flow2 and flow3 fluctuate between 2.4 and 2.1 Mbps respectively; Between 30 and 40 seconds, flow3 completes its transmission, and the bandwidth previously allocated to it is dynamically reassigned to flow1; in the final 10 seconds, flow2 also completes its transmission, and flow1 occupies all available bandwidth resources.

In summary, the queue management module can implement queue priority functions, allowing the highest priority services to be allocated the most bandwidth to ensure fast transmission, while other priority services are also assigned a minimum guaranteed bandwidth to prevent them from being 'starved.' Simulation tests have fully demonstrated the reliability of data transmission for the cloud-edge collaborative data interaction control system designed in this article.

### *III. C. Different model simulation experiments for intelligent manufacturing*

After verifying the performance of key algorithms and subsystems, it is necessary to further evaluate the comprehensive advantages of the cloud-edge collaboration model in real intelligent manufacturing scenarios from the perspective of overall architectural efficiency. Comparison simulations are conducted with local computing models and fog computing models. This section uses simulation experiments to separately calculate the latency, energy consumption, and reliability performance results of the three computing modes.

The simulation platform is configured with an Intel i5 CPU running at 3.3 GHz and 4 GB of memory. The task processing process is simulated in MATLAB R2014a software under three computing modes. The task latency, energy consumption, and reliability are defined as follows:

Delay: The time interval between the start of task execution $t_{start}$ and the end of task execution $t_{end}$, as shown in Equation (9).

$$TT_{\text{latency}} = t_{end} - t_{start} \tag{9}$$

Energy consumption: The energy consumed by the computing node to execute all tasks, as shown in Equation (10).

$$E = E_{task1} + E_{task2} + \ldots + E_{taskN} \tag{10}$$

Reliability: Task reliability is represented by the task execution success rate, which is the ratio of the number of successfully executed tasks $N_{success}$ to the total number of tasks $N_{total}$ within the maximum tolerance time range, as shown in Equation (11).

$$\gamma = \frac{N_{success}}{N_{total}} \times 100\% \tag{11}$$

Establish a smart production line production scenario, with the production object being personalised customised candy packaging, and the production process being the identification of multiple categories of candy. The cloud server provides cloud computing mode for task requests, and tasks are executed serially on the cloud server. The selected smart terminal device sends one task request at a time, with the number of tasks remaining unchanged and the task data volume gradually increasing from 1 Mb to 10 Mb.

Randomly select 10 smart terminal devices to provide local computing mode for task requests, and randomly select 3 fog nodes to provide fog computing mode for task requests. Tasks are distributed to the selected fog nodes in a round-robin manner, and tasks are executed serially on each fog node. To ensure experimental accuracy, each experiment is conducted 20 times, and the average value is taken to obtain the simulation results for local computing mode (LC), fog computing mode (FC), and cloud computing mode (CC) in terms of latency, energy consumption, and reliability.

#### III. C. 1) Comparison and analysis of delay results

By changing the size of the task data volume, simulation calculations were performed on the latency results of tasks under three computing modes, and the task latency simulation results are shown in Figure 8.

Under the LC algorithm, latency increases sharply as data volume grows, rising from 0.22 seconds at 1 Mb to 16.57 seconds at 10 Mb, indicating its inability to handle large-scale data. This is due to the limited computational resources of smart terminal devices in the local computing mode, which cannot quickly process tasks with massive computational requirements. Typically, as the volume of task data increases, the computational workload also increases accordingly, placing significant computational pressure on terminal devices with limited computational capabilities. These devices require a substantial amount of time to complete task processing.

The computational capabilities of the other two computing modes are both higher than that of the local computing mode. Therefore, for computationally intensive tasks, the latency of the local computing mode is significantly higher than that of the other two computing modes. The latency of fog computing (FC) increases more gradually, from 0.57 seconds (1 Mb) to 8.45 seconds (10 Mb); The cloud computing (CC) model proposed in this paper consistently performs the best, with latency of only 4.36s at 10 Mb, which is less than 52% of FC and 26% of LC.

CC mode has significant advantages in latency-sensitive tasks, FC is suitable for medium data volumes, and LC is only suitable for very small-scale tasks.

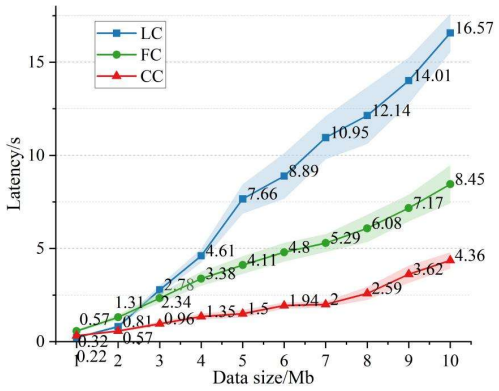


Figure 8: The latency in different computing modes

**III. C. 2) Comparison and analysis of energy consumption results**

By changing the size of the task data volume, the energy efficiency of the three computing modes was evaluated separately, and the task energy consumption simulation results are shown in Figure 9.

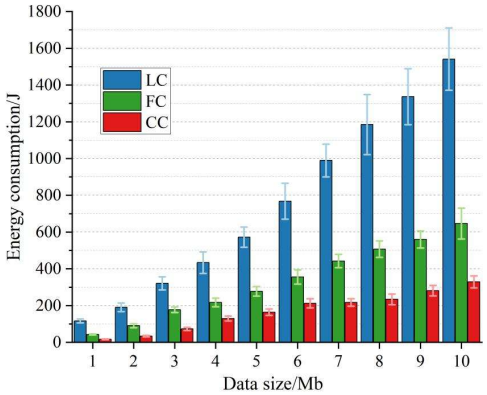


Figure 9: The energy consumption in different computing modes

As the volume of task data increases, the energy consumption results of the three computing modes also increase gradually. The energy consumption of the local computing (LC) mode is significantly higher than that of the other two computing modes, and the increase in LC energy consumption is particularly pronounced, rising from 116.74 J at 1 Mb of task data to 1,540.72 J at 10 Mb, an increase of over 12 times. The energy consumption growth of the cloud computing mode (CC) and fog computing mode (FC) is relatively slow. The energy consumption of fog computing (FC) is moderate, increasing from 42.99 J to 646.61 J, approximately 42% of LC's energy consumption. Cloud computing (CC) has the highest energy efficiency, with the lowest energy consumption throughout, reaching 329.41 J at 10 Mb, which is only 51% of FC's and 21% of LC's energy consumption.

When the task data volume does not exceed 2 Mb, the energy consumption of the local computing mode is not high, which is determined by two factors: first, the low latency of the local computing mode; second, the low power consumption of smart terminal devices. However, as the task data volume increases, when the task data volume exceeds 3 Mb, the energy consumption of the local computing mode begins to surge significantly, which is caused by the extremely high latency of the local computing mode. For the low-latency, low-power cloud computing mode, regardless of how the task data volume changes, the energy consumption of the cloud computing mode remains at the lowest level, highlighting CC's sustainable advantage in energy efficiency. Therefore, the cloud computing mode has the optimal energy efficiency among the three computing modes.

### III. C. 3) Reliability Results Comparison Analysis

Figure 10 shows the reliability simulation results of tasks with different data volumes under three computing modes. By changing the size of the task data volume, the task execution success rate of the three computing modes was simulated and calculated to obtain the reliability simulation results of the task.

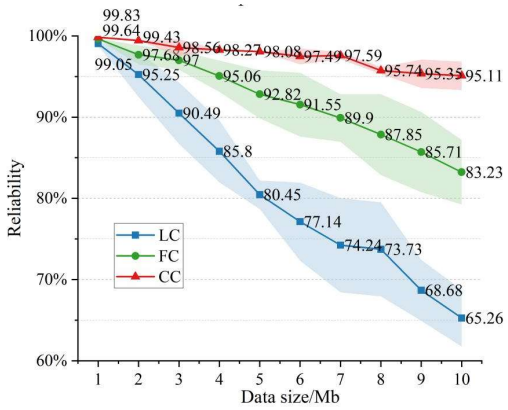


Figure 10: The reliability in different computing modes

As the volume of task data increases, the reliability of all three computing modes shows a declining trend. The reliability of the cloud computing mode decreases relatively slowly, followed by the fog computing mode, while the reliability of the local computing mode decreases at the fastest rate. The reliability of cloud computing (CC) remains above 95%, reaching 95.11% at 10 Mb, with the smallest fluctuations and the highest stability. The reliability of fog computing (FC) decreases from 99.64% (1 Mb) to 83.23% (10 Mb), with a controlled decline of 16.41%, and still maintains 92.82% at an intermediate data volume of 5 Mb. The reliability of local computing (LC) declines most significantly, dropping sharply from 99.05% to 65.26%, and falls below 80% under large task loads (≥6 Mb).

When the task data volume is small, the reliability of all three computing modes is very high, and the differences between them are not significant. When the task data volume exceeds 3MB, the reliability gap between the three computing modes gradually widens. This is because, for tasks with smaller data volumes, the latency of all three computing modes is very low, and most tasks can be completed within their maximum tolerable time. However, as the task data volume continues to increase, the reliability of all three computing modes decreases. This is because, as task latency increases, an increasing number of tasks cannot be completed within their maximum tolerable time, leading to a decrease in reliability. The reliability of the cloud computing mode is significantly higher than that of the other two computing modes. This is because the low latency advantage of the cloud computing mode results in a higher task execution success rate compared to the local computing mode and fog computing mode. Therefore, among the three computing modes, the cloud computing mode has the highest reliability.

In summary, based on the simulation results for task latency, energy consumption, and reliability, it can be concluded that as the amount of task data increases, the overall performance of the cloud computing model is superior to the other two computing models. The local computing model performs better when processing tasks with smaller data volumes, while the fog computing model's latency and reliability results fall between the two.

## IV. Conclusion

This paper addresses the bottlenecks of real-time performance, reliability, and energy efficiency in data collection and transmission in smart manufacturing scenarios and proposes an optimisation method based on cloud-edge collaboration. By integrating key technologies such as CNN data real-time processing, heterogeneous network protocol conversion, and dynamic QoS control, it achieves full-process optimisation of manufacturing data.

In metal coating thickness detection, the fusion result of 61.1146 μm obtained by the CNN-based cloud-edge collaborative fusion algorithm deviates only 0.0088% from the reference value of 61.11 μm, which is significantly better than traditional algorithms. The arithmetic mean method has an error of 0.135%, and the evidence theory method has an error of 0.036%. Furthermore, the processing delay is as low as 0.12 ms, meeting the millisecond-level real-time requirements.

The design of the service module implements three levels of dynamic bandwidth allocation: EF class: 5.8 Mbps, AF class: 2.4 Mbps, BE class: 2.1 Mbps. The queue management module ensures that high-priority services can preempt bandwidth, while reserving a minimum bandwidth (≥2.1 Mbps) for low-priority services, completely solving the problem of ‘starvation.’

In 10Mb-level intelligent manufacturing tasks, the cloud-edge collaboration model CC is comprehensively superior. In terms of latency performance, task processing only takes 4.36 seconds, which is 51.6% faster than fog computing FC and 73.7% faster than local computing LC. Energy consumption is 329.41J, which is 49.1% more energy efficient than FC and 78.6% lower than LC. The task success rate for reliability tasks is 95.11%, which is over 20% higher than FC (83.23%) and LC (65.26%).

## References


[1] Liang, S., Rajora, M., Liu, X., Yue, C., Zou, P., & Wang, L. (2018). Intelligent manufacturing systems: A review. International Journal of Mechanical Engineering and Robotics Research, 7(3), 324-330.

[2] Ma, S., Zhang, Y., Liu, Y., Yang, H., Lv, J., & Ren, S. (2020). Data-driven sustainable intelligent manufacturing based on demand response for energy-intensive industries. Journal of Cleaner Production, 274, 123155.

[3] Qu, Y. J., Ming, X. G., Liu, Z. W., Zhang, X. Y., & Hou, Z. T. (2019). Smart manufacturing systems: state of the art and future trends. The international journal of advanced manufacturing technology, 103(9), 3751-3768.

[4] Zhou, Y., Zang, J., Miao, Z., & Minshall, T. (2019). Upgrading pathways of intelligent manufacturing in China: Transitioning across technological paradigms. Engineering, 5(4), 691-701.

[5] Liu, Q., Liu, A., Li, Y., Xu, W., Liu, J., Chen, G., & Dai, W. (2017). Intelligent condition perception network towards sustainable manufacturing capability for manufacturing systems. International Journal of Manufacturing Research, 12(3), 287-304.

[6] Yin, S., Zhang, N., Ullah, K., & Gao, S. (2022). Enhancing digital innovation for the sustainable transformation of manufacturing industry: a pressure-state-response system framework to perceptions of digital green innovation and its performance for green and intelligent manufacturing. Systems, 10(3), 72.

[7] Zhang, Y., Wang, W., Du, W., Qian, C., & Yang, H. (2018). Coloured Petri net-based active sensing system of real-time and multi-source manufacturing information for smart factory. The International Journal of Advanced Manufacturing Technology, 94(9), 3427-3439.

[8] Wang, H., Wang, C., Liu, Q., Zhang, X., Liu, M., Ma, Y., ... & Shen, W. (2024). A data and knowledge driven autonomous intelligent manufacturing system for intelligent factories. Journal of Manufacturing Systems, 74, 512-526.

[9] Xu, K., Li, Y., Liu, C., Liu, X., Hao, X., Gao, J., & Maropoulos, P. G. (2020). Advanced data collection and analysis in data-driven manufacturing process. Chinese Journal of Mechanical Engineering, 33(1), 43.

[10] Wan, N., Zhang, W., Li, T., Chen, M., Song, W., Chen, H., ... & Li, R. (2023, October). Research on Data Acquisition and Real-Time Communication for Intelligent Manufacturing Training Equipment Based on Model of Things and Intranet Penetration. In Asia Simulation Conference (pp. 13-27). Singapore: Springer Nature Singapore.

[11] Qian, C., Zhang, Y., Jiang, C., Pan, S., & Rong, Y. (2020). A real-time data-driven collaborative mechanism in fixed-position assembly systems for smart manufacturing. Robotics and Computer-Integrated Manufacturing, 61, 101841.

[12] Saqlain, M., Piao, M., Shim, Y., & Lee, J. Y. (2019). Framework of an IoT-based industrial data management for smart manufacturing. Journal of Sensor and Actuator Networks, 8(2), 25.

[13] Ko, D., & Park, J. (2018). A study on the visualization of facility data using manufacturing data collection standard. The Journal of The Institute of Internet, Broadcasting and Communication, 18(3), 159-166.

[14] Chen, Z., Zheng, H., & Lin, J. (2023). Exploring Data Acquisition and Real-Time Analysis Algorithms in Smart Manufacturing with A Focus on Automation and Inspection Technologies. Scalable Computing: Practice and Experience, 24(4), 1169-1176.

[15] Ling, S., Guo, D., Rong, Y., & Huang, G. Q. (2022). Real-time data-driven synchronous reconfiguration of human-centric smart assembly cell line under graduation intelligent manufacturing system. Journal of Manufacturing Systems, 65, 378-390.

[16] Zhang, C., Zhang, J., Ji, W., & Peng, W. (2022). Data acquisition network configuration and real-time energy consumption characteristic analysis in intelligent workshops for social manufacturing. Machines, 10(10), 923.

[17] Guo, Y., Sun, Y., & Wu, K. (2020). Research and development of monitoring system and data monitoring system and data acquisition of CNC machine tool in intelligent manufacturing. International Journal of Advanced Robotic Systems, 17(2), 1729881419898017.

[18] Rajabion, L., Shaltooki, A. A., Taghikhah, M., Ghasemi, A., & Badfar, A. (2019). Healthcare big data processing mechanisms: The role of cloud computing. International Journal of Information Management, 49, 271-289.

[19] Wang, Y., Man, K. L., Lee, K., Hughes, D., Guan, S. U., & Wong, P. (2020). Application of wireless sensor network based on hierarchical edge computing structure in rapid response system. Electronics, 9(7), 1176.

[20] Sharma, S. K., & Wang, X. (2017). Live data analytics with collaborative edge and cloud processing in wireless IoT networks. IEEE Access, 5, 4621-4635.

[21] Fan, W., Zhao, L., Liu, X., Su, Y., Li, S., Wu, F., & Liu, Y. A. (2022). Collaborative service placement, task scheduling, and resource allocation for task offloading with edge-cloud cooperation. IEEE Transactions on Mobile Computing, 23(1), 238-256.

[22] Devarajan, M. V., Yallamelli, A. R. G., Kanta Yalla, R. K. M., Mamidala, V., Ganesan, T., & Sambas, A. (2024). Attacks classification and data privacy protection in cloud-edge collaborative computing systems. International Journal of Parallel, Emergent and Distributed Systems, 1-20.

[23] Fang, W., Zhu, C., & Zhang, W. (2023). Toward secure and lightweight data transmission for cloud–edge–terminal collaboration in artificial intelligence of things. IEEE Internet of Things Journal, 11(1), 105-113.

[24] Lu, J., Qu, Z., Liu, A., Zhang, S., & Xiong, N. N. (2023). MLM-WR: A swarm-intelligence-based cloud–edge–terminal collaboration data collection scheme in the era of AIoT. IEEE Internet of Things Journal, 11(1), 243-255.

[25] Shi, Y., Yi, C., Chen, B., Yang, C., Zhu, K., & Cai, J. (2022). Joint online optimization of data sampling rate and preprocessing mode for edge–cloud collaboration-enabled industrial IoT. IEEE Internet of Things Journal, 9(17), 16402-16417.

[26] Wu, Y., Yang, B., Zhu, D., Liu, Q., Li, C., Chen, C., & Guan, X. (2023). To transmit or predict: An efficient industrial data transmission scheme with deep learning and cloud-edge collaboration. IEEE Transactions on Industrial Informatics, 19(11), 11322-11332.

[27] Hu, Z., Xu, X., Zhang, Y., Tang, H., Cheng, Y., Qian, C., & Khosravi, M. R. (2022). Cloud–edge cooperation for meteorological radar big data: a review of data quality control. Complex & Intelligent Systems, 8(5), 3789-3803.

[28] Yang, H., Ong, S. K., Nee, A. Y., Jiang, G., & Mei, X. (2022). Microservices-based cloud-edge collaborative condition monitoring platform for smart manufacturing systems. International Journal of Production Research, 60(24), 7492-7501.