Applied Soft Computing 136 (2023) 110082

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Innovative soft computing-enabled cloud optimization for next-generation IoT in digital twins

Hailin Feng [a], Liang Qiao [b], Zhihan Lv [c,*]

[a] School of mathematics and computer science, Zhejiang A & F University, Hangzhou, China
[b] Qingdao University, Qingdao, China
[c] Department of Game design, Faculty of Arts, Uppsala University, Sweden

ARTICLE INFO

Article history:
Received 20 December 2021
Received in revised form 27 December 2022
Accepted 30 January 2023
Available online 10 February 2023

Keywords:
Internet of things
Cloud optimization
Edge–cloud collaboration
Edge computing

ABSTRACT

The research aims to reduce the network resource pressure on cloud centers (CC) and edge nodes, to improve the service quality and to optimize the network performance. In addition, it studies and designs a kind of edge–cloud collaboration framework based on the Internet of Things (IoT). First, raspberry pi (RP) card working machines are utilized as the working nodes, and a kind of edge–cloud collaboration framework is designed for edge computing. The framework consists mainly of three layers, including edge RP (ERP), monitoring & scheduling RP (MSRP), and CC. Among the three layers, collaborative communication can be realized between RPs and between RPs and CCs. Second, a kind of edge–cloud matching algorithm is proposed in the time delay constraint scenario. The research results obtained by actual task assignments demonstrate that the task time delay in face recognition on edge–cloud collaboration mode is the least among the three working modes, including edge only, CC only, and edge–CC collaboration modes, reaching only 12 s. Compared with that of CC running alone, the identification results of the framework rates on edge–cloud collaboration and CC modes are both more fluent than those on edge mode only, and real-time object detection can be realized. The total energy consumption of the unloading execution by system users continuously decreases with the increase in the number of users. It is assumed that the number of pieces of equipment in systems is 150, and the energy-saving rate of systems is affected by the frequency of task generation. The frequency of task generation increases with the corresponding reduction in the energy-saving rate of systems. Based on object detection as an example, the system energy consumption is decreased from 18 W to 16 W after the assignment of algorithms. The included framework improves the resource utility rate and reduces system energy consumption. In addition, it provides theoretical and practical references for the implementation of the edge–cloud collaboration framework.

1. Introduction

In the context of the rapid development of the Internet of Things (IoT), the basic network framework is faced with huge challenges because of the surge in industrial data [1]. The mass production and use of IoT has brought many security issues. Although several organizations have published guidelines for IoT use security, few IoT providers are able to properly follow these guidelines due to lack of accountability [2]. Traditional centralized cloud computing centers cannot deal with massive digital business. In this context, edge computing demonstrates its significant advantages. For example, it can perform the initial analysis of the input data on edge nodes and upload a few data that it cannot process to the cloud center (CC) for processing, which effectively reduces the storage computing and data transmission costs of edge nodes. In addition, the network resource pressures on edge computing nodes and CC are also reduced correspondingly [3]. CC possesses strong storage and computing capacities. The real-time performance of edge servers is significant, and they respond quickly and are flexible. The combination of edge servers with CC to support the 5th generation (5G) basic network can promote the rapid development of domestic manufacturing and accelerate its digital transformation [4,5].

The edge–cloud collaboration framework is a hot topic of the current research into the IoT. The advantages of the edge–cloud collaboration framework are more obvious, especially when the actual node tasks are considered [6]. After the reinforcement of the processing capacity of edge nodes, they can deal with the uploading of deep learning and other heavy tasks. In addition, actual application scenarios constrain voltage regulation frequency and chip technology to achieve flexible assignment in industry. At the moment, edge nodes unload intensive computing tasks with

* Corresponding author.
E-mail address: lvzhihan@gmail.com (Z. Lv).

https://doi.org/10.1016/j.asoc.2023.110082
1568-4946/

the constraints of their own resources [7,8]. Related literature analyses IoT data by the assignment of different user roles and further proposes a kind of IoT data analysis framework structure, which can make the maximum use of cloud resources and then generate corresponding computing models. On the edge sides, the model structure is adopted in the real-time operation on controllers [9–11].

However, heterogeneous platforms are used as the edge nodes in most of the current studies. The performances of these edge nodes are different, and programming in the deployment of application programs is very difficult. Raspberry pi (RP) can be applied in different IoT environments and network frameworks. It can not only enhance the flexibility of the operation of systems, but its hardware chip technology is more exquisite. The included IoT-edge–cloud new collaboration system offers a kind of collaboration working framework. Based on the advantages of IoT, cloud computing and edge computing can be applied in the actual manufacturing process and have good application prospects.

The main innovations of this research are as follows. 1. In the multitask scenario, this research proposes an RP edge–cloud collaboration architecture to simplify the implementation of edge–cloud collaboration. The system consists of three parts: ERP, task MSRP, and CC. To avoid heterogeneity among nodes, RP is used as the working node. Through mathematical modeling, the system working time of the proposed framework is minimized. In addition, the architecture is implemented with an edge–cloud collaborative test platform, which reduces hardware resource consumption and implementation complexity. Two representative tasks are deployed in the system: face recognition and object detection. Face recognition can be performed on one system due to the small amount of computation, while object detection requires edge–cloud collaboration due to model loading and computation. Experiments demonstrate the feasibility of this architecture. 2. An energy-minimizing task offloading mechanism for edge–cloud collaboration is proposed.

2. Related works

2.1. Current research into edge–cloud collaboration framework

In terms of edge–cloud collaboration frameworks and platforms, Yang et al. (2020) proposed a kind of open evolutionary system structure with edge–cloud collaboration intelligent cloud manufacturing systems. The slicing gateway connecting and managing workshop appliances at edge ends is introduced to support application programs sensitive to delay and to realize real-time response [12]. Alves et al. (2020) expand cloud computing by moving computing closer to end users or data sources. In consideration of edge computing, three layers of the framework are designed (cloud equipment, edge equipment, and terminal equipment layers) to meet the needs of low delay, geographical location, and energy efficiency of new IoT applications [13]. In addition, there is some research work related to platform tests in the research into edge computing or cloud computing. Cheng et al. (2022) constructed an energy-efficient model of digital hub network and a model data center network traffic prediction algorithm based on the principle of high-precision traffic prediction. They also proposed an energy-efficient multilayer virtual traffic scheduling algorithm, fused the two algorithms, and conducted an empirical study. Their study provides an improved direction for the development of IoT technology and the construction of smart cities [14]. In industrial settings, edge task nodes are needed to unload decisions according to the demands for tasks. In other words, tasks should be executed locally, by edge–cloud collaboration, or at cloud ends. The key reference index of decision downloading is time delay, which is related to the efficiency of production lines and affects the subsequent judgment results. Wang et al. (2022) designed a lightweight and fine-grained retrieval and data sharing scheme based on wireless body domain networks. The cipher text policy attribute encryption scheme was improved mainly by adding a partially hidden encryption algorithm, and the improved scheme was defined as attribute partially hidden access control. Data users can access data only when their attributes match the access policy set by the data owner. The encrypted access control policy proposed avoids excessive expenditure on computational and storage resources by the end users while enabling them to have flexible access control [15].

2.2. Current research into digital twins

Digital twins were proposed by a professor from an American university and attracted widespread attention from researchers. Zhang et al. (2020) pointed out that smart manufacturing technology is a next-generation manufacturing model with powerful learning and cognitive capabilities. Taking autonomous manufacturing cells as an implementation scenario, the author proposed a data- and knowledge-driven digital twins manufacturing cell framework that supported autonomous manufacturing through intelligent perception, simulation, understanding, prediction, optimization, and control strategies. Through the constructed digital robot, the realization method of the digital twin manufacturing cell framework was studied [16]. Mylonas et al. (2021) pointed out that intelligent manufacturing or Industry 4.0 is a trend that emerged ten years ago, aiming to utilize a technology-driven approach that revolutionizes traditional manufacturing [17]. Modern digital technologies such as the Industrial Internet of Things, big data analytics, augmented/virtual reality and artificial intelligence are key enablers of new approaches to smart manufacturing. A digital twins is an emerging concept that creates a digital replica of any physical object. To this end, the author proposes a digital twin-based remote semiphysical debugging method for an open-process intelligent manufacturing system and verifies the proposed method through a case of digital twins-based remote semiphysical debugging of a smartphone assembly line. In the digital twins of products, Moghadam et al. (2021) propose a driveline multidimensional torsional model, which is utilized to monitor the remaining service life of driveline components. By the selective collection of sensors and data in virtual models, the corresponding data collection of physical model entities is established and then optimized and adjusted continuously. Finally, virtual data were relatively consistent with actual data to provide conditions for subsequent predictive maintenance [18]. Chen et al. (2022) studied the role of digital twins technology in industrial manufacturing energy efficiency optimization, and the experimental results proved that the optimized algorithm improved industrial manufacturing efficiency and reduced energy consumption [19].

In terms of edge computing and edge–cloud collaboration framework platform issues, current methods are based mainly on simulation software demonstration without considering communication and computing among the actual equipment. In addition, no connected entity equipment platform demonstrates the results of edge computing and the deployment and unloading processes of edge computing tasks, which results in the gap in the actual demonstration development platforms of the current edge computing, the difficulty in development, low development efficiency, and the difficulty in visualization. In the research, RP, which is easy to deploy and to be developed second, is adopted to be fused into multitask scenarios. In addition, task scheduling methods and programs are utilized to realize the measurement monitoring of tasks and edge–cloud collaboration processing, which enhance

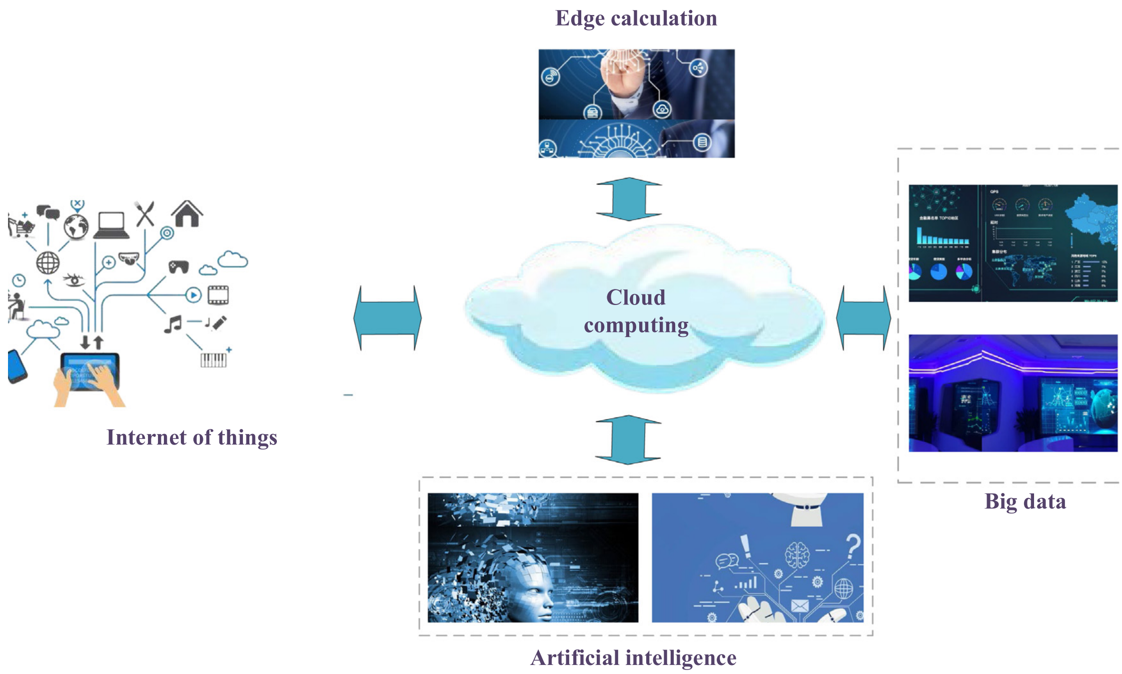


Fig. 1. The link between edge computing and the new generation of information technology.

the utility rate of computing resources. In addition, the advantages of the IoT are combined to enable edge computing and cloud computing technologies to take IoT equipment as the carriers, and the technologies can be applied in actual industrial production lines, which has reference value in the implementation of the features of collaboration edge–cloud processing, significant practical meaning, and application prospects.

3. System framework methods of new edge–cloud collaboration

3.1. Framework design of RP-based new edge–cloud collaboration

Edge computing refers to the provision of service, data, and application programs on network edges based on new basic framework technology. Compared with those of traditional centralized processing, the advantages of edge computing include the proximity of data sources to computing resources and resource allocation according to different task slices. The new generation of information technology is delivered to users from CC. Fig. 1 demonstrates the exercise between edge computing and the new generation of information technology as follows.

(1) System Models

Because the current edge computing is not implemented completely in industrial IoT, there are still many disadvantages in the deployment of the actual task framework. Therefore, a kind of edge collaboration framework is proposed based on the RP IoT platform in the research to meet the allocation needs of computing resources in different task scenarios, and the development experiment platforms are designed to evaluate the performance of the framework, which is shown in Fig. 2 as follows.

According to Fig. 2, the RP edge–cloud collaboration framework diagram mainly includes CC, multiple scheduling monitoring nodes, and multiple bottom-level edge computing nodes. One task scheduling node can connect only one bottom-level edge equipment. With the least delay emerging in the process of task completion by systems, the network structure can be divided into the following 3 layers. The first layer is the bottom level called edge raspberry pi (ERP), whose main task is the response to and implementation of edge tasks, such as smart home, object detection, and face recognition. The middle layer is monitoring and scheduling raspberry pi (MSRP), whose main task is the monitoring of RP on edge nodes at the bottom level. When edge nodes receive the tasks out of their processing scope, the middle layer requires edge nodes to transmit the tasks they cannot process to CC by the allocation scheduling task strategy. CC is at the top level. When CC receives the tasks edge nodes cannot process, it processes these tasks [20–22]. The design objectives of the system mainly include edge–cloud collaboration flexibility, time delay minimum, and priority in the processing of edge tasks. The specific processes are as follows. In multitask scenarios, edge node RP receives tasks at first and then executes corresponding tasks. In the execution of tasks, they receive the indexes needed for task execution. If edge node RP receives heavy edge tasks, the overload of edge node RP occurs. At the moment, the scheduling monitoring RP becomes aware of this and requires edge node RP to stop executing this task. After that, the scheduling monitoring RP delivers the task to the CC. Fig. 3 demonstrates its system modeling as follows.

To minimize the time delay in the task completion process, the communication network model shown in Fig. 3 is adopted in the system modeling in the research. The model consists of three layers, including M scheduling monitoring RP and N edge nodes RPs. In addition, three layers of structures, including the clod center, scheduling monitoring RP, and edge node RP, are all connected by wireless links.

(2) Local Execution Modes

Bottom-level edge node RP is at the bottom level of the platform. The layer structure is very close to task ends. Therefore, its main function is the processing of the lightweight tasks received by edge nodes. For example, the recognition and collection of facial information by cameras and face recognition with a local template according to the collected facial information. Compared with CC, the advantages of bottom-level edge node RP lie in being closer to users and the direct processing of the received lightweight data on network edges without the transmission of these data to CC for processing. In addition, this type of framework can shorten the response time of systems, guarantee data

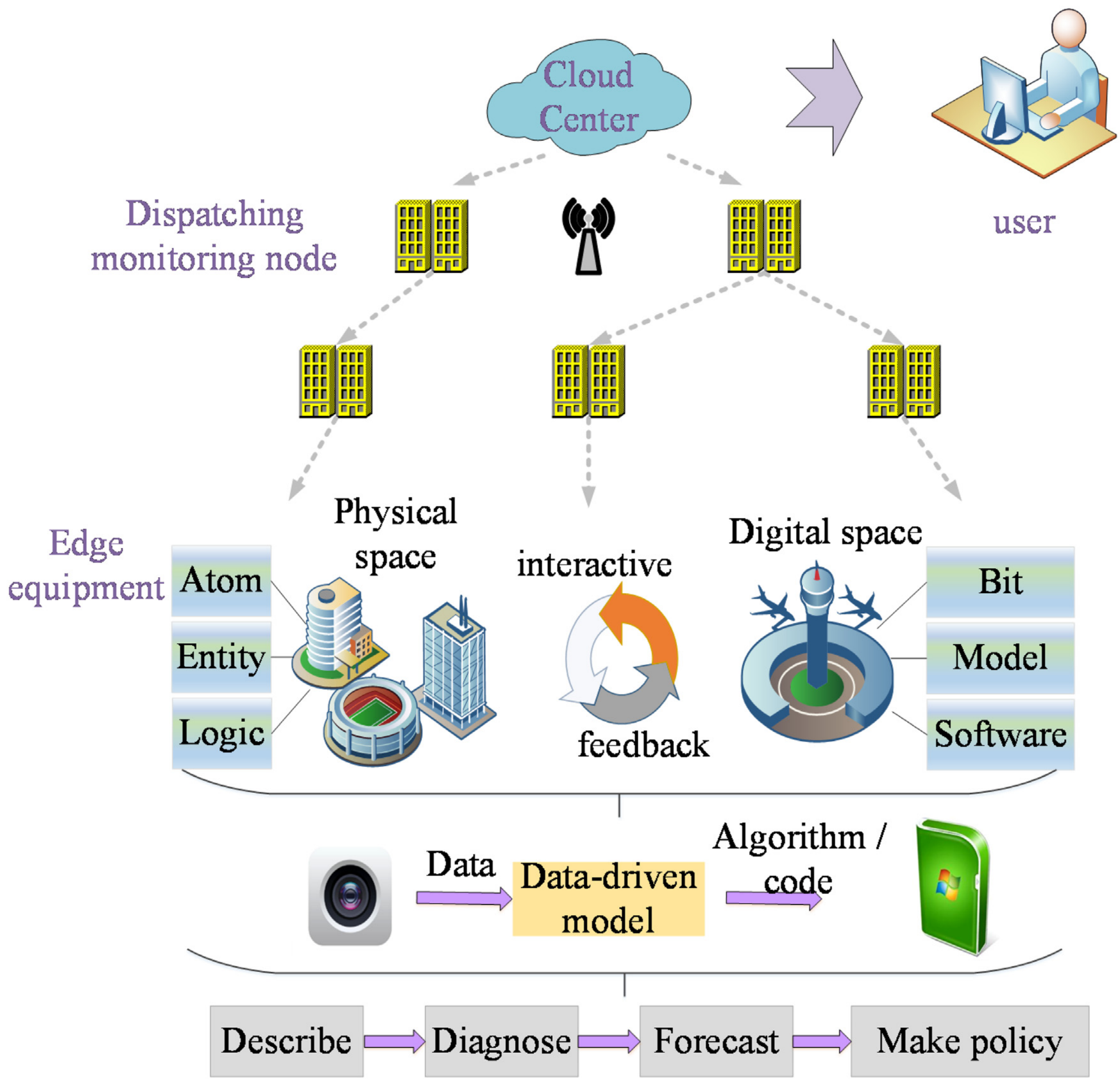


Fig. 2. RP edge–cloud collaboration framework.

security and reduce network broad bands. The bottom-level edge node RP in the platform is responsible mainly for the processing of lightweight tasks, including smart home and object detection. In the task execution process, edge node RP obtains the computing delay and task demand. The time demanded by the execution of the ith task is shown in Eq. (1) as follows.

$$T_i^{ERP} = \sum_{veV} I_{v,i} \cdot \frac{c_v}{p_i} \tag{1}$$

In Eq. (1), $I_{v,i}$ refers to the ith task performed by the bottom edge node v, whose value is between 0 and 1; p_i represents the processing speed of the edge node; and c_v represents the transmission speed of the bottom edge node v. The time required for offloading local tasks on the edge side is shown in the following equation.

$$\overline{T}_i^{ERP} = \sum_{veV} (1 - I_{v,i}) \cdot \frac{c_v}{p_i} \tag{2}$$

(3) Monitoring Scheduling Modes

The main functions of scheduling monitoring nodes include the monitoring of the computing status of bottom-level edge computing nodes, such as nodal bearing capacity and load capacity, and the arrangement and execution of tasks according to task priority. The jth task scheduling monitoring node can be expressed by MSRPf, and its value ranges between 1 and M, which is adopted to monitor the status of bottom-level edge computing nodes, including network flow usage, memory, and CPU utility [23]. It has great application value in real-time monitoring [24]. When bottom-level edge computing nodes receive the tasks out of their storage capacity, scheduling monitoring nodes play their roles in requiring edge computing nodes to upload some or all the tasks they receive to CC and then informing the could center about the processing of these tasks [25].

The time spent uploading the tasks that edge computing nodes cannot process is expressed by Eq. (3) as follows.

$$T_{i,j}^{trans} = \sum_{(C,u,v)\in E} (I_{u,j} - I_{v,i}) \cdot \frac{(d_{uv} + d_{vC})}{r_i} \tag{3}$$

In Eq. (3), $I_{u,j}$ indicates whether the bottom-level edge computing node i communicates with the task scheduling monitoring node j, whose values range between 0 and 1. d_{uv} refers to the data amount between bottom-level edge computing nodes and task scheduling monitoring nodes. d_{vC} represents the amount of data the bottom-level edge computing node i uploads to CC. r_i denotes the bandwidth received by the bottom-level edge computing node i.

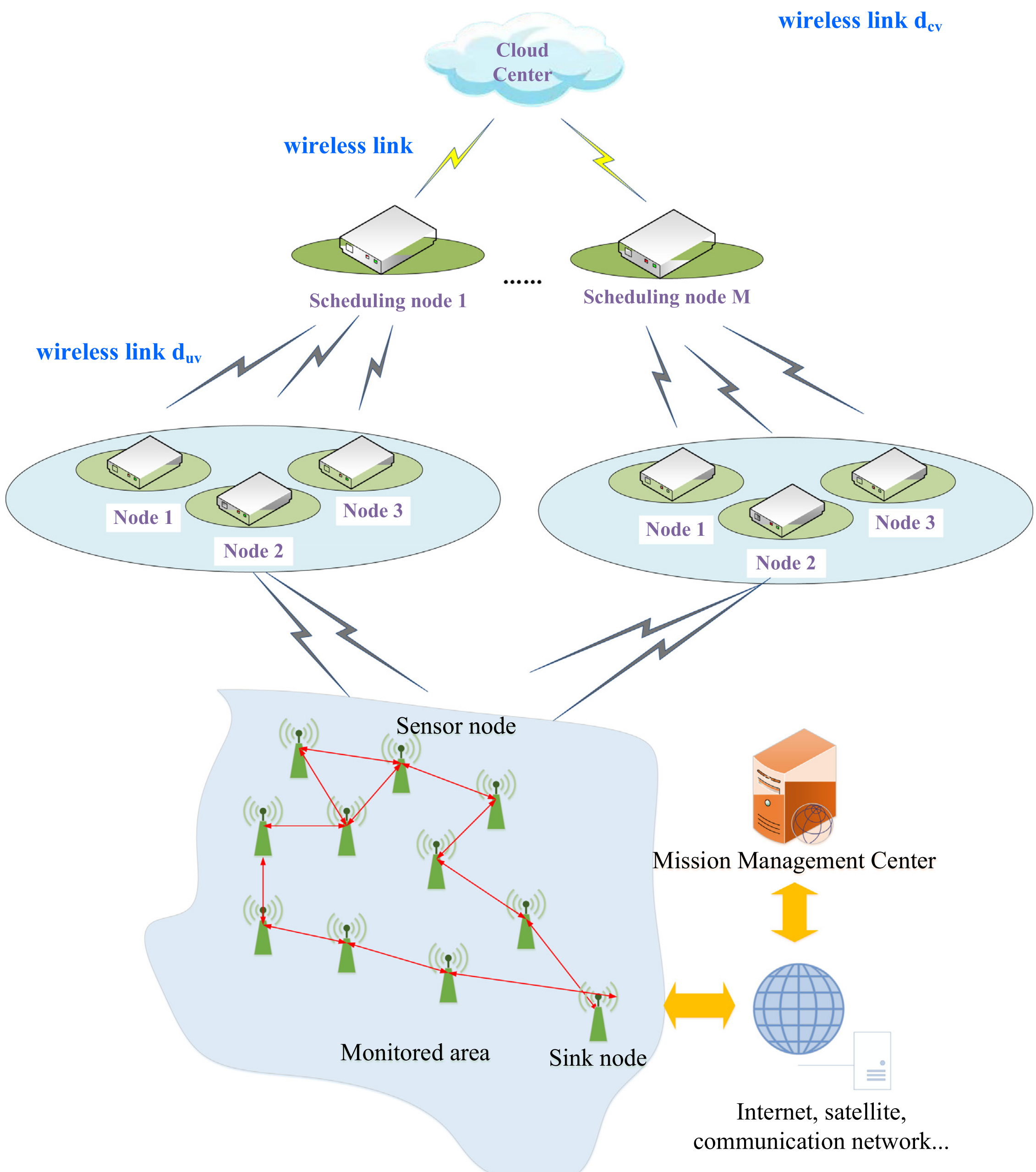


Fig. 3. System modeling.

The scheduling time spent on difficult bottom-level tasks is expressed by Eq. (4) as follows.

$$T_{i,j}^{sched} = \sum_{j=1, v\in V}^{M} (1 - I_{v,j}) \cdot \frac{C_s}{p_j} \tag{4}$$

In Eq. (4), C_s refers to the computing amount in the execution of scheduling commands, and p_j represents the computing rate.

(4) Cloud End Execution Mode

CC mainly processes the tasks that bottom-level computing nodes cannot process and then upload. Different from bottom-level computing nodes, it possesses pretentious virtualization, mainly including network application software and programs and development control platforms. Users can invoke resources according to their own needs. In addition, the computing capacity of CC is very strong. If the computing rate of the original server needs to be enhanced, the cloud computing function can be added [26–28]. After the tasks edge computing nodes cannot process are uploaded to cloud ends and processed, the results are displayed at cloud ends. The time spent on the task execution by CC is expressed by Eq. (5) as follows.

$$T_{i,j}^{cloud} = \sum_{v\in V} (1 - I_{v,i}) \cdot \frac{c_v}{p_0} \tag{5}$$

In Eq. (5), p_0 refers to the processing rate of CC.

(5) Issue Modeling

To reduce the time delay of the system framework, systems are optimized and task time delay is minimized, which are

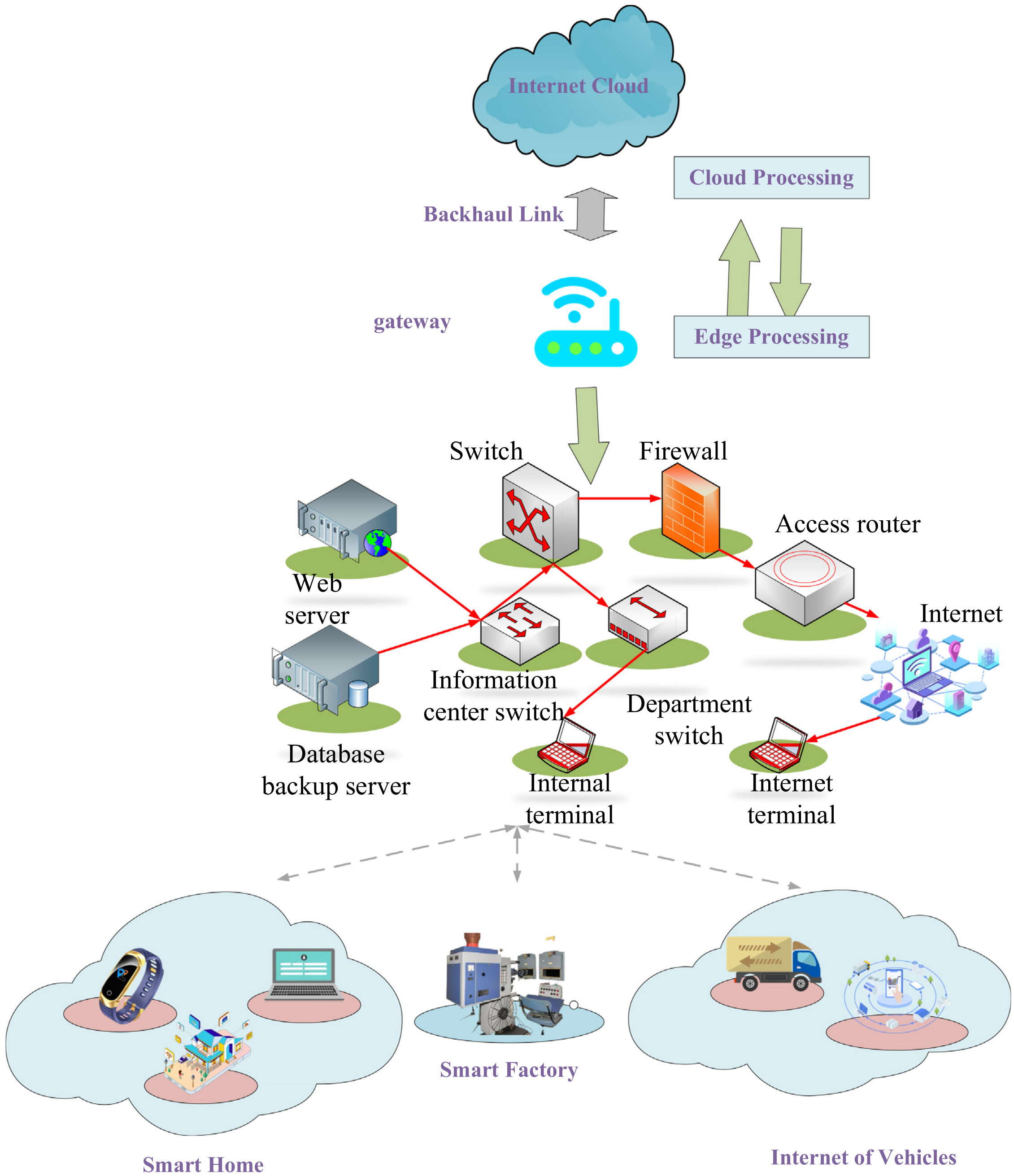


Fig. 4. Edge–cloud collaboration process.

demonstrated in Eq. (6) as follows.

$$\min_{I_{MN},r_i} \sum_{i=1}^{N}\sum_{j=1}^{M}(T_i^{ERP} + T_{i,j}^{sched} + T_{i,j}^{trans} + T_i^{cloud})$$
$$s.t. C1: \sum_{i=1}^{N} r_i \leq R$$
$$C2: I_{v,i} \leq I_{u,j}, \forall e(C,u,v) \in E, \forall i \in [1,N], j \in [1,M]$$
$$C3: \overline{T}_i^{ERP} - (T_i^{trans} + T_i^{sched} + T_i^{cloud}) > \delta, \forall i \in [1,N], j \in [1,M] \tag{6}$$

In Eq. (6), I_{MN} refers to the index signal matrix of M and N in rows and columns, and the linking number is set by users. $C1$ represents the edge bandwidth limitation. $C2$ denotes the prevention of the ping-pong effects caused by signal switching. $C3$ means the avoidance of resource competition and instruction conflict between MSRP and ERP. To ensure system reliability, the collection consisting of responsive bottom-level edge computing nodes is expressed by $\{Alive = fn_1, fn_2..., fn_N,\}$.

3.2. Research into the mechanism of energy minimization in task unloading by edge–cloud collaboration

The process of edge–cloud collaboration is displayed in Fig. 4 as follows.

After detecting the task unloading by edge equipment, CC processes the loading assignment of the task and requires edge equipment to upload the task to it. There are a total of N edge equipment and only one CC in the system. Based on the obtained information about channel status, CC can monitor the status of each edge node and process unloading decisions according to central processing unit (CPU) frequency and utility. The collection of equipment delivering edge tasks is expressed by $ED = \{ED_1, ED_2 ..., ED_N\}$. CC, which is the task receiver, is denoted by D_c, and the ith task of CC is expressed by D_{ci}. In this section, the diagram $G = (ED, E)$ is adopted to describe the connective tasks between edge computing edges and CC. ED refers to the collection of peak points. It is assumed that there is no communication link between CC and ED_j, and then $e_{ci,j} = 0$; otherwise, $e_{ci,j} = 1$. In addition, the collection consisting of edges is expressed by Eq. (7) as follows.

$$E = \left\{ e_{ci,j}, \forall ci, j \in ED \right\} \tag{7}$$

(1) System Modeling

Task Modeling: A binary model is adopted as the unloading mode between the CC and edge computing nodes, and orthogonal spectral resources are utilized. At the moment, each peak point can choose to execute tasks on edges or execute them after unloading them to CC. The channel gain between the peak point and CC center is expressed by Eq. (8) as follows.

$$g_{j,ci} = \left| h_{j,ci} \right|^2 d^{-\alpha} \tag{8}$$

In Eq. (8), $h_{j,ci}$ refers to the channel fading coefficient, d means the middle distance, and a denotes the path loss exponent.

At the moment, the rate of data transmission from the peak point to CC is expressed by Eq. (9) as follows.

$$r_{j,ci} = \log_2(1 + \frac{P^t_{j,ci} \cdot g_{j,ci}}{N}) \tag{9}$$

In Eq. (9), $P^t_{j,ci}$ refers to the power value of the transmission by the peak point, and N represents white noise:

Because edge node j is waiting for tasks, the maximum value of the CPU frequency at peak point j is $F_{j,\max}$ at the moment, and the load rate of j is δ_j. In general, the task cannot be unloaded when peak point j is executing tasks. At the moment, the value of the load rate of j ranges between 0 and 1. The remaining available processing capacity of ED_j is expressed by Eq. (10) as follows.

$$f_{j,\max} = (1 - \delta_j) \cdot F_{j,\max} \tag{10}$$

The inconvenience rate caused by resource utility is expressed as $\beta_j = \frac{1}{1-\delta_j}$.

Edge–cloud unloading energy consumption modeling: The tasks on peak point j are constructed as a tetrad $\langle I_j, O_j, W_j, \tau_j \rangle$. In this tetrad, the four letters from left to right refer to the input data amount, the output data amount, the CPU frequency demanded in task completion, whose value is closely related to the computing density and task amount, and the maximum value of the constraint of the time delay task, respectively. The time demanded by the execution of tasks by edge nodes is expressed by $T^l_j = \frac{W_j}{f_j}$, and the energy consumption in the process is expressed by $E^l_j = P^c_j \cdot T^l_j$. P^c_j denotes the CPU power of peak pint j. In addition, the energy consumption of edge side ED_j, which is based on dynamic voltage and frequency scaling (DVFS) technology, is expressed by Eq. (11) as follows.

$$E^l_j = kW_j f^2_{j,\max} \tag{11}$$

In Eq. (11), k represents a constant, whose value is related to the performance of the hardware chip structure. The time demanded by the task unloading to CC by edge computing nodes is expressed by Eq. (12) as follows.

$$T^l_{j,ci} = \frac{I_j}{r_{j,ci}} \tag{12}$$

The energy consumption during task unloading is expressed by Eq. (13) as follows.

$$E^t_{j,ci} = (P^t_{j,ci} + P^r_{j,ci}) \cdot T^t_{j,ci} \tag{13}$$

In Eq. (13), $P^r_{j,ci}$ refers to the power when the CC receives tasks, and $P^t_{j,ci}$ represents the power when tasks are delivered.

After the task from the edge computing nodes to the CC is unloaded, the CC processes the task. The time and energy consumption in the process are expressed by Eqs. (14) and (15) as follows.

$$T^e_{j,ci} = \frac{Wj}{f_{ci}} \tag{14}$$

$$E^e_{j,ci} = P^c_{ci} \cdot T^e_{j,ci} \tag{15}$$

In Eqs. (14) and (15), P^c_{ci} represents the power consumption during the processing of task i by CC. Based on the above equations, the total time delay and energy consumption in the process of task unloading by CC can be obtained, as Eqs. (16) and (17) demonstrate below.

$$T^0_{j,ci} = T^l_{j,ci} + T^e_{j,ci} \tag{16}$$

$$E^0_{j,ci} = E^l_{j,ci} + E^e_{j,ci} \tag{17}$$

(2) Issue Modeling

To enable system equipment to meet time delay constraints and to reduce energy consumption at the same time, a time delay constraint penalty mechanism is established first to meet the time delay constraints of tasks, which is expressed by Eq. (18) as follows.

$$U^T_j = \begin{cases} \varphi_j, T_j > \tau_j \\ 0, T_j \le \tau_j \end{cases} \tag{18}$$

In Eq. (18), T_j refers to the time spent on the execution of task j on ED_j, τ_j represents the tolerance time, and φ_j denotes the constant. When the time demanded by task execution does not meet the time delay requirement, the task limits the delay penalty parameters. As a result, a utility function is designed for task participants, as Eq. (19) shows below.

$$U_j = \begin{cases} U^l_j = U^T_j + E^l_j, \textit{Edge execution} \\ U^0_{j,ci} = U^T_j + E^0_{j,ci}, \textit{Uninstall to cloud for execution} \end{cases} \tag{19}$$

In Eq. (19), U^l_j is the utility value at the time of the task execution by edge nodes, and $U^0_{j,ci}$ represents the utility value at the time of the execution of remote task unloading to cloud ends. With the guarantee of the time delay constraints of tasks, the energy consumption during task execution by all edge equipment in systems is minimized, which is described by Eq. (20) below.

$$\begin{aligned} &\min_{\phi} \sum_{j=1}^{N} (\phi_{j,j} U^l_j + (1 - \phi_{j,j}) \sum_i \phi_{j,ci} U^0_{j,ci}) \\ &s.t. C1: \phi_{j,ci} = 0, \forall e_{i,j} \notin E \\ &C2: \sum_i \phi_{j,ci} = \varphi_c, \forall j \in D \\ &C3: \sum_{j=1}^{N} \phi_{j,ci} \le 1, \forall ci \in D \\ &C4: P^l_{j,ci} \le P^{\max}_j, \forall j \in D \end{aligned} \tag{20}$$

In Eq. (20), ϕ refers to the binary variables in task deployment, $\phi_{j,ci} = 0$ indicates that task j of ED_j is not unloaded to cloud ends to be processed, otherwise $\phi_{j,ci} = 1$, φ *denotes the* binary variables that describe task status, $\varphi = 0$ demonstrates that the equipment is available for use, otherwise $\varphi = 1$, and $\phi_{j,j} = 1$ reveals that the task is being processed on edges.

It is assumed that all equipment is distributed randomly within a square area with a side length of 100 m, and the simulation experimental parameters are set as follows. The transmission power from ED to CC is 200 mV, the path loss index is

3, the Gaussian background white noise is 10^{-8}, the maximum bearing capacity of the CPU is [2,10] GHz, the initial CPU load is [0,0.7], and the input task data amount is [500,2000] kB. After the independent Monteca simulation 5000 times, the simulation results are generated. Because of the rich computing resources in CC, the energy consumption during the process from task unloading to execution is effectively reduced after the unloading of the tasks of the edge node ED_j to CC.

3.3. Optimal design of the edge–cloud unloading strategy

(1) Edge Power Control

The energy efficiency (EE) function is only correlated with delivery power on the exclusive use mode. The EE function is defined as $EE = \frac{r}{(P+P_{cir})}$. P_{cir} refers to path loss. After the derivative of the delivery power P is taken by the EE function, Eqs. (21) and (22) are generated as follows.

$$\frac{\partial EE}{\partial P} = \frac{\partial \Psi(P)}{\partial (P+P_{cir})^2} \tag{21}$$

$$\Psi(P) = (P+P_{cir})\frac{g}{(1+\frac{Pg}{N})\ln 2} - \log_2(1+\frac{Pg}{N}) \tag{22}$$

In Eqs. (21) and (22), g stands for channel gain. The second-order derivative of $\Psi(P)$ is taken to generate Eq. (23) below.

$$\frac{\partial \psi}{\partial P} = -(P+P_{cir})\frac{g^2}{(1+\frac{Pg}{N})^2 \ln 2} < 0 \tag{23}$$

Based on Eq. (23), the function value decreases monotonically in the domain of definition, and the energy efficiency demonstrates the trend of increase followed by decline with the growth of the delivery power due to $\Psi(0) = \frac{gP_{cir}}{\ln 2}$, $\Psi(\infty) < 0$. Hence, an optimal task transmitting power $P_{j,ci}^{optimal}$ needs to be searched between the minimum transmitting power $P_{j,ci}^{t,\min}$ and the maximum transmitting power $P_j^{\max}$. The minimum transmitting power generated by the unloading mode with the constraints of time delay penalty factors is expressed by Eq. (24) below.

$$P_{j,ci}^{t,\min} = \frac{\left\{ 2^{\frac{l_j}{\tau_j - \frac{W_j}{f_j}}} - 1 \right\} N}{g} \tag{24}$$

To obtain the optimal transmitting power, a kind of binary search-based optimal power search algorithm is proposed in the research. Fig. 5 displays the specific steps of the algorithm as follows.

(2) Edge–cloud Unloading Strategy

To tackle the matching issue, a fast matching algorithm is proposed according to the ED optimal transmitting power, which is based on a greedy algorithm. According to the preferences for the contents of each edge ED_j, lists are created for them. After that, the preference list of edge j is labeled $list(j)$, and the content of the list is the weight values of all edges linking with ED. Fig. 6 demonstrates the specific algorithm process.

Compared with the Kuhn–Munkres (KM) algorithm, the improved greedy fast matching algorithm is more practical for distributed systems and can effectively reduce the time complexity. Fig. 7 displays its system structure.

4. Platform performance analysis

4.1. Analysis of platform performance of new edge–cloud collaboration

To evaluate the performance of the new edge–cloud collaboration platform, six typical computing tasks are deployed, including

Start

Input: $M = P_{j,ci}^{t,\min}, N = P_j^{\max}, \varepsilon = 10^{-4}$

While $N - M > \varepsilon$ **do**

$P = \frac{M+N}{2}$

Calculation $\psi(P)$

If $\psi(P) < 0$ **then**

$N = p$

Else $M = p$

End if

End while

Output: $p = P_{j,ci}^{optimal}$, P is the optimal transmission power

Fig. 5. Specific steps of the edge optimal power search algorithm.

health management, smart home, object detection, face recognition, smart factory, and licence plate recognition, which are compared on cloud only, edge only, and edge–cloud collaboration modes.

The loading time of the edge node RP when detecting the first face during operation is 2.14 s, and the display of the calculation results is relatively rapid. However, when the target tasks of another edge node RI are detected, it needs to load the single shot multibox detector (SSD) target detection training model. In this case, it will take longer to detect the first face, as shown in Fig. 8 and Table 1, which show the comparison results of platform performance in different modes.

As shown in Fig. 8 and Table 1, compared with the response time under edge-only operation or cloud-only operation mode, the response time of edge–cloud collaboration will be improved to a certain extent. Compared with the frame rate of edge-only operation, there is no significant difference between the frame rate of cloud-only operation and edge–cloud collaborative operation, but both show obvious advantages compared with the frame rate of edge-only operation It is also more fluent in the recognition of the results and can be used for real-time detection of the target.

The values 1 and 2 are adopted to denote the rate of the stream transmission method. A value of 1 indicates that the rate of the video streaming transmission method is relatively high, while a value of 2 means that the rate of the video streaming transmission method is relatively low. Finally, the comparison results of different types of video streaming transmission methods are obtained, and the energy consumption in the execution of different tasks by the system is also calculated. Fig. 9 demonstrates the comparison results and the energy consumption.

According to Fig. 9, the comprehensive comparison and analysis reveal that the video streaming transmission method of http flow+ open CV guarantees the corresponding time delay, and ERP CPU utility is significantly superior to that of other streaming transmission methods. Therefore, the method is adopted in the

Table 1
Comparison of platform performance in different modes.

Comparison factors	Task time delay (s)			Framework rate (s)		
Compare items	Edge only	Edge cloud collaboration	Cloud center only	Edge only	Edge cloud collaboration	Cloud center only
Face recognition	2.5	7.5	6.5	9.1	15.2	13.5
Object detection	20.1	13.6	16.1	1.2	14.1	13.2
Health management	9.3	3.9	9.2	2.9	6.8	7.1
Smart home	2.3	12.4	10.8	0.2	12.1	7.8
Smart factory	27.6	22.4	25.3	6.7	13.1	11.8
Licence plate recognition	20.2	12.7	15.4	1.1	14.2	13.7

Start

Input: Set C (including all unmatched edges), list j (list (j))

While $C \neq \varnothing$ **do**

for $j \in [1, N]$ **do**

if $match(j) = \varnothing$、$list(ED_j \neq \varnothing)$ **then**

find the edge with the maximum utility value in list j (best (list (j)))

If $match(best(list(j))) = \varnothing$ **then**

$match(j) = best(list(j))$

Else if $W_{ED_{j+1}} > W_{ED_j}$ **then**

$match(j) = best(list(j))$ **Cancel previous match**

Else

Keep the original match and remove the corresponding value from the preference list

End if

Else

Continue

End if

End for

End while

Output: Matching results

Fig. 6. Improved greedy fast matching algorithm.

video transmission process in the research. Because of its flexible deployment of RP and stable running of each node, the long-term adoption of this system in the deployment of IoT environments can effectively reduce the energy consumption of systems. In addition, it makes the establishment of small and medium IoT edge computing clusters easier.

4.2. Analysis of simulation results of optimal edge–cloud unloading strategy

Fig. 10 shows the comparison of the task execution performances of different unloading strategies.

In Fig. 10, the task caching and offloading (TCO) algorithm refers to energy efficiency-based task caching and unloading algorithms [29], the caching and local execution algorithm (CLA) denotes local execution and caching algorithms [30], and caching and edge cloud execution algorithms (CEA) represent edge–cloud execution and caching algorithms [31]. According to Fig. 10, the total energy consumption of system users in unloading execution is reduced with the increase in the number of users, which is caused by the decrease of available equipment with the constant increase in the number of equipment in cloud systems. Therefore, users have more choices in task execution. Compared with other algorithms, optimal matching algorithms and fast matching algorithms can ensure the highly efficient completion of tasks with time delay constraints. Fig. 11 displays the results of the comparison of algorithm performances.

In Fig. 11, the task cache and offload algorithm (TPO) refers to task effort caching and unloading algorithms, the task random cache and offload algorithm (TRO) means task random caching and unloading algorithms, and the task limited cache and offload algorithm (TFO) denotes task limited caching and unloading algorithms. According to Fig. 11, the energy consumption of the TCO algorithm is the lowest. The difference in the energy consumption between the TCO and CEA algorithms is not significant when the task data amount is small. In contrast, the difference in the energy consumption between the TCO and CLA algorithms is not obvious when the task data amount is large. In addition, energy consumption is also affected by caching capacities. The stronger caching capacity of the edge cloud means more caching tasks with less energy consumption.

To simplify the realization of edge–cloud collaboration in multitask scenarios, the optimal edge–cloud unloading strategy is tested by three representative tasks, including face identification, object detection, and multimedia. The test results are demonstrated in Fig. 12.

According to Fig. 12, the system energy-saving rate is affected by the task generation frequency. A higher task generation frequency indicates a lower system energy-saving rate, which is caused by more random tasks generated by the decrease in the number of available resources and the difficulty in the processing of tasks loaded by users in sequence.

4.3. System server test results

The average response time measurement of the system server can reflect whether the optimization and improvement algorithm can improve the running speed of the server. In the experimental process of this paper, different numbers of connection threads are established in a step-by-step manner, requests are sent to the server through these connection threads, and the throughput of the server is calculated. The test results of the system server are shown in Fig. 13:

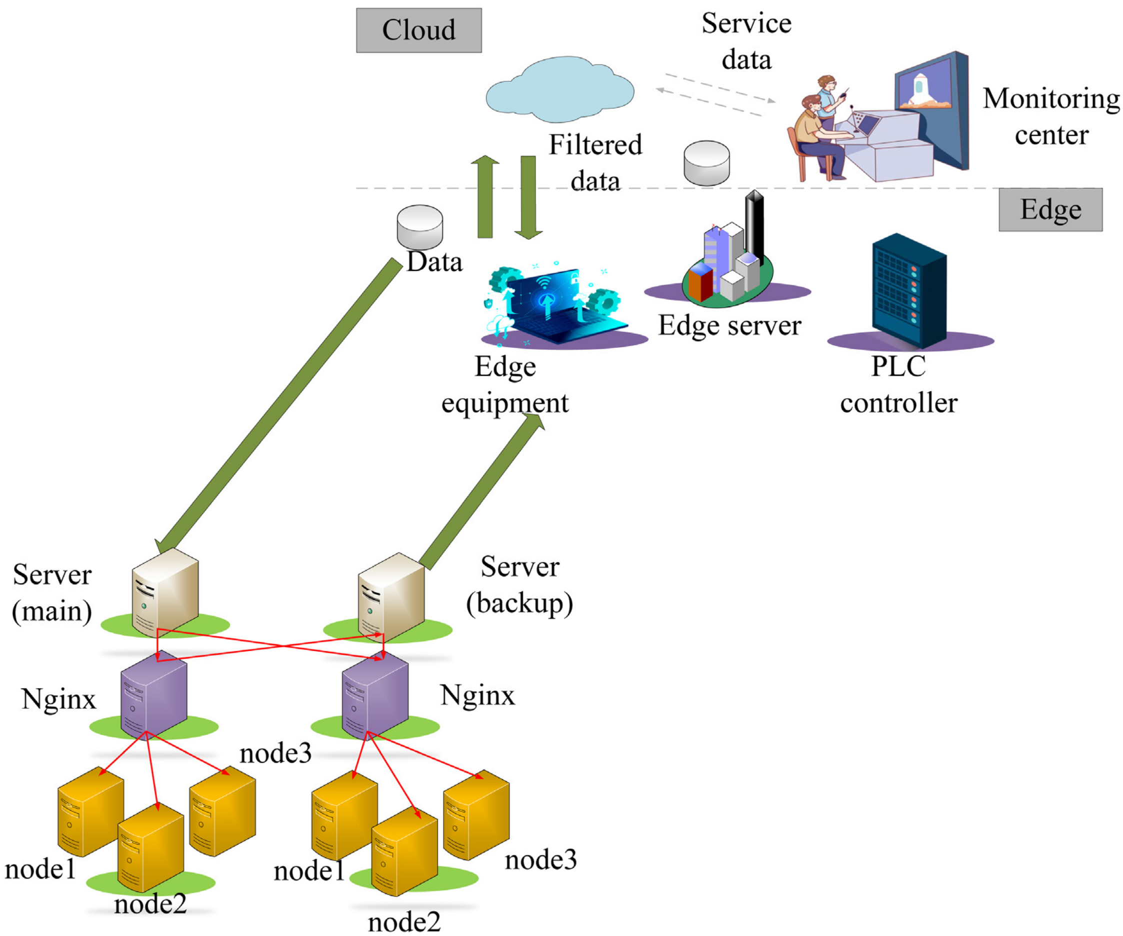
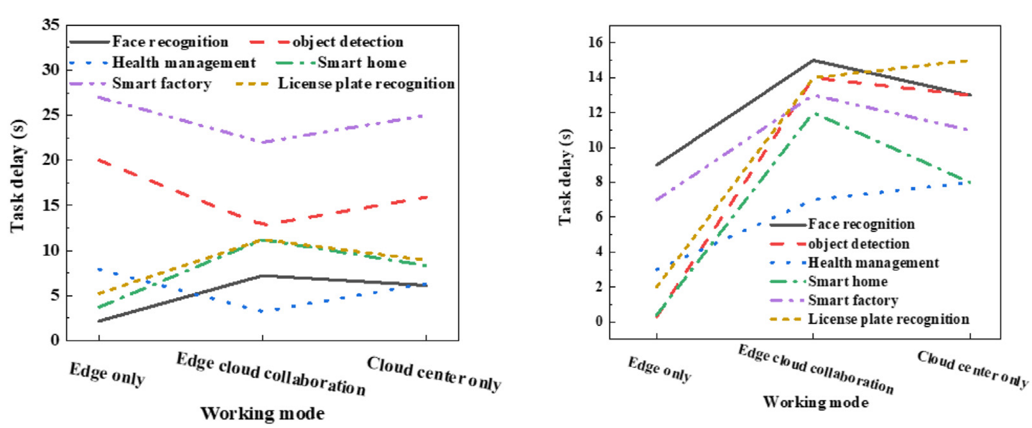


Fig. 7. System framework diagram.



(a) (b)

Fig. 8. Comparison of platform performance on different modes. (a shows the comparison of task time delay, and b demonstrates the comparison of framework rate).

As shown in Fig. 13a, the response time of TCO (min) algorithm is shorter than that of the other five algorithms, and the gap between the response time of TCO (min) algorithm and that of other five algorithms increases as the number of connections increases. As shown in Fig. 13b, with the increase of the number of connection threads, the overall throughput of TCO (min) algorithm is more and more different from that of other five algorithms. As the number of connections increases, the throughput of the TCO (min) algorithm is higher than that of the other five algorithms. Compared with the study of Nasirahmadi and Hensel (2022) [32], this work not only designs a more comprehensive algorithm model and introduces more advanced technical means, but also conducts a more comprehensive comparative study on the model, thus highlighting the advantages. By comparing with the current study results, the study in this work has achieved

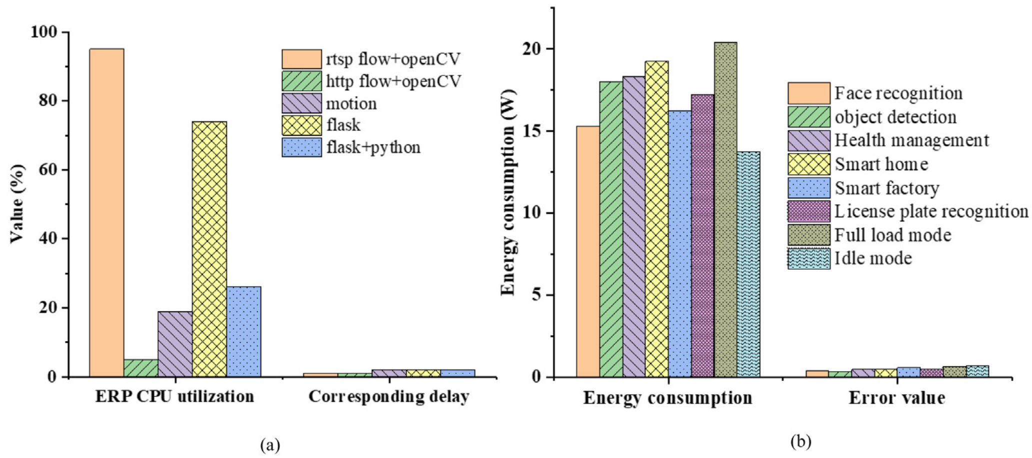


Fig. 9. Analysis of system energy consumption. (a shows different types of video streaming transmission methods, and b displays the comparison of system energy consumption).

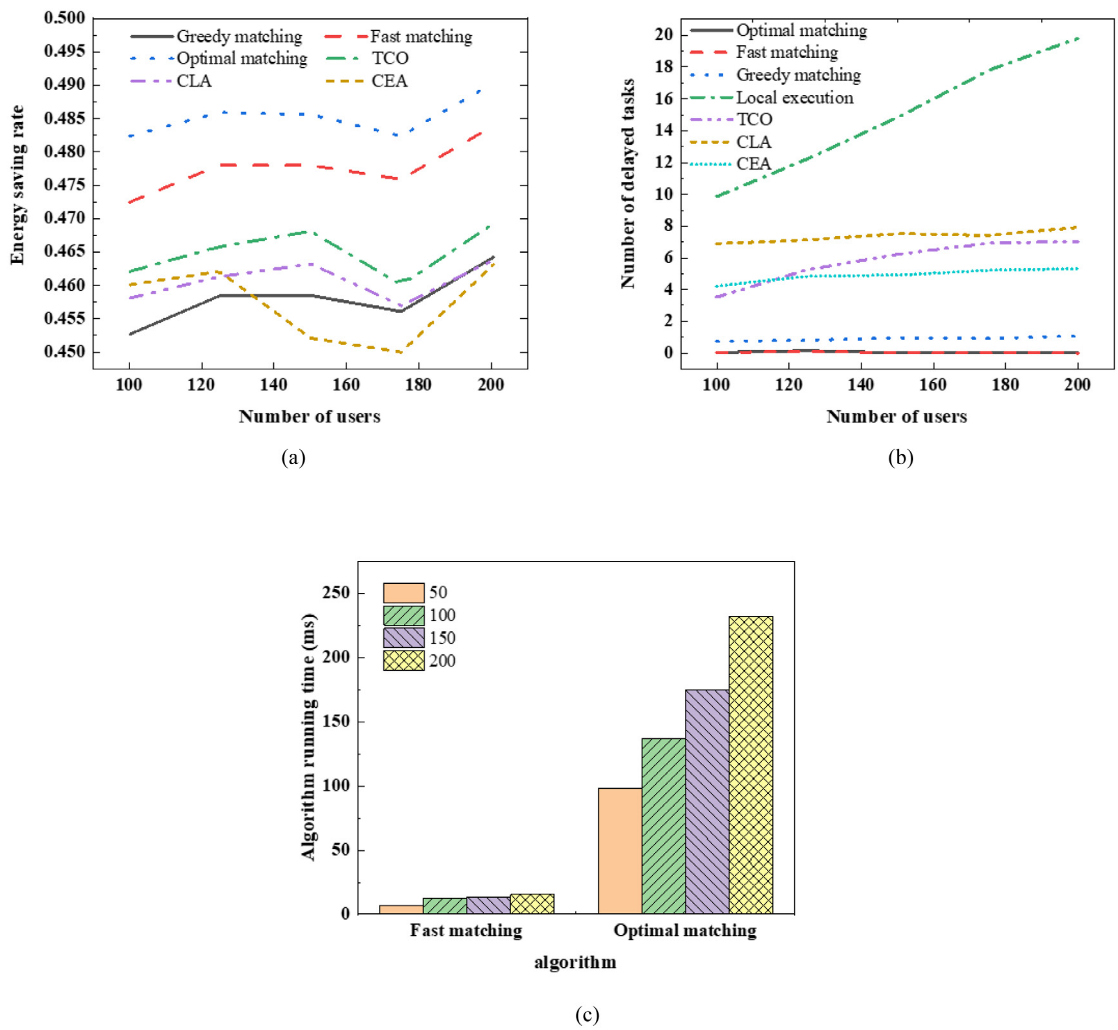


Fig. 10. Comparison of task execution performances of different unloading strategies. (a shows the change curves of energy-saving rates, b demonstrates the change curves of the number of expired tasks, and c displays the comparison of the running time of algorithms).

a greater degree of breakthrough, and provides more advanced technical methods for the future development of science and technology. When contrasting the proposed algorithm with the study of NasirahmadiHensel (2022) [32], Zhang et al. (2022) [33], Ahmad et al. (2022) [34], the experimental results are shown in Table 2:

It can be concluded from Table 2 that the algorithm proposed in this work has the best performance in terms of efficiency,

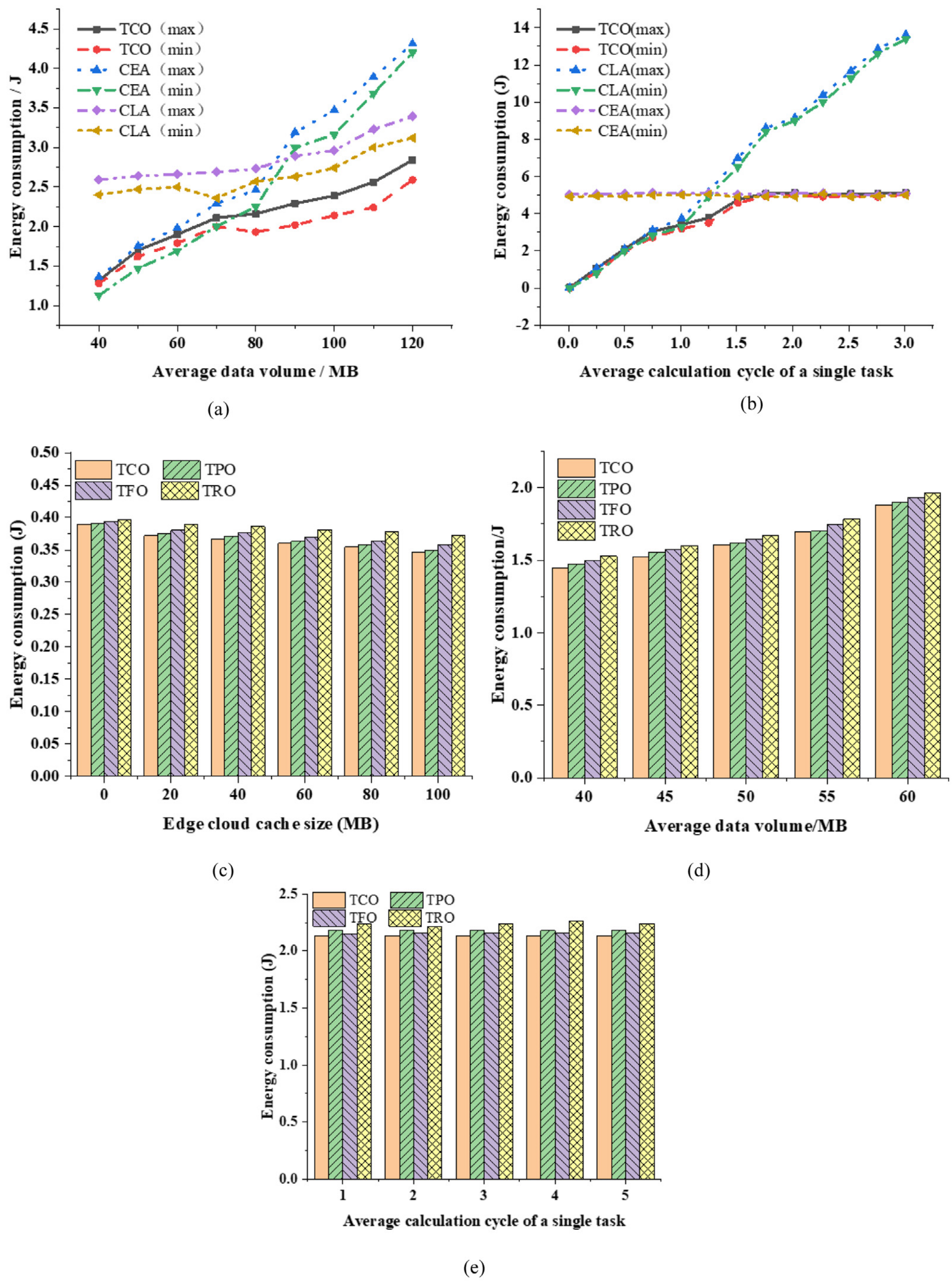


Fig. 11. Comparison of algorithm performances. (a demonstrates the changes in task data amount, b shows the changes in task request computing capacities, c presents the changes in the size of edge cloud caching, d shows the changes in task data amount, and e indicates the changes in task request computing capacities).

Table 2
Algorithm comparison.

Index	Accuracy (%)	Efficiency (s)	Coverage (%)
Algorithm proposed in this work	96	34	92
Algorithm proposed by Nasirahmadi and Hensel (2022) [32]	91	37	90
Algorithm proposed by Zhang et al. (2022) [33]	90	41	89
Algorithm proposed by Ahmad et al. (2022) [34]	89	45	89

accuracy, and coverage. This shows that the proposed algorithm not only has the excellent performance of a single algorithm, but also makes up for the deficiencies of a single algorithm, thus improving the overall effect of the system.

5. Conclusion

With the background of IoT, it is difficult to meet the requirements of the high-quality low time delay by delay-sensitive

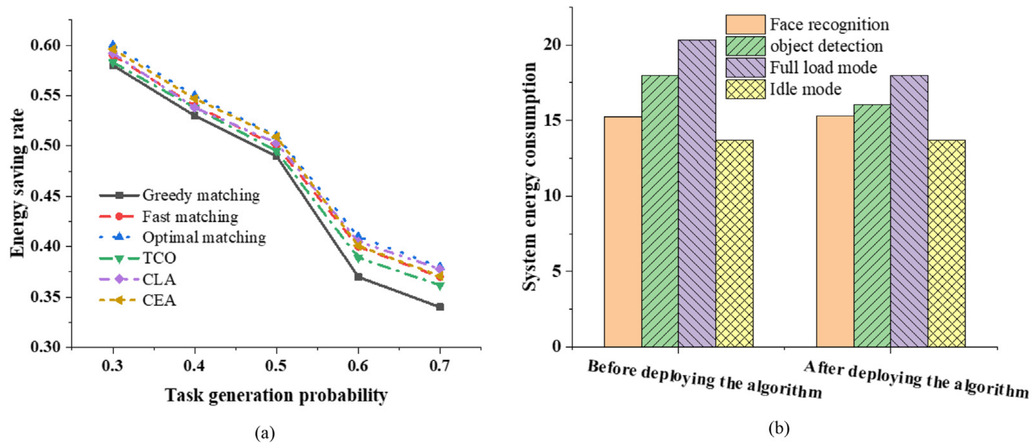


Fig. 12. Comparison of the energy-saving effects of platforms. (a shows the change curves of energy-saving rates, and b the energy-saving effects of platforms after algorithm deployment).

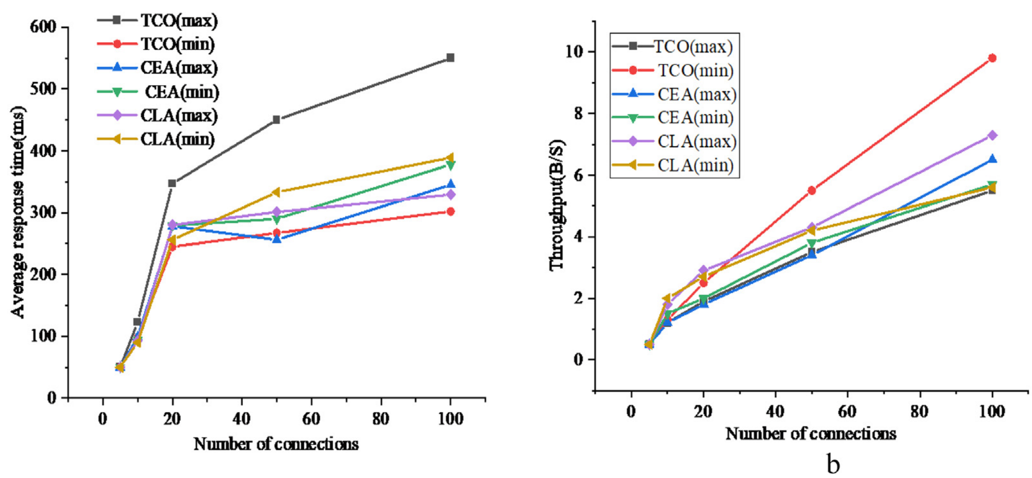


Fig. 13. System server test results. a. Response time; b. Throughput.

applications because of the long physical distance between CC and sensors. As a result, the service quality can hardly be guaranteed. In this research, cloud computing is combined with edge computing to study and design a kind of edge–cloud collaboration framework. First, RP card machines are utilized as the working nodes in multitask scenarios, and a kind of edge–cloud collaboration framework is designed for edge computing. The framework consists mainly of ERP, MSRP, and CC layers. Among each of the layers, the collaborative communication between RPs and between RP and CC can be realized by wireless networks. Second, a kind of edge–cloud matching algorithm is proposed in time delay constraint scenarios to achieve the deployment of industrial manufacturing lines by initial edge–cloud collaboration frameworks. In addition, the resource utility rate is enhanced, and the system energy consumption is saved. Due to time limits, there are still many disadvantages in the research, and further studies are necessary. During the framework construction in the research, remote command, only remote data streaming transmission, and file transmission are taken into account. In subsequent research, lightweight node tasks will be virtualized combined with KUbemet and lightweight container technologies (such as Docker).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

References

[1] R. Breillat, Industrial artificial intelligence, internet of things smart devices, and big data-driven decision-making in digital-twin-based cyber-physical production systems, Econ. Manage. Financ. Mark. 15 (1) (2020) 11.

[2] M.K. Kagita, A framework for intelligent IoT firmware compliance testing, Internet Things Cyber-Phys. Syst. 1 (2021) 1–7.

[3] W. Sun, et al., Dynamic digital twin and distributed incentives for resource allocation in aerial-assisted internet of vehicles, IEEE Internet Things J. 9 (8) (2021) 5839–5852.

[4] D. Chen, et al., Digital twin for federated analytics using a Bayesian approach, IEEE Internet Things J. 8 (22) (2021) 16301–16312.

[5] T. Liu, et al., Digital-twin-assisted task offloading based on edge collaboration in the digital twin edge network, IEEE Internet Things J. 9 (2) (2021) 1427–1444.

[6] Y. Xie, et al., Digital twin for cutting tool: Modeling, application and service strategy, J. Manuf. Syst. 58 (2021) 305–312.

[7] S. Jamil, M.U. Rahman, A comprehensive survey of digital twins and federated learning for industrial internet of things (IIoT), internet of vehicles (IoV) and internet of drones (IoD), Appl. Syst. Innov. 5 (3) (2022) 56.

[8] K. Takahashi, et al., Development of edge-node map based navigation system without requirement of prior sensor data collection, J. Robot. Mechatronics 32 (6) (2020) 1112–1120.

[9] F. Paoluccic, P4 edge node enabling stateful traffic engineering and cyber security, IEEE/OSA J. Opt. Commun. Networking 11 (1) (2019) 84–95.
[10] W. Zhao, et al., ETC-IoT: Edge-node-assisted transmitting for the cloud-centric internet of things, IEEE Netw. 32 (3) (2020) 101–107.
[11] H. Elayan, M. Aloqaily, M. Guizani, Digital twin for intelligent context-aware IoT healthcare systems, IEEE Internet Things J. 8 (23) (2021) 16749–16757.
[12] C. Yang, et al., Big data driven edge-cloud collaboration architecture for cloud manufacturing: a software defined perspective, IEEE Access 8 (1) (2020) 45938–45950.
[13] M.P. Alves, et al., LW-CoEdge: a lightweight virtualization model and collaboration process for edge computing, World Wide Web 23 (2) (2020) 1127–1175.
[14] C. Cheng, J. Dou, Z. Zheng, Energy-efficient SDN for internet of things in smart city, Internet Things Cyber-Phys. Syst. 2 (11) (2022) 145–158.
[15] K. Wang, S. Xie, J. Rodrigues, Medical data security of wearable tele-rehabilitation under internet of things, Internet Things Cyber-Phys. Syst. 2 (2) (2022) 1–11.
[16] C. Zhang, et al., Manufacturing blockchain of things for the configuration of a data-and knowledge-driven digital twin manufacturing cell, IEEE Internet Things J. 7 (12) (2020) 11884–11894.
[17] G. Mylonas, et al., Digital twins from smart manufacturing to smart cities: A survey, IEEE Access 9 (2021) 143222-143249.
[18] F.K. Moghadam, G. Rebouas, A.R. Nejad, Digital twin modeling for predictive maintenance of gearboxes in floating offshore wind turbine drivetrains, Forschung Ing. 85 (2) (2021) 273–286.
[19] R. Chen, H. Shen, Y. Lai, A metaheuristic optimization algorithm for energy efficiency in digital twins, Internet Things Cyber-Phys. Syst. 2 (2022) 159–169.
[20] W. Jiang, et al., Three-dimensional photocatalysts with a network structure, J. Mater. Chem. A 5 (12) (2020) 5661–5679.
[21] Y. Lu, S. Maharjan, Y. Zhang, Adaptive edge association for wireless digital twin networks in 6G, IEEE Internet Things J. 8 (22) (2021) 16219–16230.
[22] J. Vrana, Industrial Internet of things, digital twins, and cyber–physical loops for NDE 4.0, Handb. Nondestr. Eval. 4.0 3 (1) (2021) 1–34.
[23] I. Volkov, G. Radchenko, A. Tchernykh, Digital. Twins, Internet of things and mobile medicine: A review of current platforms to support smart healthcare, Program. Comput. Softw. 47 (8) (2021) 578–590.
[24] T. Benhmad, A.B. Abdennour, A. Darghouthi, et al., Remote control of environmental parameters in rabbitry based on IoT, Internet Things Cyber-Phys. Syst. 2 (2022) 111–119.
[25] P. Subramanian, V. Babu, R. Karthikeyan, Remote monitoring of elderly patients and infants and dispatching of emergency services using Internet of Things, Int. J. Pure Appl. Math. 119 (16) (2019) 2109.
[26] X. Li, et al., Big data analysis of the internet of things in the digital twins of smart city based on deep learning, Future Gener. Comput. Syst. 128 (2022) 167–177.
[27] A. Fuller, et al., Digital twin: Enabling technologies, challenges and open research, IEEE Access 8 (2020) 108952-108971.
[28] C. Mouradian, et al., An IoT platform-as-a-service for NFV based – Hybrid Cloud/ Fog Systems, IEEE Internet Things J. 7 (7) (2020) 6102–6115.
[29] J. Meshkati, F. Safi-Esfahani, Energy-aware resource utilization based on particle swarm optimization and artificial bee colony algorithms in cloud computing, J. Supercomput. 75 (5) (2019) 2455–2496.
[30] A.R. Al-Ali, et al., Digital twin conceptual model within the context of internet of things, Future Internet 12 (10) (2020) 163.
[31] M. Jacoby, T. Usländer, Digital twin and internet of things—Current standards landscape, Appl. Sci. 10 (18) (2020) 6519.
[32] A. Nasirahmadi, O. Hensel, Toward the next generation of digitalization in agriculture based on digital twin paradigm, Sensors 22 (2) (2022) 498.
[33] H. Zhang, H. Xiong, J. Xu, Dynamic simulation research on the effect of governance mechanism on value co-creation of blockchain industry ecosystem, Sustainability 14 (1) (2022) 12.
[34] R.W. Ahmad, K. Salah, R. Jayaraman, et al., Blockchain in oil and gas industry: Applications, challenges, and future trends, Technol. Soc. 68 (1) (2022) 14.