Scientific Programming

Intelligent Mobile Edge Computing for Smart Internet of Things: Architecture, Algorithm, and Application

Lead Guest Editor: Jiwei Huang

Guest Editors: Ying Chen, Phu Thinh Do, and Xu Zhang

Intelligent Mobile Edge Computing for Smart Internet of Things: Architecture, Algorithm, and Application

Scientific Programming

Intelligent Mobile Edge Computing for Smart Internet of Things: Architecture, Algorithm, and Application

Lead Guest Editor: Jiwei Huang
Guest Editors: Ying Chen, Phu Thinh Do, and Xu Zhang

Contents

Hindawi
Scientific Programming
Volume 2022, Article ID 1163177, 15 pages
https://doi.org/10.1155/2022/1163177

Research Article

CSO-DRL: A Collaborative Service Offloading Approach with Deep Reinforcement Learning in Vehicular Edge Computing

Yuze Huang, Yuhui Cao, Miao Zhang, Beipeng Feng, and Zhenzhen Guo

School of Information Science and Engineering, Chongqing Jiaotong University, Chongqing 400074, China

Correspondence should be addressed to Yuze Huang; huangyz@cqjtu.edu.cn

Received 3 April 2022; Revised 5 August 2022; Accepted 9 August 2022; Published 5 September 2022

Academic Editor: Ying Chen

In vehicular edge computing, vehicles move along the road and request the services from the nearest edge servers with low latency. Due to the limitation of computation capacity of vehicular devices, the services should be offloaded on RSUs equipped with edge servers to provide service with low latency. Noticed that the location of service offloading may affect the service requesting delay directly, and it may exist some interrelationship between interacting services; all of these are rarely considered in recent studies. To address such problems, we propose a collaborative service offloading approach with deep reinforcement learning in vehicular edge computing named CSO-DRL. Our approach first divides the road segments by k-means-based algorithm through analyzing the trajectory data of vehicles, and then the offloading location is determined by observing the vehicle running status. Secondly, the interacting services are discovered by a parallel frequent pattern-based algorithm efficiently. Furthermore, the collaborative service offloading algorithm is presented by the DDPG model for offloading the interacting services, which can minimize the service requesting delay and data communication delay between interacting services. Finally, the efficiency of the algorithm is evaluated by real-world data-based simulation experimental evaluations. The results show our algorithm can obtain a lower delay than other baseline algorithms in searching for the optimal service offloading strategy.

1. Introduction

With the development of intelligent transportation technologies, the internet of vehicles (IoV) has raised public attention and applications in the urban transportation system, which can be regarded as the evolution of vehicular ad hoc networks. In IoV, the vehicles equipped with intelligent devices are capable of vehicles to everything communication (V2X) through the wireless network [1]. Through the V2X communication, the vehicles can realize the information interaction among vehicles, roadside units (RSUs), and humans to assist the decision of auto-driving and avoid the traffic accidents [2]. In the IoV-based intelligent transportation system, the data including vehicles status information and traffic information are collected by numerous sensors for analysis and computation [3]. Since the vehicles have limited computation capability to execute complex computation tasks, the most direct approach is transmitting the traffic data to a cloud data center for computing. Generally speaking, a cloud data center is far away from the vehicles, which may bring some problems such as network congestion, time-consuming, and data privacy leaking [4]. With the complexity of urban traffic, the response time of cloud service will be increasingly difficult to satisfy the real-time decision for auto-driving. To solve such problems, edge computing regarded as the supplement of traditional cloud computing is used in IoV, which can efficiently provide service to users with low latency [5, 6].

In edge computing, the services are offloaded on edge servers (ESs) in close proximity to users, which mainly undertake the capacity of data computing and most services executions. While the intelligent devices are only responsible for the services of raw data preprocessing and few non-computation-intensive services executions, and then only a small amount of intelligent computation services are delivered to remote cloud servers for execution [7–9]. As an efficient approach, service offloading (also called computation offloading) can address the problems of insufficient devices and the high latency for cloud services, which has raised public attention in edge computing [10–12]. During the service offloading, the users' experience may be greatly affected by service requesting delays and the energy consumption of intelligent devices [13, 14]. In such conditions, some offloading approaches are produced to minimize the latency and energy, but some defections are revealed in IoV service offloading.

In vehicular edge computing, when the vehicles move along the roads, the vehicles request the services from the nearest edge server to gain low-latency services. Due to the coverage limitation of the RSUs equipped with edge servers, when the vehicles reach the boundaries of edge servers, the edge servers must be switched to guarantee the continuity of service, so the selection of edge servers may influence the delay of service requesting to a large extent [15]. Thus, the speed and location of vehicles must be considered in service offloading. Moreover, the interrelationship between services is another factor that must be considered in service offloading. Many services collaborate with different services to complete a certain complex business goal, which may produce lots of transmission data among the interacting services. If the interacting services are offloaded separately, the data communication delay between them will be increased. As the number of data increases exponentially, the communication delay between interacting services becomes a nonignored problem for service offloading. Throughout the studies for services offloading [16, 17], although some works have a focus on the collaborative problem of cloud and edge servers [13, 18], a few studies considered the collaboration of services. In summary, the mobility of vehicles and the interrelationships between services bring new challenges for service offloading.

To address the mentioned challenges, a collaborative service offloading approach named CSO-DRL is proposed to search optimal offloading strategy for interacting services. In our approach, an algorithm for offloading location selection is presented to consider the impact of speed and location of vehicles. This algorithm divides the road segments by analyzing the trajectory of vehicles, and then the location of offloading is determined by the status of vehicles. Furthermore, the interacting services are discovered based on a parallel frequent pattern mining algorithm. Finally, we constructed the system models and put forward a collaborative offloading algorithm based on deep reinforcement learning to offload the interacting services while considering the optimization problem for minimizing the service requesting delay and data communication delay between interacting services. Specifically, the contribution of this paper is threefold as follows:

(i) An offloading location decision algorithm is presented by analyzing the trajectory of vehicles, which divides the road segments with the k-means clustering algorithm first, and then the offloading location is determined via the status of vehicles.

(ii) In order to reduce the service requesting delay and data communication delay, the interacting services are discovered by a parallel frequent pattern-based algorithm first, and then the offloading models are constructed by analyzing the communication environments and the computation models.

(iii) The collaborative service offloading algorithm with deep deterministic policy gradient is put forward to obtain the optimal service offloading strategy. This algorithm reveals the interrelationship of interacting services by minimizing the service requesting delay and data communication delay in service offloading.

The rest of this paper is organized as follows. Section 2 introduces the related work of this research. Section 3 presents an overall framework of the collaborative service offloading, and then an offloading location decision algorithm and the system model based on the interacting services discovered are produced in this section. Furthermore, a collaborative service offloading algorithm with deep deterministic policy gradient (DDPG) model is proposed to offload the interacting services in Section 4, which can realize the optimization of minimizing the service requesting delay and data communication delay between interacting services. Finally, the experimental evaluations are conducted in Section 5, and then we conclude this paper in Section 6.

2. Related Work

With the advent of the internet of everything, the data produced and collected by the various sensors in IoT are experiencing a steep rise [19–21]. Though the traditional cloud computing paradigm can solve the inefficient computation capacity of intelligent devices, it can also bring some problems, such as high latency, network congestion, and data privacy leaking. In order to address such problems, a new computing paradigm named edge computing is introduced [22]. In edge computing, the data are preprocessed by intelligent devices, and the edge servers nearby users undertake the majority of tasks for computing data and executing the services, while the remote cloud servers are only responsible for the training of deep neural network models [23, 24]. Due to the resource-constrained of end devices, the services should be offloaded on edge servers or cloud servers [25].

Service offloading (also called computation offloading) aims to address the inefficient storage capacity, computing resources, and energy of numerous devices at the user level [26]. In the depth of offloading researches, the service offloading mainly contains offloading decisions and offloading optimization. Offloading decision is mainly studies on whether the service should be offloaded or not, and which service should be offloaded. The main studies can be divided into two types, which are 0/1 offloading and partial offloading [27, 28]. 0/1 offloading means the service can be executed either locally or on the other devices fully, which cannot archive the optimal delay. In partial offloading, the service can be split into a quantity of subservices. The offloading strategy is responsible for determining which subservices should be executed locally and which part should be executed on edge or cloud. During the service offloading, latency and energy are the main factors considered by researchers. For time-sensitive services, for example, VR gaming and multimedia services, the researchers mainly focus on how to reduce the latency [16]. Some studies have examined the optimization of latency and energy together, which is difficult to search for the balance optimal result in latency and energy consumption [29]. Some studies noticed that deep reinforcement learning (DRL) is a better way to search for the optimal offloading strategy than the traditional optimization algorithm. Since then, the task offloading approaches with deep reinforcement learning are increasingly, and some efficient approaches based on various DRL algorithms are introduced to get the optimal latency or energy [30–32]. Besides these studies, the load balance brings new challenges for resource allocation; some efficiency approaches are presented to address these challenges [33, 34].

Internet of vehicles (IoV) is a typical IoT, which connects vehicles and RSUs through wireless communication technologies. As the number of vehicles increases exponentially, as a complex system, the cloud computing-enabled IoV cannot meet the real-time demands for vehicular service, while vehicular edge computing (VEC) can provide lower data process delay and a more stable network transmission rate. Some studies have introduced for VEC framework and task scheduling in VEC [35, 36]. In IoV, since the vehicles move along the roads, the task offloading in VEC is a dynamic process. The new challenges for task offloading in VEC have been produced, and some efficient approaches have been introduced. Wang et al. [37] put forward a solution for task offloading in the fog-based IoV system, which decomposes the offloading optimization problem into two subproblems and schedules the traffic queue between different edge servers, the experimental evaluation results show that the algorithm can obtain the lower latency than other algorithms. In [38], the privacy conflicts of offloading are considered, and the NSGA-II algorithm is designed to solve the multiobjective optimization for reducing the service computation delay and the energy consumption of devices while guarding against privacy conflicts. He et al. [39] considered the QoE of services, and then a DRL-based algorithm is introduced to save energy consumption and improve the QoE value.

In the sight of these studies, few works have examined the location selection and the data communication delay between interacting services, which are the nonignored factors for service offloading. In view of the importance of location decisions for offloading and the interrelationships between services, this paper proposes a collaborative service offloading approach with deep reinforcement learning to address the above challenges.

3. System Model and Problem Formulation

In this section, we present the framework of the collaborative service offloading approach first, and then the algorithm for offloading location selection is produced. Finally, the system model, contained communication model, and computation model for our approach are put forward to formulate the offloading problem, which should be solved in the following sections.

3.1. Framework of Collaborative Service Offloading. In vehicular edge computing, the vehicular devices collect the data from numerous sensors and then transmit the preprocessed raw data to RSUs equipped with edge servers. The RSUs receive the data and execute the service to send back the decisions to vehicle users. Due to the limitation of computation capacity of vehicular devices, the computing-intensive services should be offloaded to the edge servers, and the vehicular devices only undertake the task of preprocessing the raw data and execute the noncomputing-intensive services. Thus, how to offload the services with low latency is important for vehicular edge computing.

As Figure 1 shows, the vehicles move along the road and communicate with RSUs through a wireless network. Due to the coverage limitation of RSU, the vehicle cannot request the service from a fixed RSU. When the vehicle moves at the boundary of RSU, the network should be handoff and the vehicle accesses another RSU nearby the vehicle. Therefore, the mobility of the vehicles may affect the service requesting a delay. In the sight of the studies for service offloading, most of them focus on the optimization of service requesting a delay, and few of them study the location selection for service offloading. In order to obtain a lower delay, during the process of service offloading, the location selection for offloading is an important problem, which must be considered. Besides offloading location selection, the interrelationship of services is another nonignored problem for service offloading. We noticed some services may interact with each other to complete a complex business goal, which may produce lots of transmission data among them. If the interacting services are offloaded separately, the data communication delay between them will be increased. Thus, the data communication between these interacting services is a nonignored factor.

In order to address the problems mentioned above, we propose a collaborative service offloading approach with deep reinforcement learning named CSO-DRL to offload the interacting services. As Figure 2 shows, the k-means-based road segments dived algorithm is presented first, and then the suitable RSU for offloading is selected based on the mobility of vehicles. Furthermore, the interacting services are discovered by analyzing the service requesting log to reveal the relationship between interacting services. Finally, a DDPG-based collaborative service offloading algorithm is presented for offloading the interacting services, which can optimize the service requesting delay and data communication delay between interacting services, the detailed process can be expressed as follows.

Step 1. The trajectory data of vehicles are collected, and then a k-means-based road segments dividing algorithm is put forward to obtain the road segments information. For each segment, an RSU is deployed as the location of service offloading in this segment.

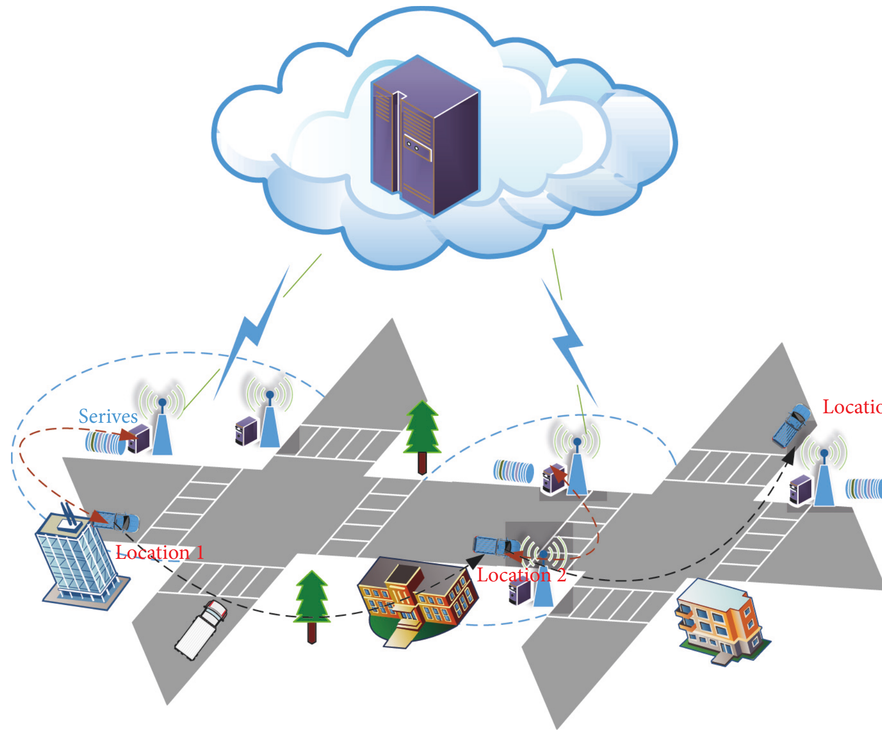


FIGURE 1: Vehicular edge computing scenario.

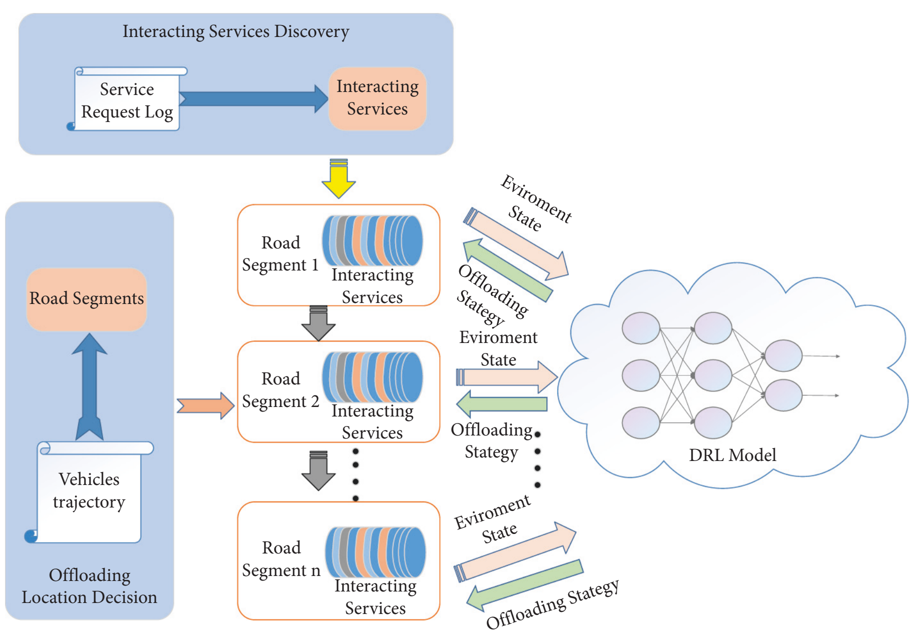


FIGURE 2: Framework of CSO-DRL in VEC.

Step 2. A vehicle is randomly selected to observe the running status. When the vehicle reaches the boundary of this segment, the RSU should not be switched to reduce the time consumption produced by network handoff; otherwise, another RSU is selected for offloading.

Step 3. The service requesting log is exacted to discover the interacting relationship for services. A parallel frequent pattern-based algorithm is presented to mine the frequent service 2-itemsets for interacting services, which can be found in our previous works [40, 41].

Step 4. A DDPG-based collaborative service offloading algorithm is proposed to offload the interacting services, which can optimize the service requesting delay and data communication delay between interacting services. In this

Input. Data set of vehicles trajectory V_t; number of road segments k
Output. Road segments $R = \{r_1, r_2, \ldots, r_k\}$
Select the k samples as the initial vector $\{u_1, u_2, \ldots, u_k\}$.
while True **do**
  num = 0
  for $i = 0, 1, \ldots, k$ **do**
    $C_i = \Phi$
  end for
  for $j = 1, 2, \ldots, n$ **do**
    Compute the distance of v_j from each vector $u_i\,(1 \leq i \leq k)$ according to equation (1).
    Determine the clusters' label according to the nearest vector, $\tau_j = \text{argmin}_{i\in\{1,2,\ldots,k\}} \text{d}_{ji}$
    $r_{\tau_i} = r_{\tau_i} \cup v_j$
  end for
  for $i = 1, 2, \ldots, k$ **do**
    $u_i' = 1/C_i \sum_{x\in C_i} x$
    if $u_i' \neq u_i$ **then**
      $u_i = u_i'$
    else
      num + +
    end if
  end for
  if num $= k$ **then**
    break
  end if
end while
return$R = \{r_i, r_2, \ldots, r_k\}$

Algorithm 1: Algorithm for dividing road segments.

algorithm, the training model is deployed on the cloud servers to receive the vehicle running environment state, and then the optimal offloading strategy is obtained and performed after numerous interaction computing.

3.2. Offloading Location Decision. In VEC, the mobility of vehicles may affect the service requesting a delay. Due to the coverage limitation of each RSU, the vehicles access the different RSUs when the vehicles reached different locations. The selection of offloading location is an important problem, which should not be ignored during the service offloading. In this section, we put forward a service offloading location selection approach to assist service offloading action decisions.

In reality, it is difficult to deploy the edge servers on all the RSUs, as it is a time-consuming and resource-consuming process. Hence, a k-means-based algorithm is introduced to divide the road segment and select the RSU to deploy the edge server as the offloading location. In this algorithm, the trajectory data of vehicles are collected as the input of our proposed algorithm, and then the segments of the road are divided by a k-means clustering-based algorithm. In our algorithm, the distance of time t_i and t_j can be expressed as d_{ij}, which is computed by the Euclidean distance as follows:

$$\text{d}_{ij} = \sqrt{\left(\text{lon}_{t_i} - \text{lon}_{t_j}\right)^2 + \left(la_{t_i} - la_{t_j}\right)^2}, \tag{1}$$

where lon_{t_i} denotes the longitude of the vehicles in time episode t_i and the latitude of the vehicle in time episode t_i can be denoted as la_{t_i}.

The mobility of vehicles may produce the dynamic of service requesting delay. During the services offloading, the main purpose is deciding the strategy of offloading and optimizing the service requesting delays, but the location decision is also a nonignore problem, which may affect the time delay. Thus, we put forward an algorithm for dividing road segments based on the k-means clustering algorithm [42]. Details of this algorithm can be found as Algorithm 1 shows.

With the road segments divided, we select an RSU in each segment to equip with the edge server, which mainly undertakes the task of receiving the data transmitted from vehicles and executing the computing-intensive services requested by the vehicles running in the area of this road segment. For each segment, a vehicle is randomly selected as the observed object. When the vehicle moves in the segment and the corresponding edge server is selected as the offloading location. With the running of the vehicle, the vehicle may reach the boundary of the segment; the edge server should not be switched to reduce the time consumption produced by the network handoff. In each road segment, the vehicle is observed, and the environment states are transferred to the collaborative offloading model, which is deployed on the cloud server. After the iteration computing, the offloading actions are obtained and sent to the vehicle for performing. If the vehicle moves out of the coverage of the current segment, the edge server must be switched and another RSU will be selected as the offloading location, and then the training steps are adopted as previous actions.

Table 1: List of important notations.

Notations	Description
H	Channel power gain
P	Transmission power
σ^2	Noise power
W	Transmission bandwidth
C_i	Computation capacity for executing service s_i
f_{dev}	Computation capacity of devices
f_{edge}	Computation capacity of edge servers
D_i^{off}	Data transmission size from devices to edge servers
C_i^{off}	Computation capacity for executing the offloaded service
C_i^{local}	Computation capacity for executing the service deployed on the devices
D_j^{re}	Data communication size between interacting services

3.3. System Model. In this section, the system model, contained communication model, and computation model are presented to formulate the problem of our approach.

Compared with previous studies, our approach offloads the interacting services to optimize the service requesting delay and data communication delay. In our previous works [40, 41], a parallel frequent pattern-based algorithm is presented to discover the interacting services. In this algorithm, the service requesting log denoted as $EL \triangleq \{\text{cid}, Ts, \prod\}$ is adapted as the input of the interacting services discovery algorithm, where cid denotes the nonempty set of service requesting case ID and the Ts denotes the finite serial of time stamps. The $\prod$ is the finite serial of services. We noticed that a composition service is composed of a series subservices to complete a certain complex business goal, which can be denoted as $S_i = \{s_1, s_2, \ldots, s_n\}$. In these subservices, it may exist the interrelationships between subservices. In our algorithm, the fine-grained frequent 2-itemsets of services (also called interacting services pairs) are discovered by a parallel frequent pattern-based algorithm directly, which can reveal the interrelationships between them. The details can be found in our previous works [40, 41]. Finally, the interacting services are offloaded by our CSO-DRL algorithm, which can be found in Section 4.

3.3.1. Communication Model. In order to make this paper clear, we give the important notations in our paper first, which can be found in Table 1.

Since the signal will be affected Gaussian white noise during the transmission process, we construct the communication model with Gaussian white noise to reflect the real communication channel. Here, we assume the transmission power is fixed, which can be denoted as P. The standard path loss propagation index can be denoted as θ. In IoV, the vehicles move along the road and access the service from the nearest edge server at RSU; thus, the distance between the vehicle and edge server in a time slot t can be denoted as d_t. The signal-to-noise ration (SNR) of the communication channel can be expressed as follows:

$$\text{SNR} = \frac{\text{PHd}_t^{-\theta}}{\sigma^2}, \tag{2}$$

where σ^2 is the power of Gaussian white noise and H denotes the channel gain. So the transmission rate can be given as follows:

$$R = W \log(1 + \text{SNR}), \tag{3}$$

where W denotes the channel bandwidth.

3.3.2. Computation Model. In this paper, we considered the interrelationships between interacting services. If some services are offloaded to remote cloud servers, which may produce lots of data communications between them. In the VEC system, due to the computation limitation of vehicular devices, the vehicular devices offload the portion services to RSUs equipped with edge servers, while the remaining services are locally executed at devices. The RSUs will receive the preprocessed data and execute the computing-intensive services, and then the decision will be sent back to vehicular users. Hence, the partial service offloading manner is adapted in each time slot T. The offloading manner contains three parts, which are locally computing, partial offloading on edge servers, and fully edge computing. Next, we will give the service delay model for these three parts:

(A) Locally computing. In our approach, the services without high computation capacity can be computed on the devices directly. So the computation delay can be given as follows:

$$T_i^{\text{local}} = \frac{C_i}{f_{\text{dev}}}, \tag{4}$$

where f_{dev} denotes the computation capacity of the vehicular device and the computation capacity for executing the service s_i can be denoted as C_i.

(B) Partial offloading. Due to the computation capacity limitation of the vehicular devices, the services are offloaded by a partial offloading strategy. In these composition services, some subservices are executed on vehicular devices, but some computation-intensive subservices should be offloaded on RSUs equipped with the edge servers; therefore, the data produced by these services should be transmitted to the edge servers. The time delay can be expressed as follows:

$$T_i^{par} = \max\left(T_i^{local}, 2T_i^{tran} + T_i^{edge}\right), \tag{5}$$

where T_i^{tran} denotes the transmission delay for data transmitting and T_i^{edge} denotes the computation delay for the edge computing part. These can be found as follows:

$$T_i^{tran} = \frac{D_i^{off}}{R}, \qquad T_i^{edge} = \frac{C_i^{off}}{f_{edge}}, \tag{6}$$

where D_i^{off} denotes the transmission data to edge, C_i^{off} denotes the computation capacity for executing the offloaded services, and f_{edge} denotes the computation capacity of the edge server. Thus, the delay for partial offloading can be expressed as follows:

$$T_i^{par} = \max\left(\frac{C_i^{local}}{f_{dev}}, 2\frac{D_i^{off}}{R} + \frac{C_i^{off}}{f_{edge}}\right), \tag{8}$$

where C_i^{local} denotes the computation capacity for executing the services deployed on devices.

(C) Fully offloading. For the computation-intensive services, all of these should be offloaded on edge servers; thus, the delay can be expressed as follows:

$$T_i^{full} = 2\frac{D_i}{R} + \frac{C_i}{f_{edge}}, \tag{9}$$

where D_i is the data for executing these services and the C_i denotes the computation capacity for executing the service.

3.4. Problem Formulation. Based on the description of the system problem in the above section, the delay can be expressed as follows:

$$T_i = \begin{cases} \dfrac{C_i}{f_{dev}}, \\ \max\left(\dfrac{C_i^{local}}{f_{dev}}, 2\dfrac{D_i^{off}}{R} + \dfrac{C_i^{off}}{f_{edge}}\right), \\ 2\dfrac{D_i}{R} + \dfrac{C_i}{f_{edge}}. \end{cases} \tag{10}$$

Besides the system computation model produced in the above section, we noticed that the data communication delay between interacting services is the nonignore problem. In this system, we assume the number of interacting service pairs is n, and the number of interacting service pairs executed on vehicular devices is a; thus, the latency of the system can be described as the following equation:

$$T_i = \begin{cases} \dfrac{C_i}{f_{dev}}, \\ \max\left(\dfrac{C_i^{local}}{f_{dev}}, 2\dfrac{D_i^{off}}{R} + \sum_{j=1}^{n-a}\dfrac{D_j^{re}}{R} + \dfrac{C_i^{off}}{f_{edge}}\right), \dfrac{D_j^{re}}{R} + \dfrac{C_i}{f_{edge}}. \\ 2\dfrac{D_i}{R} + \sum_{j=1}^{n} \end{cases} \tag{11}$$

where D_j^{re} denotes the data communication size between the interacting services.

In our approach, we let $\lambda \in [0, 1]$ denotes the ratio of service offloading. So $D_i^{off} = \lambda * D_i$. Finally, the latency of our system can be given as follows:

$$T_i^{sum} = \max\left((1-\lambda)\frac{C_i^{local}}{f_{dev}}, \lambda\left(2\frac{D_i}{R} + \frac{C_i^{off}}{f_{edge}}\right) + \sum_{j=1}^{n-a}\frac{D_j^{re}}{R}\right). \tag{12}$$

In this paper, the purpose is to optimize the latency, which combines the service requesting delay and data communication delay between interacting services. Thus, the optimization function can be given as follows:

$$\mathrm{Min}\left(T_i^{sum}\right). \tag{13}$$

Based on the analysis of the environments, the constraint conditions can be expressed as follows:

$$C_i^{local} \le C_i \le f_{dev}, \qquad C_i^{off} \le C_i \le f_{edge}. \tag{14}$$

4. Algorithm for DRL-Based Collaborative Service Offloading

In this section, an algorithm of collaborative service offloading based on DDPG is presented to offload the interacting services. The details of this algorithm can be described in the next contents.

Deep reinforcement learning is the algorithm mixed with deep learning and reinforcement learning. The reinforcement learning model is an iteration process, which can be divided into environment and agent. For each step, the agent observes the state s_t and takes actions according to the current policy π. When the state of the environment changes to s_{t+1}, the reward value r_t is received in the next step. The process for state transition of the environment and the reward computing can be constructed as a Markov process; thus, the probability and reward of the state transition only depend on the environment state s_t and action a_t. The agent receives these quantities according to the decision policy and interacts with the environment, and then the state is back propagated for maximizing the expected reward r_t.

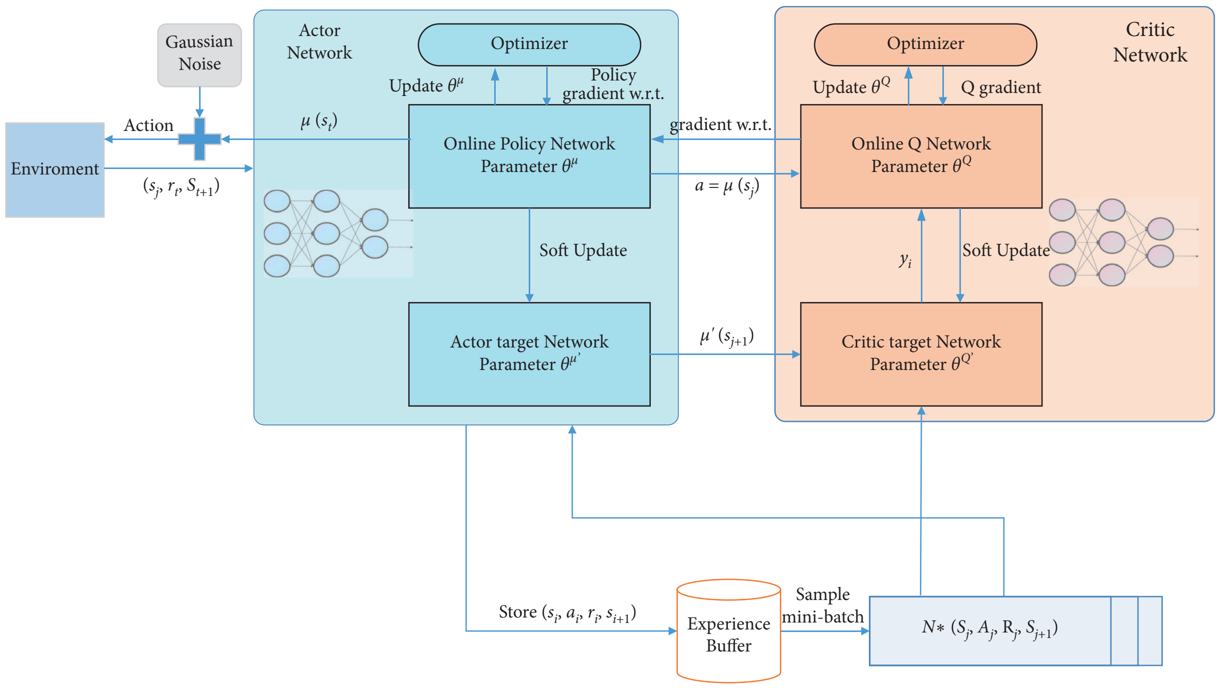


Figure 3: Deep deterministic policy gradient (DDPG) model.

The DDPG algorithm is a model-free nonpolicy action-critic algorithm based on DNN ideas, which can learn policies in the continuous action spaces [43]. The action-critic algorithm consists of the policy function and the q-value function, where the policy function undertakes the task to generate actions like an actor and the q-value function undertakes the task to evaluate the actor's performance and direct the actor's following actions.

As Figure 3 shows, the structure of DDPG is similar to actor-critic algorithm [44]. DDPG can be divided into actor network and critic network, which continues the idea of a fixed target network such as DQN [45]. First, the actor network generates $\mu(s_i)$ after the training step, and then the action space with Gaussian noise is constructed. After performing the a_t in the environment, the agent observes the next state and the immediate reward. Second, the transition is stored in the experience replay butter, and N transitions are selected by the model to make up a minibatch, which can be transferred into the actor network and critic network. With the minibatch, the actor target network outputs the action to the critic target network, and then the critic network calculates the target value. Furthermore, the critic network will be updated, and the actor network gives the minibatch action to the critic network to achieve the action's gradient. Finally, the DDPG agent updates the critic target network and the actor target network. The details can be found in [43].

Compared with the traditional DDPG algorithm, we adapt a state normalization algorithm to preprocess the observed states and add the behavior noise to the environments, which takes the difference between the maximum and minimum of each variable as the scaling factor [46]. The state normalization algorithm can solve the problem of the magnitude difference of input variables. According to the previous work [46], we can find that after normalizing the state and adding the behavior noise, our algorithm can obtain a better performance than the traditional algorithm.

4.1. State Space. In our approach, the users' requirements, network conditions, and channel and resource conditions are used as the state of the system, which will produce the state space explosion problems. In order to simplify the system, we only consider the offloading requirement in a single user scenario, which can reduce the dimension of the state space increasingly and make the algorithm easy to converge. So the state space can be expressed as follows:

$$S_t = \left(q(t), p(t), D_{\text{remain}}(t), D(t), D_{re}(t)\right), \quad (15)$$

where $q(t)$ represents the location information of the edge server and $p(t)$ denotes the location information of the vehicle. The data size of the remaining subservices is D_{remain}, and $D(t)$ denotes the task size requested by the vehicle; finally, $D_{re}(t)$ denotes the data communication size between the interacting services.

4.2. Action Space. We define the action space as the size of the offloading task, which can be expressed as follows:

$$a_t = (k(t), \lambda(k)), \quad (16)$$

where $k(t)$ indicates the edge server that undertakes the task for executing the services and $\lambda(k)$ denotes the service offloading ratio.

4.3. Reward Function. In this paper, the main purpose of our algorithm is to minimize the latency, which contains the service requesting delay and communication delay. The r_t is

Input. Training data set I; critic learning rate λ_c; actor learning rate λ_a; discount factor γ; soft update factor τ; experience replay buffer B_m; minibatch size N; Gaussian noise n_i, current state $q_i, p_i, D_{\text{remain}}(i), D(i), f_i$; state parameters with normalized $\gamma_{D_{\text{remain}}}, \gamma_{D_{ue}}, \gamma_x, \gamma_y$
Output. Reward r_i
Initialize the critic network and the actor network with weights θ^Q and θ^μ
Initialize the target network with weights $\theta^{Q'} \longleftarrow \theta^Q$, $\theta^{\mu'} \longleftarrow \theta^\mu$
Empty the experience replay buffer B_m
for episode = 1 to M **do**
  Reset the simulation parameters and receive the initial observation state s_1
  for $t = 1, 2, \ldots, T$ **do**
    Initialize s_t as the current state
    Normalize state $s_t' \longleftarrow s_t$, and obtain the feature vector $\phi(s')$
    Get the action $a_t = \pi\theta^\mu(\phi(s')) + n_t$
    Execute the action to get the reward re_t, and observe the next state s_{t+1}
    Normalize state $s_{t+1}{}' \longleftarrow s_{t+1}$
    if B_m is not full **then**
      Store $(s_t', a_t, r_t, s_{t+1}{}')$ to B_m
    else
      Randomly replace a transition in B_m with $(s_t', a_t, r_t, s_{t+1}{}')$
      Randomly sample a mini-batch from B_m
      $y_j = r_j + \gamma Q'(s_{j+1}{}', \mu'(s_{j+1}{}'|\theta^{\mu'}), \theta^{Q'})$
      $L(\theta^Q) = 1/N \sum_{j=1}^{N} (y_j - Q(s_j', (a_j|\theta^Q))^2)$
      Update θ^μ by the sampled policy gradient
      Soft update the critic target network and actor target network
    end if
  end for
end for
Get $\mu(s'|\theta^\mu)$
for episode = 1, 2, ..., E **do**
  Reset the VEC parameters and receive the initial observation s_1
  for $t = 1, 2, \ldots, T$ **do**
    Normalize state $s_t' \longleftarrow s_t$
    $a_t = \mu(s'|\theta^\mu)$
    Execute the action a_t and get the reward r_t
  end for
end for

ALGORITHM 2: Algorithm for collaborative service offloading with DDPG.

defined as the reward when the action a_t is performed at state s_t. In order to reduce the system latency while guaranteeing the sufficient computation resources of devices, the reward function can be given as follows:

$$r_t = -T_i^{\text{sum}}. \tag{17}$$

The action value function is used to describe the expected return of policy π at time episode t in the reinforcement learning algorithm, which can be expressed as follows:

$$Q^\pi(s_t, a_t) = E_{r_t}, s_{t+1} \sim E\left[r_t(s_t, a_t) + \gamma E_{a_{t+1}} \sim \pi\left[Q^\pi(s_{t+1}, a_{t+1})\right]\right]. \tag{18}$$

Therefore, if the update of the target policy is continuous, the target policy can be described as a function μ: $A \leftarrow S$, which can be expressed as follows:

$$Q^\pi(s_t, a_t) = E_{r_t}, s_{t+1} \sim E\left[r_t(s_t, a_t) + \gamma E_{a_{t+1}} \sim \pi\left[Q^\pi(s_{t+1}, a_{t+1})\right]\right]. \tag{19}$$

Algorithm 2 demonstrates the details of our CSO-DRL algorithm, which is an iteration process. During the training process, the parameters of the critic network and the actor network are trained with the iteratively update process, and then the trained actor network parameters are adopted to offload the interacting services.

5. Experimental Evaluation

In this section, we conduct experiments to evaluate the performance of our algorithm. First, all the simulation parameters are indicated, and then the efficiency of our algorithm is conducted to compare with other baseline algorithms in the same simulation environment.

5.1. Experiment Setup. To ensure the practicability of the algorithm, we use a real-world data set of taxi trajectories in Shanghai (https://cse.hkust.edu.hk/scrg/) to evaluate the experiment in this paper. This data set records the GPS trajectory data on February 20, 2007, in Shanghai, which contains the longitude and latitude of 4316 taxies. Here, we

Table 2: Simulation parameters.

Parameters	Numerical value	Unit
Channel power gain, H	−50	dB
Transmission bandwidth, W	100	MHz
Noise power, σ^2	−100	dBm
Devices computation capacity, f_{dev}	0.8	GHz
Edge computation capacity, f_{edge}	3	GHz
Task size with random values, D_i	< 60	Mbit
Data communication size between interacting services, D_j^{re}	$< M_i$	Mbit

conduct the experiment in the real world to obtain the road segments and determine the location of service offloading. Furthermore, the simulation experiments are conducted to compare the latency with other baseline algorithms, which are described as follows:

(i) Local-only: executing all services locally

(ii) Offload-only: offloading all services to RSUs deployed with edge servers

(iii) AC: actor-critic-based service offloading algorithm [44]

(iv) DQN: DQN-based service offloading algorithm [45]

We set the channel power gain $H = -50$ dB when the distance is 1 m, and the transmission bandwidth is $W = 100$ MHz. The noise power accepted without signal blocking is $\sigma^2 = -100$ dBm. The computation capacity of the vehicular devices and edge servers are set to $f_{\text{dev}} = -0.8$ GHz and $f_{\text{edge}} = 3$ GHz, respectively. We set the distance between the vehicular devices and the edge servers d_t randomly. The size of the task is randomly assigned, and $f_i = \epsilon D_i$ is the required computation resource, where ϵ is set at 0.5 Gcyles/MB. The detailed simulation parameters are listed in Table 2.

Since the data set records the location information of the vehicles every 60 s, so the time slot is set $t = 60$ s, and the total time is 24 hours. For fairness, we conduct all the algorithms in the same simulation environment. During the time slots, all the services are offloaded continually by our CSO-DRL algorithm, and then the delay is conducted to evaluate the efficiency of our algorithm.

5.2. Experimental Results. The hyperparameters in our algorithm affect the overall performance of the algorithm seriously. The experiments are conducted to determine the optimal values of these hyperparameters in our algorithm. First, we conduct experiments to select the optimal hyperparameters. Figure 4 shows the convergence of the CSO-DRL algorithm under different learning rates. In the DDPG model, there are two training networks, which are the actor network and the critic network. We set the learning rates of the critic network and actor network separately. According to the training, we find that the algorithm cannot converge when the learning rate is $\lambda_a = 0.1$ and $\lambda_c = 0.2$. When the learning rate is $\lambda_a = 0.001$ and $\lambda_c = 0.002$, the algorithm can get a better convergence value. Therefore, the optimal learning rate of actor network $\lambda_a = 0.01$, and the learning rate of critic network $\lambda_c = 0.02$.

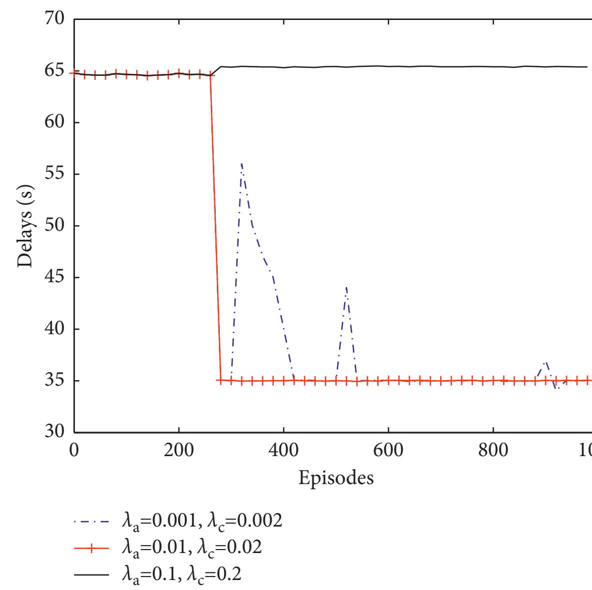


Figure 4: Convergence of CSO-DRL under different learning rates.

Next, the convergence of the algorithm under different discount factors γ are conducted, which can indicate the weight of the value for the next action. The value of γ can affect the efficiency of the algorithm. As Figure 5 shows, the trained service offloading strategy performs best when the discount factor $\gamma = 0.5$. Therefore, we set the value $\gamma = 0.5$ in the following experiments.

Furthermore, the convergence of the algorithm under different exploration parameters σ is also conducted. Noticed that, the larger value for σ allows the agent to choose a prudent strategy, and make the evaluation of the q-value become more accurate, and then the training process will become more stable. As Figure 6, when our algorithm converges at $\sigma = 0.1$, the delay fluctuates at 35*s*. The initial delay with the $\sigma = 0.1$ is lower than the situation with $\sigma = 0.01$. We noticed that when $\sigma = 1$, the algorithm decreases at 280 episodes, but the convergence is unstable. Hence, we set $\sigma = 0.1$ to obtain better performance in the following experiments.

With the parameters determined, we compare the delay of our algorithm with baseline algorithms. Figure 7 shows the performance of different algorithms. From this figure, it can be observed that the delay of the CSO-DRL algorithm is lower than the other baseline algorithms significantly under the same service size. As the number of iterations increases,

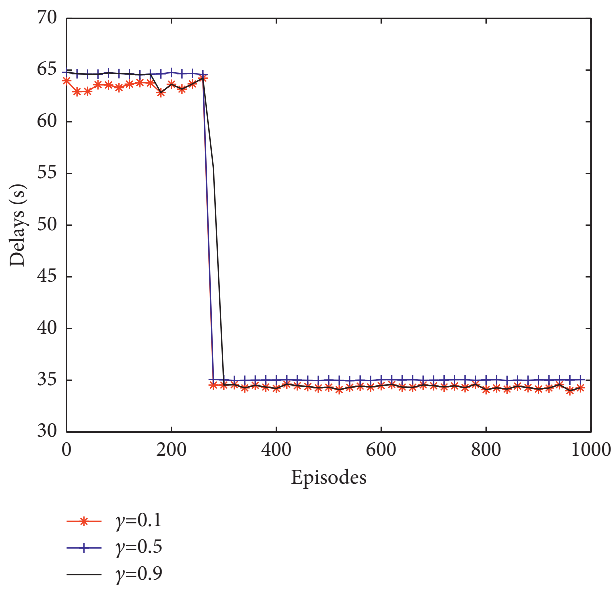


Figure 5: Convergence of CSO-DRL under different discount factors.

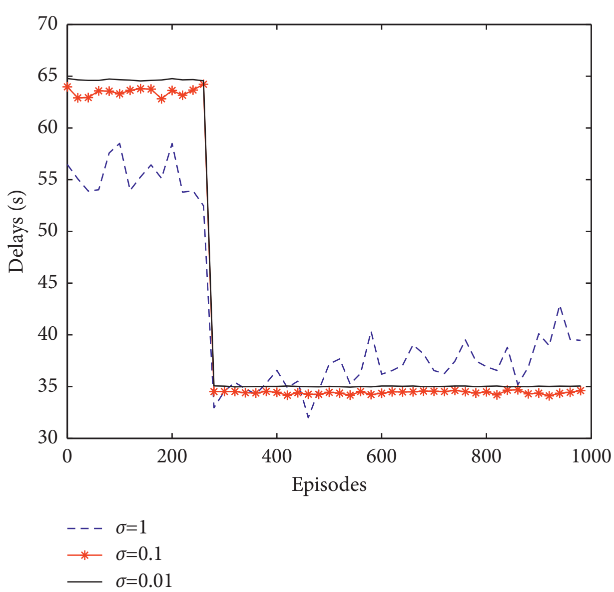


Figure 6: Convergence of CSO-DRL under different exploration parameters.

the AC algorithm cannot converge, while the DQN algorithm and CSO-DRL algorithm can achieve convergence. Because the actor network and critic network of the AC algorithm are updated at the same time, the action selection of the actor network depends on the value function of the critic network, but the critic network is difficult to converge. In contrast, since both DQN and DDPG have a dual network structure. While DQN is only suitable for discrete action spaces and the state space for service offloading is continuous, DQN cannot search the optimal offloading strategy accurately. On the other hand, the CSO-DRL algorithm explores a continuous action space and takes precise actions, which may result in an optimal policy and make the delay

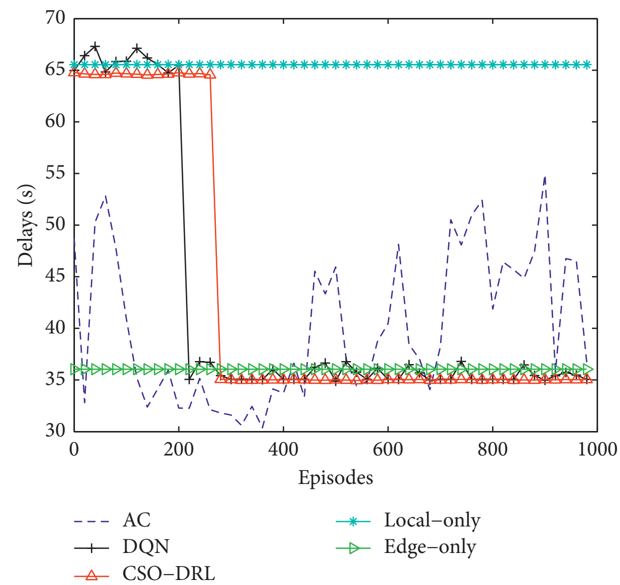


Figure 7: Convergence performance of different algorithms.

reduce significantly. We can also find that the CSO-DRL algorithm and the DQN algorithm can converge at 300 episodes, and the CSO-DRL algorithm obtains a better delay than DQN and converges at 35, which is more stable than the DQN algorithm.

Besides the comparison with the AC algorithm and the DQN algorithm, we also conduct the experiments by comparing the CSO-DRL algorithm with the local-only algorithm and the offload-only algorithm. In Figure 7, it is obvious that the delay of local computing is too large due to the limitation of the computation capacity of the vehicular devices. Since the offload-only algorithm depends on the running state and the computation resources of the edge server, and the vehicles keep the state with resource competition for a long time, the CSO-DRL algorithm can obtain a better delay than offload-only algorithm.

We also conduct the experiment under different service sizes (also called task sizes). Since the AC algorithm cannot converge, we compare the delay with other algorithms. As Figure 8 shows, with the task sizes increased, the delay of the CSO-DRL algorithm is always lower than the other baseline algorithm for the same task size. We notice that the delay of CSO-DRL increases much slower than other baseline algorithms, which is significant when the task sizes increase doubled.

In order to explore the effect of the number of edge servers for the delay, we compare the delay of our proposed algorithm with other algorithms under different numbers of edge servers. As Figure 9 shows, the delay of all algorithms except DQN is almost constant as the number of edge servers increases. With the increase in the number of edge servers, the DQN algorithm fluctuates at about 40 s. Besides, the proposed CSO-DRL algorithm achieves the lowest delay. The reason is DDPG model can find the optimal value in the continuous action and obtain the optimal control policy.

Besides the delay comparison with other algorithms under different numbers of edge servers, we compare these

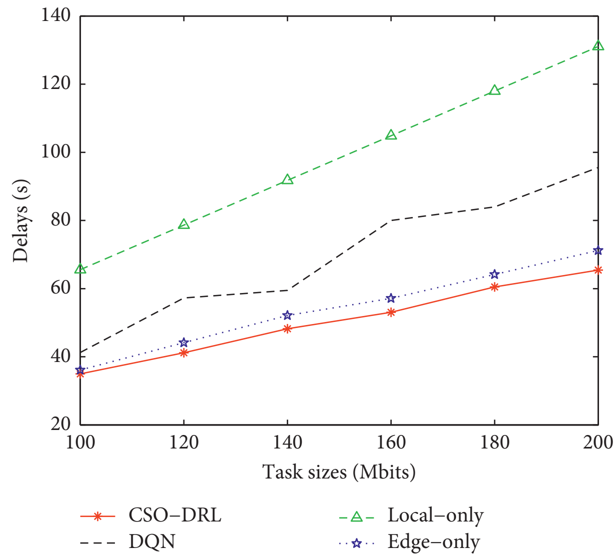


Figure 8: Delay comparison under different task sizes.

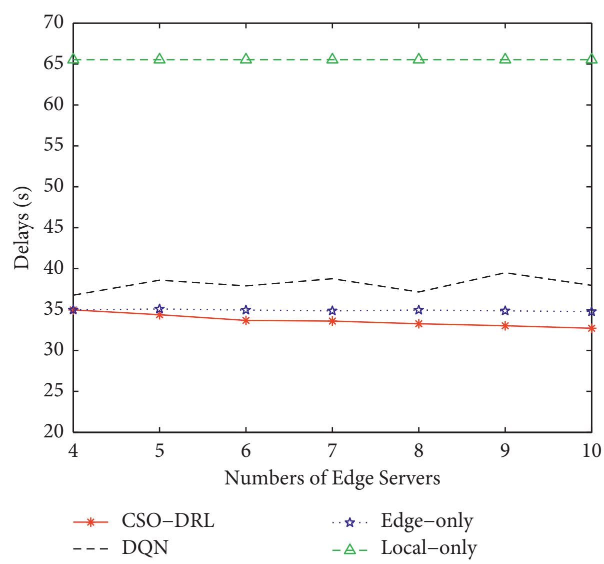


Figure 9: Delay comparison under different numbers of edge servers.

algorithms under different computing capabilities of edge servers and computing capabilities of devices, which can be found in Figures 10 and 11, respectively. Due to the non-convergence of the AC algorithm, we only compared our proposed CSO-DRL algorithm with the DQN algorithm.

Figure 10 shows the same group of experiments in terms of convergence performance and delay under different computing capabilities of edge servers. We vary the computation capabilities of edge servers from 2 GHz to 4 GHz. Figure 10(a) shows the convergence performance of the DQN algorithm and our CSO-DRL algorithm. We find that when the computation capability of the edge server is 3 GHz, the delay of two algorithms is higher than that when the computation capability of the edge server is 4 GHz. Thus, the smaller the computation capability of the edge server, the slower the service execution speed of the system at the same time, which results in a larger delay of the algorithm. We also notice that the CSO-DRL algorithm obtains a lower delay than DQN and converges at 35 s, which is stable than DQN algorithm. Figure 10(b) shows the delay comparison between the DQN algorithm and the CSO-DRL algorithm under different CPU frequency of edge servers. It is obvious that the CSO-DRL algorithm achieves a lower delay than DQN. As the CPU frequencies of edge servers increase from 2 GHz to 4 GHz, the delay decreases following and remains at 27 s when the CPU frequency of edge server is 4 GHz.

Figure 11 shows the same group of experiments in terms of convergence performance and delay under different computation capabilities of edge servers. We vary the computation capabilities of edge servers from 0.6 GHz to 1 GHz. Figure 11(a) shows the convergence performance of the DQN algorithm and our CSO-DRL algorithm. We find

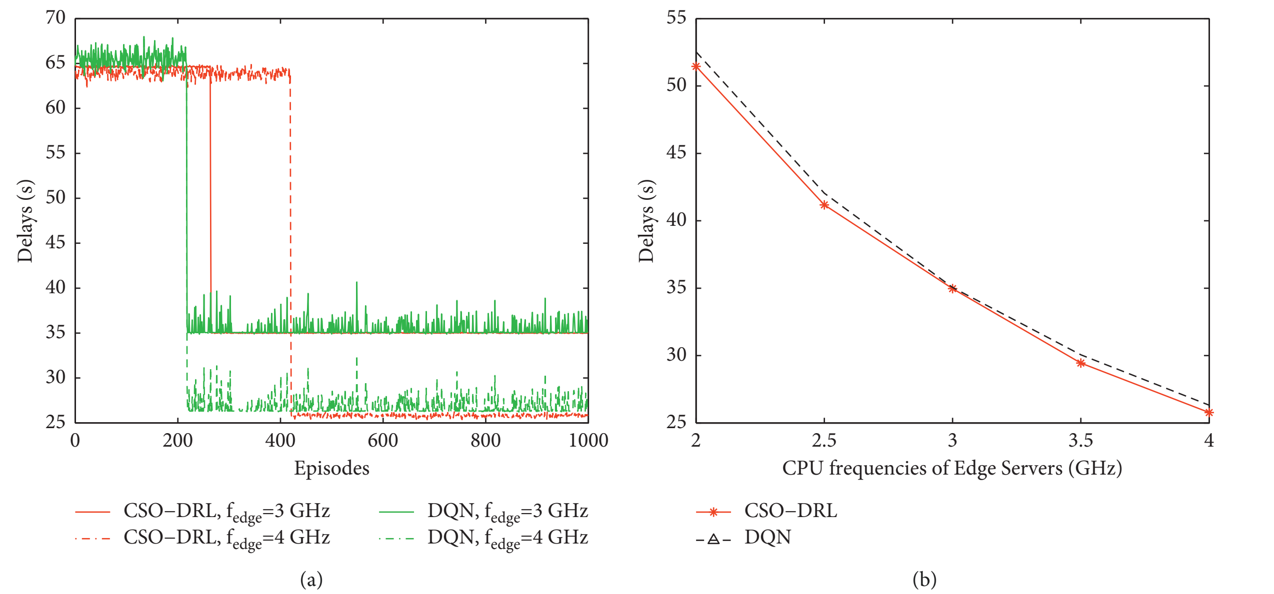


Figure 10: (a) Convergence of different algorithms under different computation capabilities of the edge server. (b) Comparison between different algorithms.

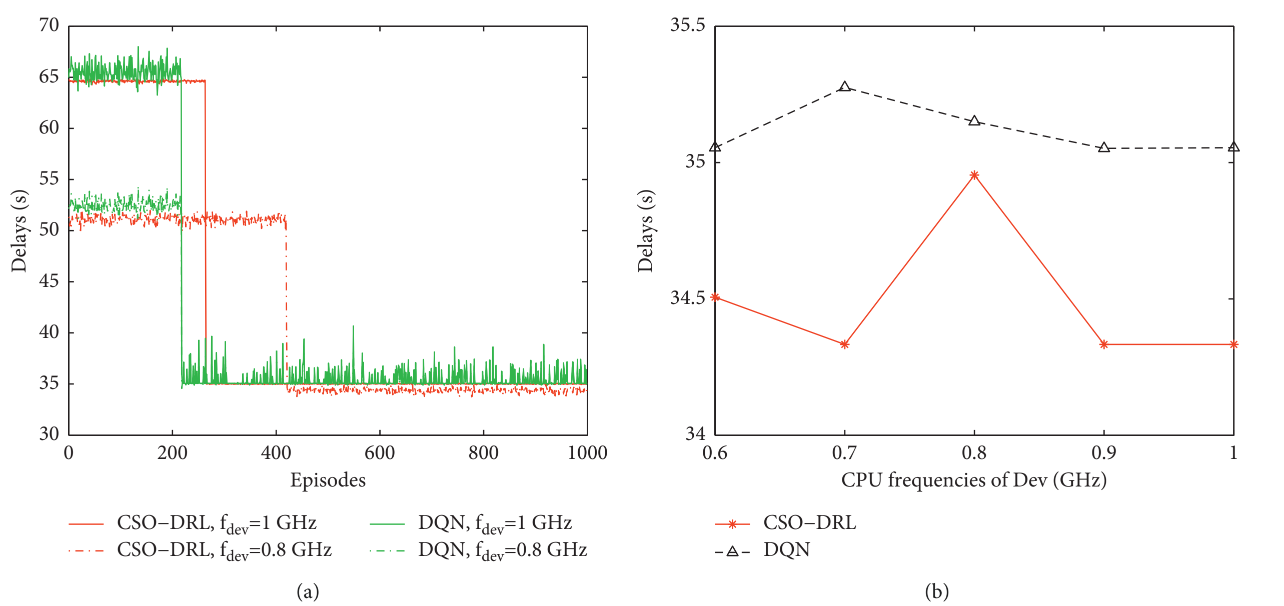


Figure 11: (a) Convergence of different algorithms under different computation capabilities of devices. (b) Comparison between different algorithms.

that when the computation capability of devices is 0.8 GHz, the delay of two algorithms is higher than that when the computation capability of devices is 1 GHz. Thus, the smaller the computation capability of devices, the slower the service execution speed of the system at the same time, which results in a larger delay of the algorithm. We also notice that the CSO-DRL algorithm obtains a better delay than DQN and converges at 35 s, which is stable than the DQN algorithm. Figure 11(b) shows the delay comparison between the DQN algorithm and the CSO-DRL algorithm under different CPU frequencies of devices. The delay of CSO-DRL algorithm increases slowly from 34.5 s to 35 s as the CPU frequency of devices increases from 0.7 GHz to 0.8 GHz and remains at about 34.3*s* when the CPU frequency is 1 GHz. It is obvious that the CSO-DRL algorithm achieves a lower delay than DQN under the same CPU frequency of devices.

6. Conclusion

In this paper, we propose a collaborative service offloading approach with deep reinforcement learning in vehicular edge computing named CSO-DRL. Our approach first divides the road segments by k-means-based algorithm through analyzing the trajectory data of vehicles, and then the offloading location is determined by observing the vehicle running status. Secondly, the interacting services are discovered by a parallel frequent pattern-based algorithm efficiency. Furthermore, the collaborative service offloading algorithm is presented by the DDPG model for offloading the interacting services, which can reduce the service requesting delay and communication delay together. Finally, the efficiency of the algorithms is evaluated by real-world data-based simulation experimental evaluations. The results show our algorithm can get a better delay in obtaining the optimal service offloading strategy than other baseline algorithms.

Although our approach considers the mobility of vehicles in service offloading and reveals the impact of interrelationship between services on service offloading. There are some avenues for our future studies. In reality, the vehicles are in a complex scenario with resource competition. Thus, how to design offloading strategies for multiusers VEC by considering the computation resource competition is a nonignored problem. We also noticed that due to the limitation of device power, we will present the service offloading strategy to investigate the multiobjective optimization of reducing the energy consumption of vehicular devices and latency in VEC.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Young Project of Science and Technology Research Program of Chongqing Education Commission of China (Nos. KJQN201900708 and KJQN202100738) and the National Natural Science Foundation of China (No. 62101080).

References

[1] J. Contreras-Castillo, S. Zeadally, and J. A. Guerrero-Ibanez, “Internet of vehicles. Architecture, protocols, and security,” *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3701–3709, 2018.

[2] X. Wang, Z. Ning, X. Hu et al., “Optimizing content dissemination for real-time traffic management in large-scale internet of vehicle systems,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1093–1105, 2019.

[3] D. Singh and M. Singh, “Internet of vehicles for smart and safe driving,” in *Proceedings of the International Conference on Connected Vehicles and Expo, ICCVE*, pp. 328-329, IEEE, Shenzhen, China, October, 2015.

[4] R. Hussain, D. Kim, J. Son et al., “Secure and privacy-aware incentives-based witness service in social internet of vehicles clouds,” *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2441–2448, 2018.

[5] L. Wu, R. Zhang, Q. Li, C. Ma, and X. Shi, “A mobile edge computing-based applications execution framework for internet of vehicles,” *Frontiers of Computer Science*, vol. 16, no. 5, Article ID 165506, 2022.

[6] J. Zhang and K. B. Letaief, “Mobile edge intelligence and computing for the internet of vehicles,” *Proceedings of the IEEE*, vol. 108, no. 2, pp. 246–261, 2020.

[7] Y. Chen, H. Xing, and M Zhuo, “Cost-efficient edge caching for noma-enabled iot services,” *China Communications*, 2022.

[8] Y. Zhang, *Mobile Edge Computing*, Vol. 9, Springer, , Berlin/Heidelberg, Germany, 2022.

[9] Z. Ning, J. Huang, X. Wang, J. J. P. C. Rodrigues, and L. Guo, “Mobile edge computing-enabled internet of vehicles. Toward energy-efficient scheduling,” *IEEE Network*, vol. 33, no. 5, pp. 198–205, 2019.

[10] J. Wang, H. Ke, X. Liu, and H. Wang, “Optimization for computational offloading in multi-access edge computing. A deep reinforcement learning scheme,” *Computer Networks*, vol. 204, Article ID 108690, 2022.

[11] F. Song, H. Xing, X. Wang, S. Luo, P. Dai, and K. Li, “Offloading dependent tasks in multi-access edge computing. A multi-objective reinforcement learning approach,” *Future Generation Computer Systems*, vol. 128, pp. 333–348, 2022.

[12] Y. Chen, F. Zhao, Y. Lu, and X. Chen, “Dynamic task offloading for mobile edge computing with hybrid energy supply,” *Tsinghua Science and Technology*, vol. 10, 2021.

[13] I. Sarkar, M. Adhikari, N. Kumar, and S. Kumar, “A collaborative computational offloading strategy for latency-sensitive applications in fog networks,” *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4565–4572, 2022.

[14] X. Gu, G. Zhang, M. Wang, W. Duan, M. Wen, and P. H. Ho, “Uav-aided energy-efficient edge computing networks. Security offloading optimization,” *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4245–4258, 2022.

[15] J. Huang, Z. Tong, and Z. Feng, “Geographical poi recommendation for internet of things. A federated learning approach using matrix factorization,” *International Journal of Communication Systems*, vol. n/a, Article ID e5161, 2022.

[16] Q. Wu, H. Liu, R. Wang, P. Fan, Q. Fan, and Z. Li, “Delay-sensitive task offloading in the 802.11p-based vehicular fog computing systems,” *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 773–785, 2020.

[17] Y. Chen, F. Zhao, X. Chen, and Y. Wu, “Efficient multi-vehicle task offloading for mobile edge computing in 6g networks,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 4584–4595, 2022.

[18] Z. Tong, X. Deng, J. Mei, B. Liu, and K. Li, “Response time and energy consumption co-offloading with SLRTA algorithm in cloud-edge collaborative computing,” *Future Generation Computer Systems*, vol. 129, no. 64–76, pp. 64–76, 2022.

[19] A. A. Shah, N. A. Bhatti, K. Dev, and B. S. Chowdhry, “MUHAFIZ. iot-based track recording vehicle for the damage analysis of the railway track,” *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 9397–9406, 2021.

[20] H. Attaullah, T. Kanwal, A. Anjum et al., “Fuzzy-logic-based privacy-aware dynamic release of iot-enabled healthcare data,” *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4411–4420, 2022.

[21] J. C.-W. Lin, G. Srivastava, Y. Zhang, Y. Djenouri, and M. Aloqaily, “Privacy-preserving multiobjective sanitization model in 6g iot environments,” *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5340–5349, 2021.

[22] R. Roman, J. López, and M. Mambo, "Mobile edge computing, fog et al. A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, pp. 680–698, 2018.

[23] W. Chen and B. Yang, "Energy efficiency analysis of e-commerce customer management system based on mobile edge computing," *Scientific Programming*, vol. 2022, Article ID 5333346, 9 pages, 2022.

[24] S. Wang, Y. Li, and S. Pang, "A task scheduling strategy in edge-cloud collaborative scenario based on deadline," *Scientific Programming*, vol. 2020, Article ID 3967847, 9 pages, 2020.

[25] Y. Zhou, L. He, B. Wang, Y Su, and H. Chen, "MCAF. developing an annotation-based offloading framework for mobile cloud computing," *Scientific Programming*, vol. 2020, no. 9, Article ID 5304612, 9 pages, 2020.

[26] K. Gasmi, S. Dilek, S. Tosun, and S. Ozdemir, "A survey on computation offloading and service placement in fog computing-based IoT," *The Journal of Supercomputing*, vol. 78, no. 2, pp. 1983–2014, 2022.

[27] Z. Hong, W. Chen, H. Huang, S. Guo, and Z. Zheng, "Multi-hop cooperative computation offloading for industrial iot-edge-cloud computing environments," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 12, pp. 2759–2774, 2019.

[28] Z. Yu, Y. Gong, S. Gong, and Y. Guo, "Joint task offloading and resource allocation in uav-enabled mobile edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3147–3159, 2020.

[29] F. Shan, J. Luo, J. Jin, and W. Wu, "Offloading delay constrained transparent computing tasks with energy-efficient transmission power scheduling in wireless iot environment," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4411–4422, 2019.

[30] Y. Chen, W. Gu, and K. Li, "Dynamic task offloading for internet of things in mobile edge computing via deep reinforcement learning," *International Journal of Communication Systems*, vol. n/a, Article ID e5154, 2022.

[31] J. Chen and Z. Wu, "Dynamic computation offloading with energy harvesting devices. A graph-based deep reinforcement learning approach," *IEEE Communications Letters*, vol. 25, no. 9, pp. 2968–2972, 2021.

[32] J. Xu, D. Li, W. Gu, and Y. Chen, "Uav-assisted task offloading for iot in smart buildings and environment via deep reinforcement learning," *Building and Environment*, vol. 222, Article ID 109218, 2022.

[33] J. Huang, C. Zhang, and J. Zhang, "A multi-queue approach of energy efficient task scheduling for sensor hubs," *Chinese Journal of Electronics*, vol. 29, no. 2, pp. 242–247, 2020.

[34] J. Huang, B. Lv, Y. Wu, Y. Chen, and X. Shen, "Dynamic admission control and resource allocation for mobile edge computing enabled small cell network," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 2, pp. 1964–1973, 2022.

[35] X. Xu, R. Gu, F. Dai, L. Qi, and S. Wan, "Multi-objective computation offloading for internet of vehicles in cloud-edge computing," *Wireless Networks*, vol. 26, no. 3, pp. 1611–1629, 2020.

[36] Q. Luo, C. Li, T. H. Luan, W. Shi, and W. Wu, "Self-learning based computation offloading for internet of vehicles. Model and algorithm," *IEEE Transactions on Wireless Communications*, vol. 20, no. 9, pp. 5913–5925, 2021.

[37] X. Wang, Z. Ning, and L. Wang, "Offloading in internet of vehicles. A fog-enabled real-time traffic management system," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4568–4578, 2018.

[38] X. Xu, Y. Xue, L. Qi et al., "An edge computing-enabled computation offloading method with privacy preservation for internet of connected vehicles," *Future Generation Computer Systems*, vol. 96, no. 89–100, pp. 89–100, 2019.

[39] X. He, H. Lu, M. Du, Y. Mao, and K. Wang, "QoE-based task offloading with deep reinforcement learning in edge-enabled internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2252–2261, 2021.

[40] Y. Huang, J. Huang, B Cheng, T. Yao, and J. Chen, "Poster. Interacting data-intensive services mining and placement in mobile edge clouds," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking, MobiCom*, pp. 558–560, ACM, Snowbird, UT, USA, October, 2017.

[41] Y. Huang, J. Huang, C. Liu, and C. Zhang, "PFPMine. A parallel approach for discovering interacting data entities in data-intensive cloud workflows," *Future Generation Computer Systems*, vol. 113, pp. 474–487, 2020.

[42] L. Rokach, "A survey of clustering algorithms," in *Proceedings of the Oded Maimon and Lior Rokach, editors, Data Mining and Knowledge Discovery Handbook, 2nd ed*, pp. 269–298, Springer, Berlin/Heidelberg, Germany, 2010.

[43] T. P. Lillicrap, J. J. Hunt, and P. Alexander, "Continuous control with deep reinforcement learning," in *Proceedings of the Yoshua Bengio and Yann LeCun, editors, 4th International Conference on Learning Representations, ICLR 2016*, Conference Track Proceedings, San Juan, Puerto Rico, May, 2016.

[44] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation,"vol. 12, pp. 1057–1063, in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 12, pp. 1057–1063, The MIT Press, Denver, Colorado, USA, November, 1999.

[45] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[46] Y. Wang, W. Fang, Y Ding, and N. Xiong, "Computation offloading optimization for uav-assisted mobile edge computing. a deep deterministic policy gradient approach," *Wireless Networks*, vol. 27, no. 4, pp. 2991–3006, 2021.

Hindawi
Scientific Programming
Volume 2022, Article ID 7394689, 14 pages
https://doi.org/10.1155/2022/7394689

Research Article

A Game-Theoretic Scheme for Parked Vehicle-Assisted MEC Computation Offloading

Ruoyu Chen, [1,2,3] **Yanfang Fan**, [1,2] **Mengxin Jia**, [1] **and Shuang Yuan** [1]

[1] *School of Computer Science, Beijing Information Science & Technology University, Beijing, China*
[2] *Beijing Key Laboratory of Internet Culture & Digital Dissemination Research, Beijing Information Science & Technology University, Beijing, China*
[3] *Southeast Institute of Information Technology, Beijing Institute of Technology, Fujian 351100, China*

Correspondence should be addressed to Yanfang Fan; fyfhappy@bistu.edu.cn

Received 2 April 2022; Revised 14 July 2022; Accepted 20 July 2022; Published 24 August 2022

Academic Editor: Xu Zhang

By offloading computation tasks, multi-access edge computing (MEC) supports diverse services and reduces delay and energy consumption of mobile devices (MDs). However, limited resources of edge servers may be the bottleneck for task computing in high-density scenarios. To address this challenge, by leveraging the underutilized resources of parked vehicles to execute tasks, we propose a parked vehicle-assisted multi-access edge computing (PV-assisted MEC) architecture, which enables MEC servers to expand their capability flexibly. To achieve efficient offloading, we propose a PV-assisted MEC offloading scheme in a multi-MD environment. We design a game-based distributed algorithm to minimize the overhead of MDs and further reduce the burden on the MEC server. Simulation results show that compared with the common MEC system, our scheme can reduce the burden on the MEC server by 5% and the offloading overhead by 17%.

1. Introduction

With the improvement of mobile devices' capabilities and the ever-increasing interest in mobile applications, delay-sensitive and computation-intensive mobile applications have been emerging and drawing significant attentions, spanning technologies such as augmented reality, speech-to-text conversion, image processing, and interactive online games. However, due to the scarcity of resources, mobile devices are usually unable to meet the massive computing demands. The solution to this problem lies in improving the communication infrastructure by computation offloading [1–3]. Multi-access edge computing (MEC) is regarded as a key technology and architectural concept for the improvement of the computation offloading efficiency. MEC aims at extending cloud computing capabilities to the edge. Mobile devices (MDs) can offload tasks to nearby network edge servers [4]. For example, video streams and images collected through sensors or cameras mounted on the vehicles must be processed in real time to detect surrounding objects, recognize traffic lights, etc., to ensure the safety of autonomous driving. However, vehicles do not have the capacity to process large amounts of images and videos instantly, so tasks are offloaded to edge server for processing, reducing the incidence of traffic accidents. Computation offloading technology in MEC not only overcomes the shortage of computing capabilities on mobile terminals but also avoids huge latency caused by transferring tasks to the cloud [5, 6]. However, existing MEC servers tend to have lightweight computing resources due to cost constraints, which means they are still not well equipped to handle the ever-growing task demands.

Scholars have studied the problem that it is difficult for a single VEC server to meet the strict latency requirements of MDs. The authors in [5] proposed a tiered offloading framework for edge computing, which utilizes nearby backup computing servers to make up for the insufficient MEC server resources. Guo and Liu [2] proposed a cloud-MEC collaborative computation offloading scheme with centralized cloud and multi-access edge computing over Fi-Wi network architecture. In addition, idle resources in unmanned aerial vehicles (UAVs) were used as a supplement to the edge computing server to provide effective resource utilization and reliable service [7].

With the rapid development of the automotive industry, vehicles are equipped with an ever-increasing amount of communication and computing resources. Several works have focused on vehicle-assisted edge network to improve network service quality by leveraging idle resources in vehicles. In this network, idle resources are used to compute tasks to assist the edge network as vehicles with idle resources approaching vehicles carrying computation tasks. In daily life, 70% of personal vehicles are parked for an average of more than 20 hours per day [8]. These parked vehicles have a lot of idle computing, storage, and communication resources, as well as plenty of energy. Therefore, utilizing these idle resources is a promising way to improve network efficiency.

The use of parked vehicles to support network services has two advantages that cannot be ignored. On the one hand, parked vehicles are relatively stable in terms of communication. A moving vehicle may change its position frequently, which may cause the connection between the vehicle and the server to become unstable and affect the efficiency of task execution. In contrast, parked vehicles may remain stationary for long periods of time. On the other hand, parked vehicles involved in task offloading indirectly extend the service area of VEC. Outside the coverage of roadside units (RSUs), parked vehicles can serve as static nodes and service infrastructure, alleviating the shortage of edge server resources and supporting interconnection between vehicles and servers [9].

In this work, unlike existing computation offloading studies, we focus on reducing MDs' delay and energy consumption to improve quality of service (QoS). In addition, parked vehicles that can be used as service nodes in this work include not only those parked centrally in parking lots but also those parked scattered on the roadside, where legally permitted. We focus on the design of parked vehicle-assisted MEC architecture and the corresponding efficient computation offloading scheme. The main contributions of this study are as follows:

(i) A parked vehicle-assisted multi-access edge computing (PV-assisted MEC) architecture is presented, in which nearby parked vehicles can help extend the service capabilities of the MEC system.

(ii) The offloading decision problem is formulated as a noncooperation game. A game-based PV-assisted task offloading algorithm (GPTOA) is proposed, which decides whether each MD should offload and, if so, to which channel of MEC server or which PV.

(iii) Simulation results show that the GPTOA not only effectively reduces the burden on the MEC server but also achieves significant performance improvements in terms of offloading overhead.

The rest of this study is organized as follows. First, related works are discussed in Section 2. Second, the PV-assisted MEC architecture is described in Section 3. Next, Section 4 presents the system model. After that, Section 5 formulates the task offloading problem and proposes a game-based PV-assisted task offloading algorithm. Extensive simulation results are provided in Section 6, followed by conclusions in Section 7.

2. Related Work

There are a number of studies focusing on mobile applications in MEC. Most of these focused on processing data and improving service qualities [10–15]. Zhang et al. [16] considered load balancing of computation resources on the edge servers and the highly dynamical nature of the vehicular networks, which led them to introduce fiber-wireless (Fi-Wi) technology to enhance vehicle edge computing network (VECN). Then, they used a game theory-based nearest task offloading algorithm and an approximate load balancing task offloading algorithm to solve the delay minimization problem. Cheng et al. [17] proposed a method to predict Wi-Fi offload potential and access costs by jointly considering user satisfaction, offload performance, and mobile network operators' revenues. The results showed that this scheme can improve the average utility of users and reduce service latency. Chen et al. [18] showed that it is NP-hard to find centralized optimum for task offloading in MEC with the goal of minimizing the overall computation overhead. Hence, they adopt a game-theoretic approach for achieving efficient offloading in a distributed manner.

The recent advent of vehicle-to-everything (V2X) communication technology makes vehicles an important network resource for improving network performance. Ding et al. [19] used CR (cognitive radio) router-enabled vehicles to transmit data to the desired location. Feng [20] proposed the hybrid vehicle edge cloud (HVC) framework, which made it possible to share available resources with neighboring vehicles through vehicle-to-vehicle (V2V) communication. Zhang et al. [21] investigated the effectiveness of computational transport strategies for vehicle-to-infrastructure (V2I) and V2V communication modes. They proposed an efficient predictive combination-mode relegation scheme that adaptively offloaded tasks to the MEC servers via direct uploading or predictive relay transmissions. Huang et al. [22] introduced the concept of vehicle neighbor group (VNG), which made it convenient to share similar services through V2V communication. Considering the similarity of tasks and computational capability of vehicles, Qiao et al. [23] divided vehicles into task computing sub-cloudlet and task offloading sub-cloudlet. Based on the two sub-cloudlets, they proposed a collaborative task offloading scheme that can effectively reduce the number of similar tasks transferred to MEC servers.

Furthermore, certain existing works focused on exploring ways to leverage the communication, storage, and computation capacity of parked vehicles, in which vehicles became service nodes for computation offloading. Liu et al. [24] proposed a vehicle edge computing network architecture in which vehicles act as edge servers to compute tasks. A problem with the objective of maximizing the long-term utility of the VEC network was presented in the study, modeled as a Markov decision process, and solved using two reinforcement learning methods. Huang et al. [25] modeled the relationship between users, MEC server, and parking lot as a Stackelberg game. They presented a sub-gradient-based iterative algorithm to determine the workload distribution among parked vehicles and minimize the overall cost to the users. Li et al. [26] proposed a three-stage contract-Stackelberg offloading incentive mechanism to maximize the utility of vehicles, operators, and parking lot agents. Han et al. [27] proposed a dynamic pricing strategy that minimizes the average cost of the MEC system under the constraints on service quality by continuously adjusting the price according to the current system state.

By introducing parking lots as agents, many existing studies focused on utilizing the communication and computation capabilities of parked vehicles. The benefits and costs of parking vehicles and the costs to service users were taken into account. However, in addition to the vehicles parked centrally in parking lots, computing and communication channel resources of vehicles scattered on the roadside are not negligible. Moreover, in most cases, the quality of user experience should be prioritized. Therefore, in this study, based on the research work proposed in [18], we propose an PV-assisted MEC architecture to enhance the MEC network, in which parked vehicles can serve MDs directly. In addition, we propose a game-based task offloading algorithm to minimize the delay and energy consumption for service users.

3. Parked Vehicle-Assisted Multi-Access Edge Computing Architecture

With the advent of smart cars, more and more cars can be awakened to perform tasks even when parked. For example, when a parked Tesla car is in sentry mode or dog mode, some of its safety-related features are still working. With the increasing development of artificial intelligence, we believe that cars will become increasingly more intelligent. In the future, parked cars may support some modes that could provide services to other vehicles. The research presented in this study is conducted on this premise.

Although aspects such as incentives, communication costs, security, and scheduling should be considered if onboard computers in parked vehicles are to be used for edge computing, the focus of this study was on computation offloading strategy. Therefore, these aspects are not considered in this study, but should be taken into consideration in future studies to pursue a more complete solution.

A representative PV-assisted MEC service scenario is illustrated in Figure 1. There are a large number of MDs and parked vehicles running computationally intensive and delay-sensitive mobile applications. However, lightweight MEC servers and limited bandwidth resources are insufficient for these applications. Idle resources in parked vehicles can be used to relieve the pressure on the MEC. However, due to "selfishness," not all parked vehicles are willing to provide resources. We assume that some parked vehicles can be recruited through certain incentives, such as extended parking opportunities or reduced parking fees. In addition, we assume that the MEC system can certify recruited parked vehicles to ensure the security of the service and can update and monitor available resources of these parked vehicles in real time to improve resource utilization. We refer to these recruited certified parked vehicles as PVs. In summary, both MEC servers and PVs can provide services to MDs.

Figure 2 illustrates a representative PV-assisted MEC network architecture. Based on the original vehicular edge computing architecture, we move the vehicles capable of providing services from the device layer to the MEC layer to enable utilization of parked vehicles' resources and allow them to provide services directly to MDs.

(1) Cloud Layer. The first layer provides centralized cloud computing services and management functions such as critical or complex event handling, key data backup, and information authentication. The PV-assisted MEC architecture employs a software-defined network (SDN) controller to program, manipulate, and configure network in a logically centralized way.

(2) Edge Cloud Layer (MEC Layer). The second layer consists of edge network access devices (e.g., RSU and base station (BS)) and data service devices (e.g., MEC servers and PVs). Edge network access devices are used for communication among edge facilities or between layers. MEC servers with lightweight storage and computing capabilities are deployed on edge network access devices. MEC servers are responsible for collecting service status information from themselves and from PV service nodes parked in the coverage area of RSU. Based on this information, MEC servers can process or assign tasks to MDs. By moving PVs from mobile device layer to MEC layer, the service capacity can be improved and bandwidth consumption can be reduced.

(3) Mobile Device Layer. The third layer consists of mobile devices requesting services, such as vehicles, smartphones, tablets, and laptops. MDs request services by connecting to BSs via cellular network. MEC server and parked vehicles can provide services to terminal devices via cellular network or V2X. Here, V2X may be a link via cellular network or a link via dedicated short-range communications (DSRCs). Note that as a special kind of mobile device, vehicles are divided into two categories in this study: PVs and others. The former are located at the MEC layer as service providers, while the latter are located at the mobile device layer as service requesters.

Figure 3 illustrates the communication procedure between MD, MEC server, and PVs. First, when an MD generates a task, it sends the task request to the MEC server. Second, through iterative negotiation between the MEC server and the MDs, the task allocation result is calculated based on the status of the MDs and the MEC server. Then, the MEC server returns the task allocation result to the MD. When the task allocation result indicates that the task should be offloaded to a PV, the MEC also needs to notify the relevant PV (dotted arrow). Third, the MD sends task input

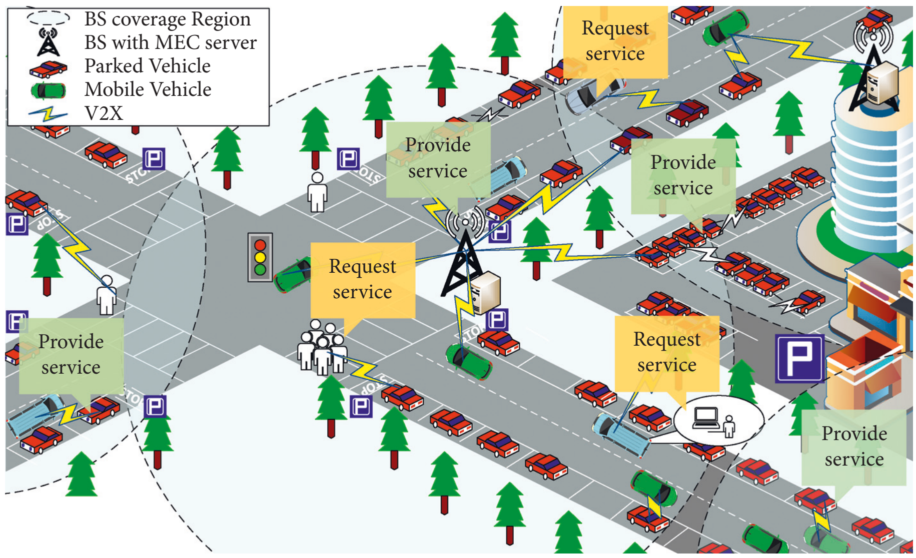


FIGURE 1: PV-assisted MEC service scenario.

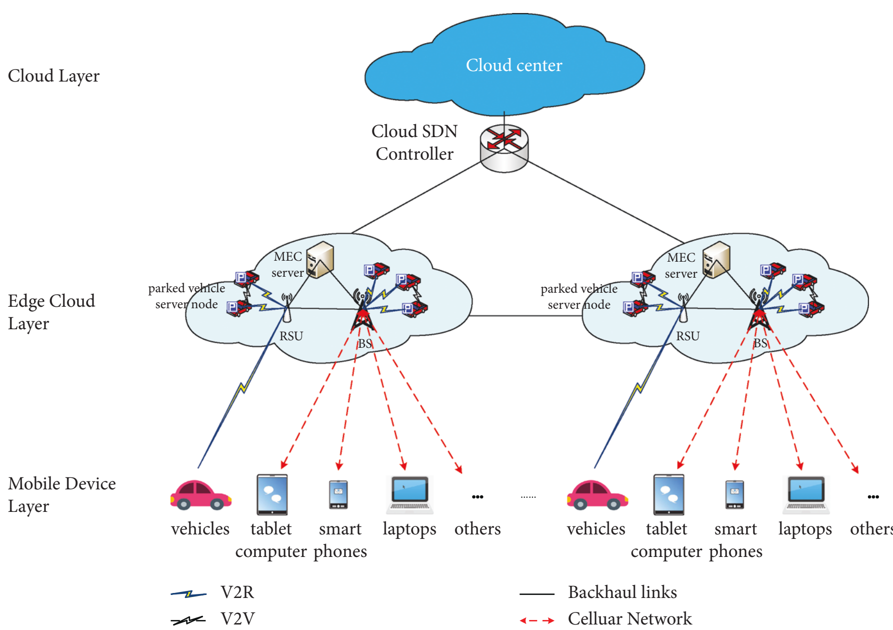


FIGURE 2: PV-assisted MEC network architecture.

data to the MEC server (solid line) or PV (dotted line) according to the task allocation information. Fourth, the MEC server (solid line) or PV (dotted line) processes the task and then returns the task result to the MD. Finally, the MD obtains the result and sends service satisfaction information back to the MEC server to reward the specific PV.

4. System Model

4.1. Network Model. We assign a unique identifier to each task and record the characteristics of tasks, such as traffic size and computation workload, in a globally shared feature table: $\mathcal{T} = \{T_1, \ldots, T_M\}$. Without loss of generality, we assume that each MD generates only one task $T_i \triangleq \{d_i, c_i\}$, $i \in \mathcal{M} = \{1, 2, \ldots, M\}$, in a time period and tasks cannot be further divided. Here, d_i denotes the size of the task generated by MD i and c_i denotes the computation resources required by this task. We assume the existence of a wireless BS through which any MD can offload its computation task to a nearby MEC server (MS). Each wireless BS has C orthogonal frequency channels, denoted as $\mathcal{C} = \{1, 2, \ldots, C\}$. Besides, in the coverage area of a BS, there is a set of PVs, denoted by $P = \{C + 1, \ldots, C + P\}$. We consider a quasi-static scenario where the status of MDs, PVs, channels, and the MEC server remains unchanged for a given time period, whereas in different time periods, the status may change. For simplicity, we ignore the cost of establishing secure connections during transmissions. We denote $s_i \in \{0\} \cup C \cup P$,

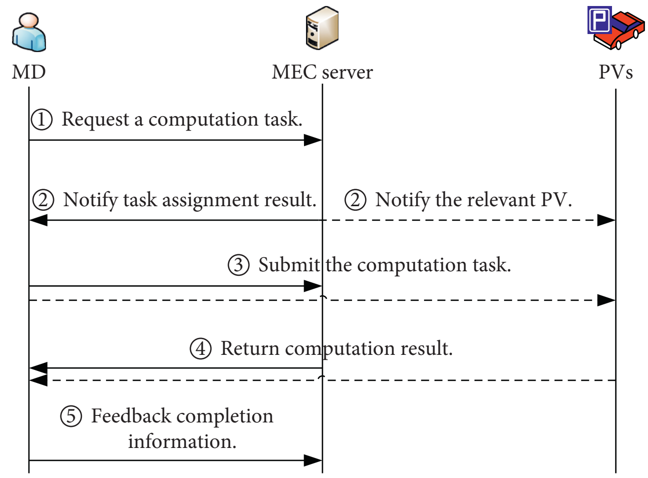


Figure 3: Sequence diagram for communication procedure.

$i \in \mathscr{M}$, as the selection decision variable. As shown in (1), let $s_i = 0$ denote that MD i executes its task locally, and $s_i > 0$ denotes that MD i chooses to offload this task. When $s_i = j$, $j \in \mathscr{C}$ indicates that MD i will offload task T_i to MEC server via channel j, while $j \in P$ indicates that the task will be executed by PV j. Let $\mathbf{s} = \{s_1, s_2, \ldots, s_M\}$ denote the set of selection decisions for all MDs. For ease of reference, we list key notations used in this study in Table 1.

$$s_i = \begin{cases} 0, & \text{if } M\ D\, i \text{ computestask } T_i \text{ locally}, \\ j, j \in C, & \text{if } M\ D\, i \text{ off loadstask } T_i \text{ to } MEC \text{ server via channel } j, \\ k, k \in P, & \text{if } M\ D\, i \text{ off loadstask} T_i \text{ to } PV\ k. \end{cases} \tag{1}$$

4.2. Communication Model. In this section, we try to define the transmission rate of offloading. It is assumed that mobile device is equipped with a single antenna that can transmit data for one task at a time. When many MDs offload their tasks to the same MEC server, severe wireless channel interference may occur. Therefore, wireless channel conditions should be considered during transmission. If MD i chooses to offload its task to the MS via wireless channel, the data transmission rate for T_i can be expressed as follows:

$$r_i^{MS} = W_M \log_2 \left(1 + \frac{q_i h_i^{MS}}{\varpi_0 + \sum_{k \in M/\{i\}:\ s_i = s_k \in C} q_k h_k^{MS}} \right). \tag{2}$$

Here, W_M is the bandwidth, q_i and h_i^{MS} are the transmission power and channel gain of MD i to the MS via nearby BS, respectively, and ϖ_0 is the background noise; $\sum_{k \in \mathscr{M}/\{i\}:\ s_i = s_k \in \mathscr{C}} q_k h_k^{MS}$ is the wireless channel interference generated by other MDs using the same channel.

MD i and PV j can communicate with each other only if the distance between them is less than a certain distance $d_{\max}^{V2V}$. We assume that any PV can only serve one MD during the computation offloading period. Therefore, there are no channel conflicts between MDs when tasks are offloaded to PVs. When MD i offloads its task T_i to PV j that is not occupied by other MDs, the data transmission rate can be expressed as follows:

$$r_i^{PV} = W_P \log_2 \left(1 + \frac{q_i h_i^{PV}}{\varpi_0} \right). \tag{3}$$

Here, W_P is the bandwidth between MD and PV.

4.3. Computation Model of Mobile Devices. We use f_i^{Loc} to denote the computational power of MD i. Thus, the delay of the locally executed task T_i can be expressed as follows:

$$t_i^{Loc} = \frac{c_i}{f_i^{Loc}}. \tag{4}$$

Similar to [28], we assume that the power consumption of a certain MD is proportional to the cube of its computational power. The energy consumption coefficient μ is related to the chip's hardware architecture. The device's energy consumption for local execution can be expressed as follows:

$$e_i^{Loc} = \mu t_i^{Loc} \left(f_i^{Loc} \right)^3. \tag{5}$$

Considering that MDs are usually energy and delay sensitive, we define parameters α_i and β_i ($\alpha_i, \beta_i \in [0, 1]$, $\alpha_i + \beta_i = 1$) as the weights for delay and energy in the computing of overhead for MD i, respectively. MDs tend to save time (larger α_i) when tasks are delay sensitive, and they tend to save energy (larger β_i) when batteries are low.

Thus, the overhead of local execution can be expressed as follows:

$$K_i^{Loc} = \alpha_i t_i^{Loc} + \beta_i e_i^{Loc}. \tag{6}$$

4.4. Computation Model of MEC Server. For most mobile applications, such as fingerprint, face, or iris recognition, and sensor data processing, the size of the computation result is much smaller than the size of the input data. We ignore the transmission time of computation results.

Table 1: Notations.

Symbol	Description
$M = \{1, 2, \ldots, M\}$	Set of mobile devices
$C = \{1, 2, \ldots, C\}$	Set of channels on the MEC server
$P = \{C + 1, \ldots, C + P\}$	Set of parked vehicles
$\mathcal{T} = \{T_1, \ldots, T_M\}$	Set of tasks generated by MD
$\mathbf{s} = \{s_1, \ldots, s_M\}$	Selection decisions set of tasks
α_i, β_i	Delay and energy weights, $\alpha_i, \beta_i \in [0, 1], \alpha_i + \beta_i = 1$
d_i	Computation data size of task T_i (bit)
c_i	Computing resources required by task T_i (cycles)
$f_i^{Loc}, f_i^{MS}, f_i^{PV}$	Computing ability of MD, MS, or PV i (Hz)
μ	Energy consumption coefficient
$d_{i,PV_j}^{V2V}, d_{i,MS}^{V2I}$	The distance between MD i and PV j /MS (m)
$d_{\max^{V2V}}, d_{\max^{V2I}}$	The maximum communication distance of V2V/V2I (m)
γ_i	Received interferences of MD i
ϖ_0	Background noise (dBm)
W_M, W_P	Uplink channel bandwidth between MD and MS/PV (Hz)
q_i	Transmission power of MD i (W)
h_i^{MS}, h_i^{PV}	Channel gain from MD i to the MS/PV
r_i^{MS}, r_i^{PV}	Transmission rate of MD i to MS/PV (bps)
$t_i^{Loc}, t_i^{MS}, t_i^{PV}$	Delay for local execution, offloading to MS/PV (seconds)
$e_i^{Loc}, e_i^{MS}, e_i^{PV}$	Energy consumption for local execution, offloading to MS/PV (joules)
$K_i^{Loc}, K_i^{MS}, K_i^{PV}$	Total overhead for local execution, offloading to MS/PV

Therefore, the delay for offloading task T_i to MEC server MS can be divided into two parts: data uploading time and task execution time, expressed as follows:

$$t_i^{MS} = t_{i,up}^{MS} + t_{i,exe}^{MS} = \frac{\mathrm{d}_i}{r_i^{MS}} + \frac{c_i}{f_i^{MS}}, \tag{7}$$

where f_i^{MS} is MS′ computing capability.

Usually, the MEC server has sufficient power supply, so the energy consumption on the MEC server can be ignored. From MD's perspective, the energy consumption of offloading task to MS comes from transmitting data over wireless network and can be expressed as follows:

$$e_i^{MS} = \frac{q_i \mathrm{d}_i}{r_i^{MS}}. \tag{8}$$

Thus, the overhead for offloading task T_i to MEC server can be expressed as follows:

$$K_i^{MS} = \alpha_i t_i^{MS} + \beta_i e_i^{MS}. \tag{9}$$

4.5. Computation Model of Parked Vehicles. Let f_j^{PV} denote the computing resource allocated to task T_i from PV j. The delay for offloading task T_i to PV $j (j \in P)$ can be expressed as follows:

$$t_i^{PV} = \frac{\mathrm{d}_i}{r_i^{PV}} + \frac{c_i}{f_j^{PV}}. \tag{10}$$

Similarly, energy consumption on PV j is ignored (which will be considered in future works), and energy consumption on MD for offloading task T_i to PV j can be expressed as follows:

$$e_i^{PV} = \frac{q_i \mathrm{d}_i}{r_i^{PV}}. \tag{11}$$

Thus, the overhead of MD i for offloading task T_i to PV j can be expressed as follows:

$$K_i^{PV} = \alpha_i t_i^{PV} + \beta_i e_i^{PV}. \tag{12}$$

5. Problem Formulation and Algorithm Design

5.1. Problem Formulation. According to Section 4, the overhead of task T_i can be expressed as follows:

$$K_i(s_i) = \begin{cases} K_i^{Loc}, & \text{if} \quad s_i = 0, \\ K_i^{MS}, & \text{if} \quad s_i = j,\ j \in C, \\ K_i^{PV}, & \text{if} \quad s_i = k,\ k \in P. \end{cases} \tag{13}$$

There are $1 + C + P$ choices available for each task. Delay and energy consumption may vary depending on offloading strategies. Therefore, the overall goal is to minimize the total overhead of all MDs. Thus, the problem of optimizing the total overhead for all MDs can be expressed as follows:

$$\begin{aligned} &\min_s \sum_{i=1}^{M} K_i(s_i), \\ &s.t. \sum_{i \in M} I_{\{s_i = 0\}} + \sum_{i \in M} I_{\{s_i \in C\}} + \sum_{i \in M} I_{\{s_i \in P\}} = M, \\ &\sum_{i=1}^{M} I_{\{s_i = j\}} \leqslant 1, \quad \forall j \in P, \\ &d_{i,PV_j}^{V2V} \leq d_{\max}^{V2V}, \quad \forall s_i = j,\ j \in P, \\ &d_{i,MS}^{V2I} \leq d_{\max}^{V2I}, \quad \forall s_i = j,\ j \in C. \end{aligned} \tag{14}$$

Here, $I_{\{E\}}$ is the indicator function with $I_{\{E\}} = 1$ if the event E is true and $I_{\{E\}} = 0$ otherwise. There are four constraints for problem (P). Constraint (C1) is that every task should be executed. Constraint (C2) is that each PV serves at most one MD. Constraint (C3) is that MD i and PV j can communicate only when they are close enough to each other. Similarly, constraint (C4) is that MD i and the MEC server can communicate only when they are close enough to each other.

The task set $\mathcal{T}$ can be divided into three mutually exclusive subsets by the selection decisions: $\mathcal{T} = \mathcal{T}_{Loc} \cup \mathcal{T}_{MS} \cup \mathcal{T}_{PV}$. $\mathcal{T}_{Loc}$ means that tasks are processed locally, $\mathcal{T}_{MS}$ means that tasks are offloaded to the MS, and $\mathcal{T}_{PV}$ means that tasks are offloaded to some PV.

By incorporating PVs as extra service providers for computation offloading, the problem proposed in this study is essentially a generalization of that proposed in [18]. However, it has been shown in [18] that the centralized optimization problem for minimizing the system-wide computation overhead is NP-hard. Therefore, with PVs as additional computation offloading providers, the problem proposed in this study is also NP-hard and difficult to solve. Similar to [18], the centralized cost minimizing problem for PV-assisted MEC computation offloading can be transformed into a distributed computation offloading decision problem among mobile device users. In the computation offloading process, each MD wants to reduce its overhead as much as possible. Therefore, they need to be aware of the choices made by other MDs. Let $s_{-i} = (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_M)$ be the selection decisions by all other MDs except MD i. Based on s_{-i}, MD i can make a proper decision s_i to reduce its overhead. The distributed computation offloading problem $((P')$ can be defined as follows:

$$\min_{s_i \in \mathbf{s}} K_i(s_i, s_{-i}), \quad \forall i \in \mathcal{M}, \tag{15}$$

in which the overhead function of mobile device i can be defined as follows:

$$K_i(s_i, s_{-i}) = \begin{cases} K_i^{Loc}, & \text{if} \quad s_i = 0, \\ K_i^{MS}, & \text{if} \quad s_i = j,\ 0 < j \leqslant C, \\ K_i^{PV}, & \text{if} \quad s_i = j,\ C < j \leqslant C + P, \quad \forall k \in \mathcal{M}/\{i\},\ s_i \neq s_k. \end{cases} \tag{16}$$

Problem p' can be formulated as a noncooperative game: $G = (\mathcal{M}, \{\mathcal{S}_i\}_{i\in\mathcal{M}}, \{K_i\}_{i\in\mathcal{M}})$ with finite players, where $\mathcal{M}$ is the set of players, $\mathcal{S}_i$ is the set of selection decisions for player/MD i, and the overhead function $K_i(s_i, s_{-i})$ is the cost function to be minimized by each MD i.

In the next subsection, we will analyze the existence of Nash equilibrium in the PV-assisted MEC computation offloading game.

5.2. Nash Equilibrium Analysis. Here is the definition of the important concept of Nash equilibrium [29].

Definition 1. A selection decision set $s^* = (s_1^*, \ldots, s_M^*)$ is a Nash equilibrium of the PV-assisted MEC computation offloading game, if at the equilibrium s^*, no MD can further reduce its overhead by unilaterally changing its selection decision, i.e.,

$$K_i(s_i^*, s_{-i}^*) \leqslant K_i(s_i, s_{-i}^*), \quad \forall s_i \in \mathcal{S}_i,\ i \in \mathcal{M}. \tag{17}$$

To study the existence of Nash equilibrium, we will first introduce the concept of potential game [30].

Definition 2. A game is said to be an ordinal potential game if the incentive of all players to change their strategy can be expressed using a single global function called the potential function: $\Phi(\mathbf{s})$, such that $\forall i \in \mathcal{M}, s_{-i} \in \prod_{k\neq i} \mathcal{S}_k$, and $s_i', s_i \in \mathcal{S}_i$, if

$$K_i(s_i', s_{-i}) < K_i(s_i, s_{-i}). \tag{18}$$

Then,

$$\Phi(s_i', s_{-i}) < \Phi(s_i, s_{-i}). \tag{19}$$

An important feature of the finite ordinal potential game is that it always has a Nash equilibrium and it has the finite improvement property. In other words, if finite players start with an arbitrary strategy profile and iteratively deviate to their unique best replies in each period, the process terminates in an NE after finite steps. Next, before giving detailed proof that the PV-assisted MEC computation offloading game is an ordinal potential game, we have the following lemma.

Lemma 1. *Given a strategy profile* $\mathbf{s}$*, MD i can reduce its computation overhead by offloading its task to the MEC server if condition* (C_m) *holds and offloading its task to PV if condition (C p) holds.*

(C_m): the received interference $\gamma_i(\mathbf{s}) = \sum_{k\in\mathcal{M}/\{i\}:\, s_i = s_k \in \mathcal{C}} q_k h_k^{MS}$ satisfies $\gamma_i(\mathbf{s}) < T_i^{ML}$ and $\gamma_i(\mathbf{s}) < T_i^{MP}$, with the following thresholds:

$$T_i^{ML} = \frac{q_i h_i^{MS}}{2^{\left(\alpha_i d_i + \beta_i q_i d_i / W_M \left(K_i^{Loc} - \alpha_i t_{i,exe}^{MS}\right)\right)} - 1} - \varpi_0, \tag{20}$$

$$T_i^{MP} = \frac{q_i h_i^{MS}}{2^{\left(\alpha_i d_i + \beta_i q_i d_i / W_M \left(K_i^{PV} - \alpha_i t_{i,exe}^{MS}\right)\right)} - 1} - \varpi_0. \tag{21}$$

(C p): PV j is not occupied by any other MD; i.e., $j \in P, \sum_{k\in M/\{i\}} I_{\{s_k = j\}} = 0$, and the computation overhead K_i^{PV}, K_i^{Loc} satisfies $K_i^{PV} < K_i^{Loc}$, and

$$\gamma_i(\mathbf{s}) > T_i^{MP}. \tag{22}$$

Proof. For condition C m: according to equation (14), we know that when the overhead satisfies $K_i^{MS} < \min\{K_i^{Loc}, K_i^{PV}\}$, i.e., $K_i^{MS} < K_i^{Loc}$ and $K_i^{MS} < K_i^{PV}$, the best strategy for MD i is to offload its task to the MEC server.

According to equations (7) to (9), the condition $K_i^{MS} < K_i^{Loc}$ is equivalent to the following:

$$\alpha_i \frac{d_i}{r_i^{MS}} + \alpha_i t_{i,exe}^{MS} + \beta_i \frac{q_i d_i}{r_i^{MS}} < K_i^{Loc}. \tag{23}$$

That is,

$$r_i^{MS} > \frac{\alpha_i d_i + \beta_i q_i d_i}{K_i^{Loc} - \alpha_i t_{i,exe}^{MS}}. \tag{24}$$

According to (2), we then have the following:

$$\sum_{k\in\mathscr{M}/\{i\}:\ s_i=s_k\in\mathscr{C}} q_k h_k^{MS} < \frac{q_i h_i^{MS}}{2^{\alpha_i d_i+\beta_i q_i d_i/W_M\left(K_i^{Loc}-\alpha_i t_{i,exe}^{MS}\right)} - 1} - \varpi_0, \tag{25}$$

which is $\gamma_i(\mathbf{s}) < T_i^{ML}$ in condition (C m).

According to equations (7) to (12), the condition $K_i^{MS} < K_i^{PV}$ is equivalent to the following:

$$\alpha_i \frac{d_i}{r_i^{MS}} + \alpha_i t_{i,exe}^{MS} + \beta_i \frac{q_i d_i}{r_i^{MS}} < K_i^{PV}. \tag{26}$$

Then, we have the following:

$$r_i^{MS} > \frac{\alpha_i d_i + \beta_i q_i d_i}{K_i^{PV} - \alpha_i t_{i,exe}^{MS}}. \tag{27}$$

Furthermore, according to (2), we can get $\gamma_i(\mathbf{s}) < T_i^{MP}$ in condition (C m).

For condition C p: the proof is straightforward and is omitted here.

Based on Lemma 1, we will show that the PV-assisted MEC computation offloading game is a potential game with the potential function as follows:

$$\begin{aligned}\Phi(\mathbf{s}) = {} & \frac{1}{2}\sum_{i=1}^{M}\sum_{k\neq i} q_i h_i^{MS} q_k h_k^{MS} I_{\{s_i=s_k\}} I_{\{s_i\in[1,C]\}} \\ & + \sum_{i=1}^{M} q_i h_i^{MS} T_i^{MP} I_{\{s_i\in(C,C+P]\}} + \sum_{i=1}^{M} q_i h_i^{MS} T_i^{ML} I_{\{s_i=0\}}.\end{aligned} \tag{28}$$

□

Theorem 1. *The PV-assisted MEC computation offloading game is a potential game with $\Phi(\mathbf{s})$ (equation 21) as the potential function and hence always has a Nash equilibrium and the finite improvement property.*

Proof. Suppose that MD $i \in \mathscr{M}$ updates its decision selection from s_i to s_i', and this leads to a decrease in the overhead function, i.e., $K_i(s_i, s_{-i}) > K_i(s_i', s_{-i})$. According to Definition 2, we must show that this also leads to a decrease in the potential function, i.e., $\Phi(s_i, s_{-i}) > \Phi(s_i', s_{-i})$. There are eight possible cases.

(1) $s_i \in [1, C]$ and $s_i' \in [1, C]$
(2) $s_i \in (C, C+P]$ and $s_i' \in (C, C+P]$
(3) $s_i \in [1, C]$ and $s_i' \in (C, C+P]$
(4) $s_i \in (C, C+P]$ and $s_i' \in [1, C]$
(5) $s_i \in [1, C]$ and $s_i' = 0$
(6) $s_i = 0$ and $s_i' \in [1, C]$
(7) $s_i \in (C, C+P]$ and $s_i' = 0$
(8) $s_i = 0$ and $s_i' \in (C, C+P]$

For case (1), according to equations (7) to (9), the premise $K_i(s_i, s_{-i}) > K_i(s_i', s_{-i})$ implies that $r_i'^{MS} > r_i^{MS}$. Then, since the function $W_m \log_2(x)$ increases monotonically with the variable x, according to equation (2), $r_i'^{MS} > r_i^{MS}$ implies that

$$\sum_{k\in M/\{i\}:\ s_i=s_k\in C} q_k h_k^{MS} > \sum_{k\in M/\{i\}:\ s_i'=s_k\in C} q_k h_k^{MS}. \tag{29}$$

Since $s_i' \in [1, C]$ and $s_i' \in [1, C]$, according to (28) and (29), we have the following:

$$\begin{aligned}\Phi(s_i, s_{-i}) - \Phi(s_i', s_{-i}) &= \frac{1}{2}\sum_{i=1}^{M}\sum_{k\neq i} q_i h_i^{MS} q_k h_k^{MS} I_{\{s_i=s_k\}} - \frac{1}{2}\sum_{i=1}^{M}\sum_{k\neq i} q_i h_i^{MS} q_k h_k^{MS} I_{\{s_i'=s_k\}} \\ &= \frac{1}{2} q_i h_i^{MS} \sum_{k\neq i} q_k h_k^{MS} I_{\{s_i'=s_k\}} + \frac{1}{2}\sum_{k\neq i} q_k h_k^{MS} I_{\{s_i=s_k\}} q_i h_i^{MS} \\ &\quad - \frac{1}{2} q_i h_i^{MS} \sum_{k\neq i} q_k h_k^{MS} I_{\{s_i'=s_k\}} - \frac{1}{2}\sum_{k\neq i} q_k h_k^{MS} I_{\{s_k=s_i'\}} q_i h_i^{MS} \\ &= q_i h_i^{MS} \sum_{k\neq i} q_k h_k^{MS} I_{\{s_i=s_k\}} - q_i h_i^{MS} \sum_{k\neq i} q_k h_k^{MS} I_{\{s_i'=s_k\}} > 0.\end{aligned} \tag{30}$$

For case (2), the premise $K_i(s_i, s_{-i}) > K_i(s_i', s_{-i})$ is equivalent to $K_i^{PV} > K_{i'}^{PV}$. According to the definition of T_i^{MP} (equation 19), if $K_i^{PV} > K_{i'}^{PV}$, then $T_i^{MP} > T_{i'}^{MP}$. Then, we have the following:

$$\begin{aligned}\Phi(s_i, s_{-i}) - \Phi(s_i', s_{-i}) &= \sum_{i=1}^{M} q_i h_i^{MS} T_i^{MP} I_{\{s_i \in (C, C+P]\}} - \sum_{i=1}^{M} q_i h_i^{MS} T_i^{MP} I_{\{s_i' \in (C, C+P]\}} \\ &= q_i h_i^{MS} T_i^{MP} - q_i h_i^{MS} T_{i'}^{MP} > 0.\end{aligned} \tag{31}$$

For case (3), since $s_i \in [1, C]$ and $s_i' \in (C, C+P]$, from Lemma 1(condition C p), we have $\gamma_i(\mathbf{s}) > T_{i'}^{MP}$. This implies that

$$\begin{aligned}\Phi(s_i, s_{-i}) - \Phi(s_i', s_{-i}) &= \frac{1}{2}\sum_{i=1}^{M}\sum_{k\neq i} q_i h_i^{MS} q_k h_k^{MS} I_{\{s_i=s_k\}} I_{\{s_i \in [1,C]\}} - q_i h_i^{MS} T_{i'}^{MP} \\ &= \frac{1}{2} q_i h_i^{MS} \sum_{k\neq i} q_k h_k^{MS} I_{\{s_i=s_k\}} + \frac{1}{2}\sum_{k\neq i} q_k h_k^{MS} I_{\{s_k=s_i\}} q_i h_i^{MS} - q_i h_i^{MS} T_{i'}^{MP} \\ &= q_i h_i^{MS} \sum_{k\neq i} q_k h_k^{MS} I_{\{s_i=s_k\}} - q_i h_i^{MS} T_{i'}^{MP} > 0.\end{aligned} \tag{32}$$

Case (4) is the opposite of case (3), and its proof is omitted here.

For case (5), since $s_i \in [1, C]$ and $s_i' = 0$, it can be deduced from $K_i(s_i, s_{-i}) > K_i(s_i', s_{-i})$ that $K_i^{MS} > K_i^{Loc}$. According to Lemma 1 (condition C m), we have $\gamma_i(\mathbf{s}) > T_i^{ML}$. Then,

$$\begin{aligned}\Phi(s_i, s_{-i}) - \Phi(s_i', s_{-i}) &= \frac{1}{2}\sum_{i=1}^{M}\sum_{k\neq i} q_i h_i^{MS} q_k h_k^{MS} I_{\{s_i=s_k\}} I_{\{s_i \in [1,C]\}} - q_i h_i^{MS} T_i^{ML} \\ &= \frac{1}{2} q_i h_i^{MS} \sum_{k\neq i} q_k h_k^{MS} I_{\{s_i=s_k\}} + \frac{1}{2}\sum_{k\neq i} q_k h_k^{MS} I_{\{s_k=s_i\}} q_i h_i^{MS} - q_i h_i^{MS} T_i^{ML} \\ &= q_i h_i^{MS} \sum_{k\neq i} q_k h_k^{MS} I_{\{s_i=s_k\}} - q_i h_i^{MS} T_i^{ML} > 0.\end{aligned} \tag{33}$$

Case (6) is the opposite of case (5), and thus, its proof is omitted here.

For case (7), $s_i \in (C, C+P]$ and $s_i' = 0$ imply that $K_i^{PV} > K_i^{Loc}$. Then, according to the definitions of T_i^{ML} (Eq. 18) and T_i^{MP} (equation 19), we have $T_i^{MP} > T_i^{ML}$. Therefore,

$$\begin{aligned}\Phi(s_i, s_{-i}) - \Phi(s_i', s_{-i}] &= \sum_{i=1}^{M} q_i h_i^{MS} T_i^{MP} I_{\{s_i \in (C, C+P]\}} - \sum_{i=1}^{M} q_i h_i^{MS} T_i^{ML} I_{\{s_i=0\}} \\ &= q_i h_i^{MS} T_i^{MP} - q_i h_i^{MS} T_i^{ML} > 0.\end{aligned} \tag{34}$$

Case (8) is the opposite of case (7), and thus, its proof is omitted here.

Combining results from the above cases, we can conclude that the PV-assisted MEC computation offloading game is a potential game. □

(1) initialize: MDs' strategy set $\mathbf{s}(\mathbf{0}) = \{0, 0, \ldots, 0\}$;
(2) **repeat**
(3) for each MD $i \in \mathcal{M}$ and each iteration t, calculate the overhead according to (16);
(4) MD i makes the best response $s_i'(t)$ as the selected strategy;
(5) **if** $s_i'(t) \neq s_i(t-1)$ **then**
(6) send update request $s_i'(t)$ to the MEC server;
(7) **if** received $s_i'(t)$ from MEC server **then**
(8) set decision selection $s_i(t) = s_i'(t)$;
(9) **else**
(10) keep the decision selection unchanged, i.e., $s_i(t) = s_i(t-1)$;
(11) **end if**
(12) **else**
(13) keep the decision selection unchanged, i.e., $s_i(t) = s_i(t-1)$;
(14) **end if**
(15) **until** received END message

Algorithm 1: Game-based PV-assisted task offloading algorithm (GPTOA).

5.3. Algorithm Design. Algorithm 1 illustrates the game-based PV-assisted task offloading algorithm (GPTOA) for problem (P'). Similar to [18], the algorithm will run iteratively on each MD. The main idea of GPTOA is that, based on the current state, each MD makes the best decision by calculating the overhead according to (16) (Line 3). Meanwhile, constraints (C1)–(C4) will be checked in each iteration. When constraints (C3) and (C4) cannot be satisfied, we set the overhead to infinity. During each iteration t, MD i updates its decision selection $s_i'(t)$ based on the best response and sends it to the MEC server as an update request, if $s_i'(t) \neq s_i(t-1)$. The MEC server randomly selects one decision selection $s_i'(t)$ from all update requests and sends $s_i'(t)$ back to MD i for updating its decision for the next iteration (Lines 4–8). The iteration continues until the decision selection remains unchanged. At the end, the MEC server will broadcast end message to all MDs and each MD will execute the computation task according to the last decision selection. According to the finite improvement property of potential game (Theorem 1), the algorithm will converge to a Nash equilibrium within finite number of iterations.

In GPTOA, MDs execute operations in parallel in each time slot. The most time-consuming operation is the computing of the best response update process in Line 3, which mainly involves the sorting operation over the overhead of available offloading strategies for all MDs. Since the sorting operation typically has a time complexity of $O(n \log n)$, and the maximum number of available choices for all MDs is not greater than $C + P + 1$, therefore, the computational complexity of each time slot will not exceed $O(x \log x)$, in which $x = C + P + 1$. If the algorithm takes y time slots to terminate, the total computational complexity of Algorithm 1 is $O(y \cdot x \log x)$.

Let $G_i = q_i h_i^{MS}$, $G_{\max} = \max_{i\in\mathcal{M}}\{G_i\}$, $G_{\min} = \min_{i\in\mathcal{M}}\{G_i\}$, and $T_{\max} = \max_{i\in\mathcal{M}}\{T_i^{ML}, T_i^{MP}\}$. For the upper bound of y, similar to [18], we have the following result.

Theorem 2. *When G_i and T_i are nonnegative integers for any $i \in \mathcal{M}$, the game-based PV-assisted task offloading algorithm will terminate within at most $(G_{\max}^2/2G_{\min})M^2 + (G_{\max}T_{\max}/G_{\min})M$ time slots, i.e., $y \leqslant (G_{\max}^2/2G_{\min})M^2 + (G_{\max}T_{\max}/G_{\min})M$.*

Proof. According to equation (21) and the definition of G_i, $G_{\max}$, $G_{\min}$, and $T_{\max}$, we have the following:

$$0 \leqslant \Phi(s) \leqslant \frac{1}{2}\sum_{i=1}^{M}\sum_{i=1}^{M} G_{\max}^2 + \sum_{i=1}^{M} G_{\max}T_{\max} = \frac{1}{2}G_{\max}^2 M^2 + G_{\max}T_{\max}M. \tag{35}$$

According to Theorem 1, during each time slot, MD $i \in \mathcal{M}$ updates its decision s_i to decision s_i' and this action leads to a decrease in its overhead function, i.e., $K_i(s_i, s_{-i}) > K_i(s_i', s_{-i})$. The key idea of this proof is to show that this also leads to a decrease in the potential function by at least $G_{\min}$, i.e.,

$$\Phi(s_i, s_{-i}) \geqslant \Phi(s_i', s_{-i}) + G_{\min}. \tag{36}$$

Similar to the proof of Theorem 1, there are eight cases to consider.

For case (1), $s_i \in [1, C]$ and $s_i' \in [1, C]$; according to (23), we have the following:

$$\Phi(s_i, s_{-i}) - \Phi(s_i', s_{-i}) = G_i\left(\sum_{k\neq i} G_k I_{\{s_i=s_k\}} - \sum_{k\neq i} G_k I_{\{s_i'=s_k\}}\right) > 0. \tag{37}$$

Since G_i are nonnegative integers, we have the following:

$$\sum_{k\neq i} G_k I_{\{s_i=s_k\}} \geqslant \sum_{k\neq i} G_k I_{\{s_i'=s_k\}} + 1. \tag{38}$$

Then, based on (37):

$$\Phi(s_i, s_{-i}) \geq \Phi(s_i', s_{-i}) + G_i \geq \Phi(s_i', s_{-i}) + G_{\min}. \tag{39}$$

For other cases, the proofs are similar and are omitted here. □

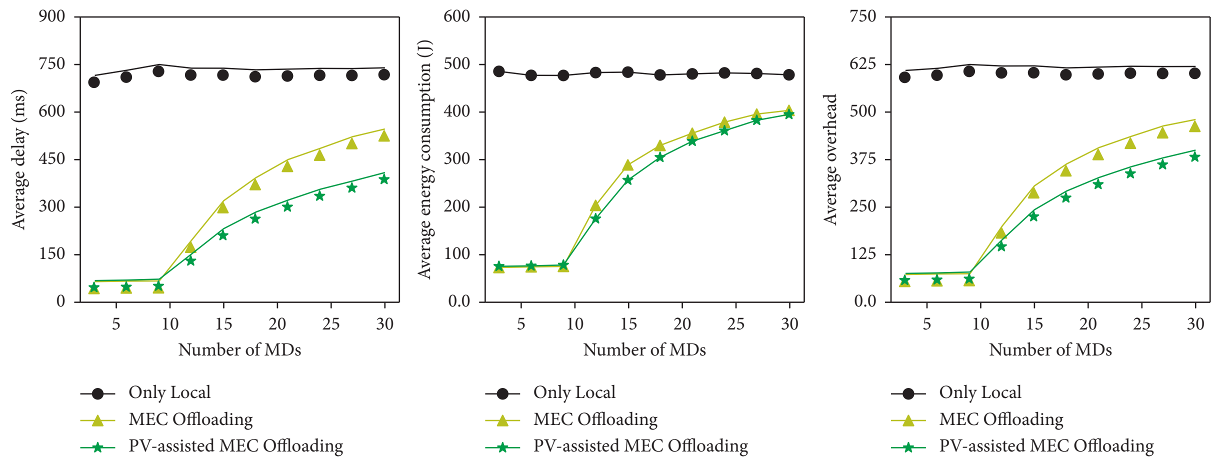


FIGURE 4: All MDs' average delay, energy consumption, and overhead with different numbers of MDs.

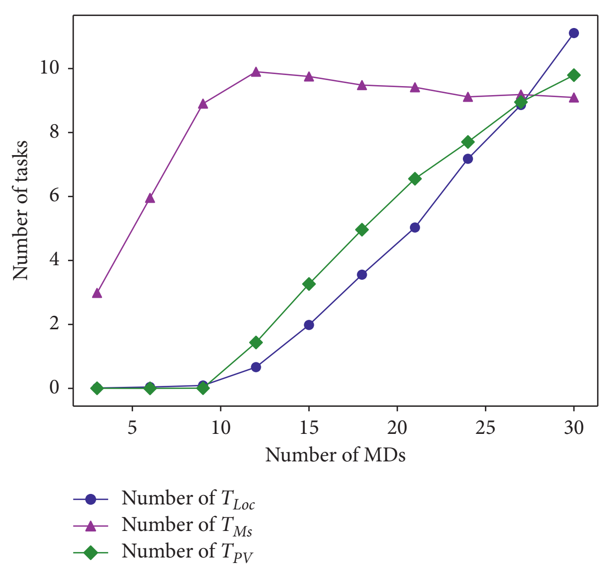


FIGURE 5: Results of task assignment with different numbers of MDs.

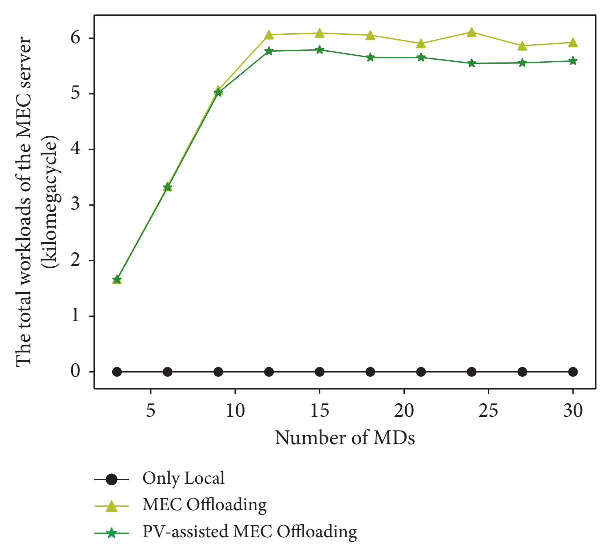


FIGURE 6: Total workload of the MEC server with different numbers of MDs.

6. Simulation Results

6.1. Parameter Settings. The GPTOA was simulated and evaluated using Python with packages such as NumPy, random, and SciPy. We considered the scenario where the wireless BS had a coverage area of 50 ∗ 50 m2. Each BS had $C = 10$ channels with a channel bandwidth of $W_M = 20$ MHz. The transmission power was $q_i = 400$ mWatts, and the background noise was $\varpi_0 = -100$ dBm. Based on the radio interference model for urban cellular radio environment, we set the channel gain to $h_i^{MS} = d_{i,r}^{-\alpha}$, where $d_{i,r}$ was the distance between MD i and the wireless BS, and $\alpha = 4$ was the path loss factor [18]. The maximum communication distances of V2X were $d_{\max}^{V2V} = 15$ m and $d_{\max}^{V2I} = 200$ m, respectively [31]. The energy consumption coefficient was $\mu = 1$.

For computational tasks, the data size of offloaded task T_i was $d_i = 1$ MB. The total number of CPU cycles (c_i) required by task T_i was randomly distributed in the interval of $(1, 1000)$ megacycles. The weight parameters for all MDs were $\alpha_i = \beta_i = 0.5$. We assumed that the values of weight parameters remained constant during a single offloading process. Since most MEC servers are equipped with multiple CPUs, and multiple CPUs can be allocated to one MD at a time, for ease of computation, it was assumed that the computation power allocated to an MD by the MEC server was $f_i^{MS} = 10$ GHz. The computation power of MDs was randomly distributed between $[0.5, 1]$ GHz. The computation power of PVs was randomly distributed between $[0.5, 1, 1.5, 2]$ GHz. The communication bandwidth between PV and MD was $W_P = 20$ MHz.

6.2. Performance Analysis. To evaluate the scheme proposed in this work, we compared three schemes: (1) local only (scheme 1): all MDs decide to compute their own tasks

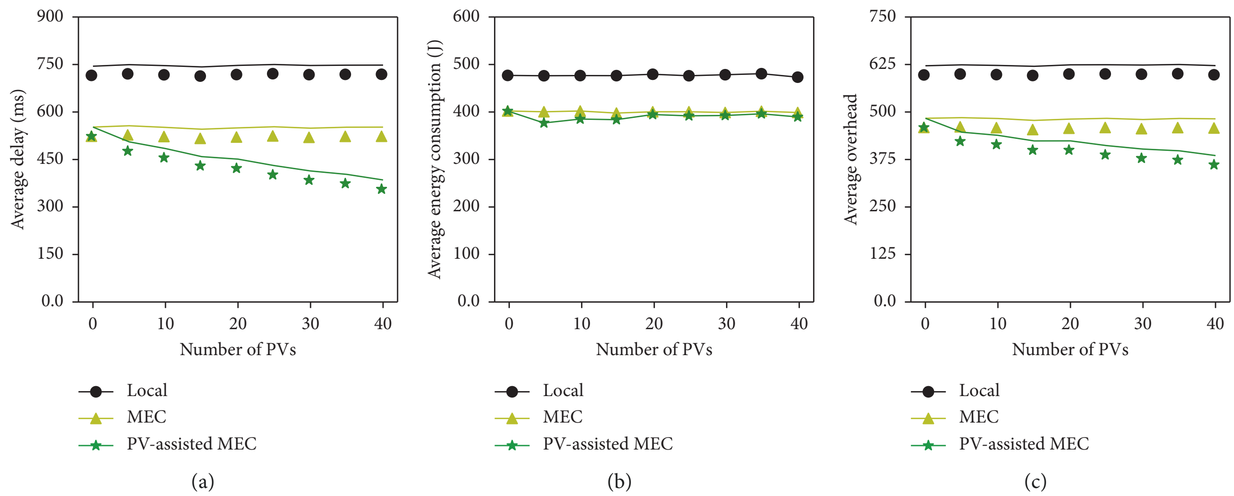


Figure 7: Average delay, energy consumption, and overhead of MDs with different numbers of PVs.

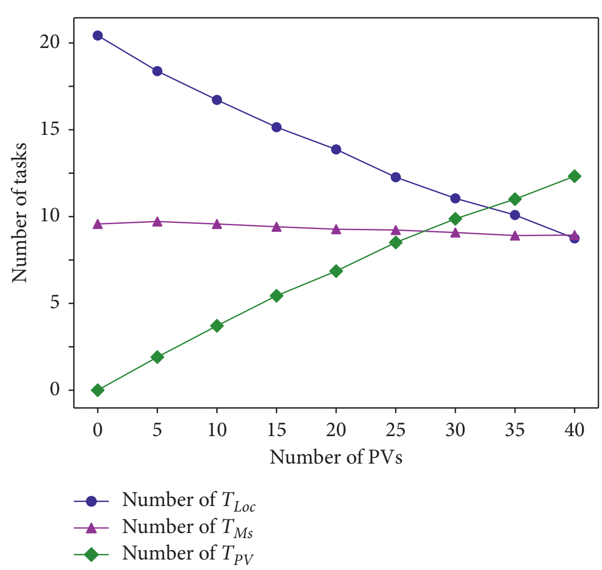


Figure 8: Task assignment of MDs with different numbers of PVs.

locally. (2) MEC offloading (scheme 2): the tasks are either computed locally or are offloaded to the MEC server [18]. (3) PV-assisted MEC offloading (our scheme): the tasks are computed locally, offloaded to the MEC or PVs. The work presented in [18] was treated as a special case with number of PVs set to 0 in this study. To eliminate the effect of randomness on the algorithm results, we conducted 1000 tests and performed statistical analysis of the results as follows.

First, we fixed the number of PVs (service vehicles) to 40 to observe the changes in the metrics (average delay, energy consumption, and total overhead of tasks, as well as task assignment results and load on the MEC server) as the number of MDs (service requesters) increased.

In Figure 4, the average delay, energy consumption, and total overhead of three schemes are compared. We can see that all three metrics of scheme 1 are higher than the other schemes due to the limited local computation power. When the number of MDs is less than 10, the three metrics are the same for schemes 2 and 3. This can be explained by the lack of tasks offloaded to PVs. As the number of MDs increases, the metrics of scheme 3 grow less rapidly than the other two schemes. When the number of MDs is 30, scheme 3 results in a 26% reduction in delay and a 17% reduction in total overhead (on average) compared with scheme 2.

Figure 5 shows the task assignment results of the proposed scheme. When the number of MDs is less than 10, all tasks will be offloaded to MS, because MS has a shorter task execution time. However, as the number of MDs increases, no more tasks can be offloaded to MS. This is due to the fact that when multiple tasks are offloaded to MS, it leads to strong channel interference and heavier computational load, which in turn causes intolerable delays. This causes some MDs to give up offloading their tasks to the MS. In addition, due to limited resources and short communication distance, only a small portion of the PVs can serve MDs. Therefore, as the number of MDs increases, eventually, the number of tasks executed locally will exceed the number of tasks offloaded to PVs.

Figure 6 shows the total workload allocated to the MEC server under the three schemes. In scheme 1, no computation tasks are offloaded, so the burden on the MEC server is 0. The workload allocated to the MEC server in scheme 3 is lower than that in scheme 2, because PVs share some of the computation tasks. When the number of MDs is 30, scheme 3 reduces the workload for MS by 5%.

Then, we fixed the number of MDs to 30 to observe the change in the metrics as the number of PVs increases.

In Figure 7, the average delay, energy consumption, and overhead of the three schemes are compared with different numbers of PVs. Scheme 3 outperforms both schemes 1 and 2. This is because as the density of PVs increases, it leads to higher utilization of idle computing resources of PVs.

Figure 8 shows the results of task allocation for the proposed scheme. As the number of PVs increases, the number of tasks offloaded to PVs also increases, while the number of locally executed tasks continues to decrease. From another perspective, as the density of PVs increases, more MDs are likely to be connected to PVs, so that MDs that have given up offloading to MS have more opportunities to offload. In addition, the computational power of the MS is much greater than that of PVs. Tasks are assigned to PVs only when MS cannot serve more tasks. Therefore, the number of tasks executed by MS will not decrease as the number of PVs increases.

7. Conclusion

In this study, we proposed a parked vehicle-assisted mobile edge computing architecture that enhances the task processing capability of MEC servers and improves the resource utilization of parked vehicles. In this work, we first discussed in detail the design principles behind the system model of PV-assisted MEC architecture, which served as a premise for the formulation of computation offloading scheme. Next, by formulating the computation offloading problem as a noncooperative game, we proposed a PV-assisted MEC computation offloading scheme that effectively reduces the burden on the MEC server. Simulation results confirmed the feasibility and high efficiency of the proposed computation offloading scheme. As mentioned in Section 3, incentives are not considered in this study; thus in the future, we will further investigate how to incorporate incentives into the PV-assisted MEC task offloading scheme proposed in this study. Deep reinforcement learning-based techniques have obvious advantages when the problem size is large or when there are multiple conflicting offloading goals [6, 32]. Therefore, another feasible research direction is to apply deep reinforcement learning to further improve the task offloading scheme proposed in this study.

Data Availability

The data used to support the findings of this study are simulated by the algorithm proposed in this article, and the parameters used in the simulation are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Qin Xin Talents Cultivation Program, Beijing Information Science & Technology University (No. QXTCP C202111).

References

[1] Y. Chen, F. Zhao, X. Chen, and Y. Wu, "Efficient multi-vehicle task offloading for mobile edge computing in 6g networks," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 4584–4595, 2022.

[2] H. Guo and J. Liu, "Collaborative computation offloading for multiaccess edge computing over fiber–wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4514–4526, 2018.

[3] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 587–597, 2018.

[4] Y. Chen, F. Zhao, Y. Lu, and X. Chen, "Dynamic task offloading for mobile edge computing with hybrid energy supply," *Tsinghua Science and Technology*, vol. 2021, Article ID 9010050, 9 pages, 2021.

[5] Ke Zhang, Y. Mao, and S. Leng, "Sabita Maharjan and Yan Zhang. "Optimal delay constrained offloading for vehicular edge computing networks"" in *Proceedings of the 16th Int. Conf. Commun. (ICC)*, pp. 1–6, IEEE, Paris, France, May 2017.

[6] Y. Chen, W. Gu, and K. Li, "Dynamic task offloading for internet of things in mobile edge computing via deep reinforcement learning," *International Journal of Communication Systems*, vol. 2022, 2022.

[7] Z. Zhou, J. Feng, Lu Tan, Y. He, and J. Gong, "An air-ground integration approach for mobile edge computing in IoT," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 40–47, 2018, https://doi.org/10.1109/mcom.2018.1701111.

[8] L. Todd, "Parking management strategies," in *Parking Management Best Practices*, pp. 86–225, Routledge, England, UK, 2018.

[9] N. Liu, M. Liu, W. Lou, G. Chen, and J. Cao, "Pva in vanets: stopped cars are not silent," in *Proceedings of the IEEE INFOCOM*, pp. 431–435, IEEE, Shanghai, China, April 2011.

[10] C. Ying, X. Hua, and M. Zhuo, "Cost-efficient edge caching for noma-enabled iot services," *China Communications*, vol. 2022, Article ID 8259817, 9 pages, 2022.

[11] L. Zhu, M. Li, Z. Zhang, and Z. Qin, "Asap: an anonymous smart-parking and payment scheme in vehicular networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 4, pp. 703–715, 2020.

[12] J. Huang, Z. Tong, and Z. Feng, "Geographical poi recommendation for internet of things: a federated learning approach using matrix factorization," *International Journal of Communication Systems*, vol. 2022, 2022.

[13] M. Li, Y. Chen, C. Lal, M. Conti, M. Alazab, and D. Hu, "Eunomia: anonymous and secure vehicular digital forensics based on blockchain," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, p. 1, 2021.

[14] J. Xu, D. Li, W. Gu, and Y. Chen, "Uav-assisted task offloading for iot in smart buildings and environment via deep reinforcement learning," *Building and Environment*, vol. 222, Article ID 109218, 2022.

[15] J. Huang, B. Lv, Y. Wu, Y. Chen, and X. Shen, "Dynamic admission control and resource allocation for mobile edge computing enabled small cell network," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 2, pp. 1964–1973, 2022.

[16] J. Zhang, H. Guo, J. Liu, and Y. Zhang, "Task offloading in vehicular edge computing networks: a load-balancing solution," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2092–2104, 2020.

[17] N. Cheng, N. Lu, N. Zhang, X. Zhang, X. S. Shen, and J. W. Mark, "Opportunistic wifi offloading in vehicular environment: a game-theory approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1944–1955, 2016.

[18] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, oct 2016.

[19] H. Ding, C. Zhang, Y. Cai, and Y. Fang, "Smart cities on wheels: a newly emerging vehicular cognitive capability harvesting network for data transportation," *IEEE Wireless Communications*, vol. 25, no. 2, pp. 160–169, 2018.

[20] J. Feng, Z. Liu, C. Wu, and Y. Ji, "Mobile edge computing for the Internet of vehicles: offloading framework and job scheduling," *IEEE Vehicular Technology Magazine*, vol. 14, no. 1, pp. 28–36, 2019.

[21] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, "Mobile-edge computing for vehicular networks: a promising network paradigm with predictive off-loading," *IEEE Vehicular Technology Magazine*, vol. 12, no. 2, pp. 36–44, jun 2017.

[22] X. Huang, R. Yu, J. Kang, Y. He, and Y. Zhang, "Exploring mobile edge computing for 5g-enabled software defined vehicular networks," *IEEE Wireless Communications*, vol. 24, no. 6, pp. 55–63, dec 2017.

[23] G. Qiao, S. Leng, K. Zhang, and Y. He, "Collaborative task offloading in vehicular edge multi-access networks," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 48–54, aug 2018.

[24] Y. Liu, H. Yu, S. Xie, and Y. Zhang, "Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, Article ID 11158, 2019.

[25] X. Huang, R. Yu, J. Liu, and L. Shu, "Parked vehicle edge computing: exploiting opportunistic resources for distributed mobile applications," *IEEE Access*, vol. 6, Article ID 66649, 2018.

[26] Y. Li, Bo Yang, Z. Chen, C. Chen, and X. Guan, "A contract-stackelberg offloading incentive mechanism for vehicular parked-edge computing networks," in *Proceedings of the 89th Veh. Technol. Conf. (VTC)*, pp. 1–5, IEEE, Kuala Lumpur, Malaysia, April 2019.

[27] D. Han, W. Chen, and Y. Fang, "A dynamic pricing strategy for vehicle assisted mobile edge computing systems," *IEEE Wireless Commun. Lett*, vol. 8, no. 2, pp. 420–423, 2019.

[28] Y. Wang, K. Wang, H. Huang, T. Miyazaki, and S. Guo, "Traffic and computation co-offloading with reinforcement learning in fog computing for industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 976–986, 2019.

[29] J. F. Nash, "Equilibrium points in n-person games," *Proceedings of the National Academy of Sciences*, vol. 36, no. 1, pp. 48-49, 1950.

[30] D. Monderer and L. S. Shapley, "Potential g," *Games and Economic Behavior*, vol. 14, no. 1, pp. 124–143, 1996.

[31] S. Chen, J. Hu, Y. Shi et al., "Vehicle-to-everything (v2x) services supported by LTE-based systems and 5g," *IEEE Communications Standards Magazine*, vol. 1, no. 2, pp. 70–76, 2017.

[32] Y. Chen, Z. Liu, Y. Zhang, Y. Wu, X. Chen, and L. Zhao, "Deep reinforcement learning-based dynamic resource management for mobile edge computing in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 4925–4934, 2021.

Hindawi
Scientific Programming
Volume 2022, Article ID 4478115, 20 pages
https://doi.org/10.1155/2022/4478115

Research Article

Research on an Artificial Intelligence-Based Professional Ability Evaluation System from the Perspective of Industry-Education Integration

Yijie Bian, Yanchi Lu, and Jingqi Li

Business School, Hohai University, Nanjing 211100, China

Correspondence should be addressed to Yanchi Lu; 1019238072@qq.com

Received 7 June 2022; Revised 1 August 2022; Accepted 6 August 2022; Published 24 August 2022

Academic Editor: Ying Chen

The rapid development of artificial intelligence technology demands higher requirements for employment and talent training. The integration of industry and education is an important way to solve the mismatch between industrial demand and talent supply. Therefore, this study starts from the perspective of the integration of industry and education. We collect recruitment texts from the perspective of "industry" and mine the specific requirements of the artificial intelligence post system through the LDA topic model and the combination of Word2Vec and *K*-means. We then conduct expert consultations and adjust the selected indicators from the perspective of "education." Finally, we construct a four-dimensional vocational ability grade evaluation index system, including basic vocational skills of artificial intelligence, database, network skills, algorithm and design skills, and research and practice skills. The intuitionistic fuzzy analytic hierarchy process, which can eliminate the subjective uncertainty of experts in the scoring process, is applied to calculate the index weights. We find that the weight of algorithm and design skill is the highest, which is an important criterion for artificial intelligence professional ability evaluation. Among the second-level indicators, practical indicators such as team spirit, innovation ability, and communication ability are the focus of investigation from the perspective of industry, while in education, the cultivation of knowledge and skills such as programming ability, applied mathematics ability, data structures, and algorithms are more important.

1. Introduction

In the 21st century, based on the advent of the Internet [1–3], the Internet of Things [4–6], big data [7], cloud computing [8, 9], and other technologies, machine learning and in-depth learning have gradually matured [10, 11]. Massive datasets have gradually taken shape and accelerated the integration of artificial intelligence and clouds [12]. At the same time, the requirements of real-time businesses have forced the artificial intelligence computing ability to continuously penetrate to the edge [13, 14] and end [15]. A new wave of artificial intelligence is rising all over the world, giving birth to a number of disruptive technologies, accelerating the cultivation of new drivers of economic development, shaping emerging industrial systems, and leading a new round of scientific and technological revolution and industrial change. The developmental trends of artificial intelligence in recent years are reflected not only in the iterative innovation of technology but also in the expansion of application fields. For example, these trends are evident in the intelligent analysis of education evaluation in the education industry [16], intelligent risk monitoring and investment prediction in the financial industry [17, 18], intelligent monitoring and control of UAVs and vehicles [19, 20], the efficient scheduling and management of dynamic resources [21–24], and other fields closely related to the economy and people's livelihood, such as intelligent cities [25, 26], intelligent medical treatment [27, 28], intelligent communication [29], and intelligent infrastructure [30, 31]. All these trends reflect the intelligent upgrading of traditional industries. According to the Artificial Intelligence and Life in 2030 Report of the 2015 Study Panel of Stanford University, artificial intelligence has been rapidly developed and highly applied in many fields, including transportation, family services, health care, education, public safety, work, and employment.

The rapid development of artificial intelligence technology and applications has had a great impact on employment and talent training. On the one hand, because technological progress improves productivity increases, some labor can be directly replaced by machines, and employment discrimination is becoming increasingly serious [32]. On the other hand, artificial intelligence promotes the expansion of the industrial scale and the upgrading of structure, thus creating new jobs and new employment opportunities and improving work quality [33]. Gartner, a consulting firm, predicts that from 2020, the number of jobs created by artificial intelligence will exceed the amount of unemployment caused by it. In particular, it will create 2.3 million new jobs while "eliminating" 1.8 million jobs. The development of artificial intelligence puts forward new requirements for the talent market. These new requirements also transfer the power of change to the talent training of colleges and universities.

The development of artificial intelligence benefits from the requirements of economic development for new technology, while the demand for emerging talent brought by technological development deepens the integration of industrial and educational reforms in talent training. The integration of industry and education helps to complement the long-standing shortcomings of talent training at all levels and provides strong talent support for economic transformation and upgrading and technological iteration. It will also help to meet the actual requirements of those to be employed to master the skills required in the industry. Therefore, this paper evaluates the professional ability of AI practitioners from the perspective of the integration of industry and education and constructs an AI professional ability evaluation index system. Our specific research route is as follows:

(i) First, strong information support is obtained through literature research and text mining. The late Dirichlet allocation (LDA) topic model is used to divide different types of artificial intelligence posts. The combination of Word2Vec and K-means can extract the career needs of different posts. Based on this, the index factors of the evaluation system are preliminarily conceived and designed.

(ii) Second, the index is pruned or modified in combination with expert consultations. Furthermore, the reliability and validity of the questionnaire are tested through a large sample.

(iii) Then, the weights of each level and each index of the index system are calculated by the fuzzy analytic hierarchy process method.

(iv) Finally, combined with the research results, this paper analyses the emphasis on different indicators in the process of artificial intelligence professional ability evaluation and puts forward corresponding suggestions for artificial intelligence industry enterprises and talent training institutions.

The research framework is shown in Figure 1.

This paper evaluates the professional ability of artificial intelligence from the new perspective of the integration of industry and education, creatively takes the text mining results as one of the bases for the selection of evaluation indicators, and makes the following main contributions:

(i) The constructed artificial intelligence professional ability evaluation index system provides a reference for the artificial intelligence industry to evaluate the professional ability of relevant talent. It also provides guidance and a basis for human resource management and recruitment for artificial intelligence enterprises.

(ii) The constructed index system defines the future learning and development directions for the employees in the artificial intelligence industry. They can use the evaluation index system to self-evaluate and choose the appropriate fields to deepen their learning and thereby improve their overall quality and skill levels.

(iii) In the process of constructing the index system, different categories of posts in the artificial intelligence industry have been identified. The skill needs of various dimensions of the artificial intelligence industry have also been clarified. As a result, various talent training institutions can formulate artificial intelligence talent training programs according to their basic education facilities, scientific research education levels, and education development orientations. By improving students' skill levels, their employment rate will also improve.

The rest of this paper is organized as follows: Section 2 summarizes the relevant research. The characteristics of artificial intelligence vocational ability evaluation from the perspective of industry-education integration are described in Section 3. Section 4 presents the main research methods used in this study. Section 5 is the most important part of the article. In this section, the preliminary selection of indicators is completed based on the text mining results, and the final indicator system is determined through the Delphi method and a questionnaire survey. Then, the index weights of the constructed system are calculated in Section 6. Finally, in Section 7, we analyze and summarize the research results and put forward some suggestions regarding the reform of talent training.

2. Related Work

2.1. The Demand for Talent Training in Artificial Intelligence. Professional ability evaluation is a worldwide issue that plays an important role in students' employment and further education, enterprise recruitment, and the development of the educational ability of colleges and universities. A large amount of research has been conducted on the talent demand for artificial intelligence and the evaluation and reform of talent training from the perspective of industry-education integration. Wang and Ren [34] divided the types of artificial intelligence posts into three levels: the basic level, technical level, and application level. Li and Chen [35]

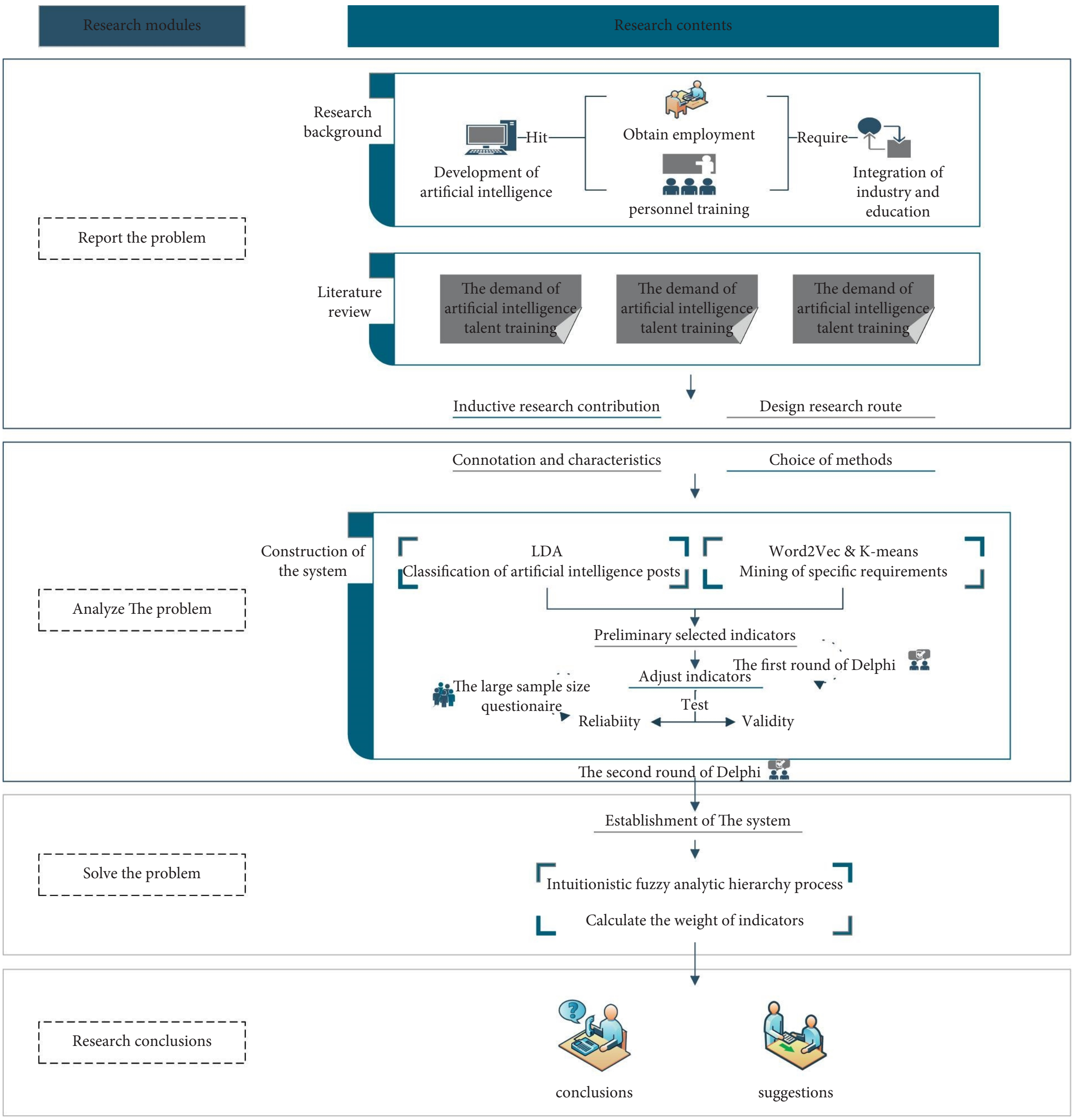


FIGURE 1: The research process framework diagram.

divided the types of artificial intelligence talent into three types: the basic research type, application practice type, and technology R&D type, with different emphases on the needed abilities. Huang [36] stated that based on the development orientation and professional characteristics of the artificial intelligence specialty, its professional knowledge structure should be considered at five levels: the infrastructure layer, core technology layer, supporting technology layer, system platform layer, and application layer.

2.2. Mismatch between Education and the Labor Market. Existing studies also pay attention to the inherent mismatch between education and the labor market. The reasons for this mismatch and the resulting waste of human resources, including "overeducation" and "insufficient education," have been studied in [37]. This mismatch has a negative impact on the wage levels of employees and the national macroeconomy [38]. In the era of artificial intelligence, the mismatch between people and positions has not been solved, but it has become more serious. According to the European Centre for the Development of Vocational Training (CEDEFOP), by 2025, approximately 48% of jobs in Europe will only be for employees with higher education qualifications, and 85% of European jobs will require at least one basic digital skill. In the period of rapid technological change, to fill the skill gap and help individuals maintain their employment success rate and continuous progress, individuals need to receive lifelong vocational education. In addition, the intelligent reform of talent training strategies in educational institutions is necessary to solve problems such as weak discipline, specialty construction, and the insufficient innovation of talent training modes [39]. Thus, the integration of industry and education is a good direction for talent training reform.

2.3. Reform Based on the Integration of Industry and Education. In December 2017, the general office of the State Council issued "Several Opinions on Deepening the Integration of Industry and Education." It started with the construction of an overall integrated development pattern of education and industry that will promote the reform of talent training for the integration of industry and education and the two-way connection between the supply and demand of industry and education. This is intended to promote the organic connections between the education chain, talent chain, industrial chain, and innovation chain. The aim is to realize the sharing of resources and information between schools and enterprises and realize "win-win" outcomes for school education, enterprises, and students. Subsequently, Dong and Huang [40] proposed that vocational education in the era of artificial intelligence should develop intelligently on the basis of the existing cooperation mode, group mode, industry-education integration mode, and other collaborative development. Hu [41] pointed out that the essence of industry-education integration is to create connections between school education and social needs through complementary advantages and resource sharing between enterprises and universities. Through innovative organizational forms and production management modes, one must realize new technology development, achievement transformation, and collaborative education. Ren and Liu [42] proposed that to build a talent training platform integrating industry and education, colleges and universities should actively establish contacts with enterprises, apply theoretical research results to the research and development of artificial intelligence products, and adopt the "system of bilateral contractual employment" to appoint teachers in the field of artificial intelligence.

2.4. Research Review. Looking at the existing related studies, it is found that traditional and emerging vocational ability evaluation index systems have similarities but are also different in their dimensions and methods. It is also clear that there is no clear view on the professional ability evaluation of talent in the artificial intelligence industry to solve the long-term mismatch between education and the labor market. The integration of industry and education is an effective angle to solve the severe mismatch between talent training and demand, and it has become an active new direction of higher education reform research. The promotion of related policies has formulated the "$1 + X$" technical talent training standard and framework for a new wave of higher vocational education research. However, it has not been applied to the research of artificial intelligence talent training reform and talent professional ability evaluation. Therefore, this paper constructs a talent professional ability evaluation index system, which provides a reference for developing artificial intelligence talent in colleges and universities and the employment criteria of enterprises.

3. Connotation and Characteristics

3.1. Artificial Intelligence Professional Ability Evaluation. Professional ability is a collection of various skills that should be possessed in a certain occupation. It is not only the ability of people to comprehensively use knowledge, experience, and skills to complete professional tasks but also the basic requirement of competence for a job. Professional ability evaluation needs to measure and evaluate specific groups according to the ability needs of professional talent and the corresponding evaluation system. In the same industry, due to different job functions, their professional ability requirements can be different. Talent in the field of artificial intelligence not only must possess general basic abilities but also must master the professional knowledge of artificial intelligence and hone professional skills according to the individuals' own professional plans. Other requirements are R&D ability, the ability for continuous learning, and the ability to integrate all aspects of experience and resources in innovation. The integration of industry and education occurs due to the in-depth cooperation carried out by colleges and universities to improve the quality of talent training and to meet the talent needs of enterprises. The two sides complement each other to realize the organic connection between the college education chain, talent chain, enterprise industry chain, and innovation chain. The existing research on professional ability evaluation for an industry often takes the enterprise as the perspective and the professional skill level as the focus to build the index system, while college education often takes academic achievements as important standards for talent training evaluations. Therefore, the evaluation of artificial intelligence professional ability should separate professional ability from the needs of college training and the artificial intelligence industry, set ability standards, and build professional ability evaluation standards in line with the needs of industrial development.

3.2. Purpose and Characteristics of Artificial Intelligence Professional Ability Evaluation. The purpose of artificial intelligence professional ability evaluation is to provide a basis for human resource management and the talent recruitment of artificial intelligence enterprises. Second, it must help practitioners conduct self-assessment, help choose areas for deeper learning, and provide a reference framework for self-development. Third, it should provide a reference basis for the cultivation of artificial intelligence talent in colleges and universities, help optimize the artificial intelligence curriculum, and cultivate "artificial intelligence +" talent in various industries. To ensure that the evaluation results of AI professional ability from the perspective of industry-education integration are meaningful, we must be aware of the characteristics of the evaluation process.

(1) Diversity of evaluation indicators:

According to the theory of multiple intelligences, the ways of human thinking and understanding are diverse, and the ability to solve a problem or create a product is also diverse. Therefore, there should also be diversity in the evaluation of talent functions. The professional ability of artificial intelligence talent often carries out a comprehensive and diversified evaluation from the aspects of basic quality, professional skills, research practice, and so on. Accordingly, scientific and diverse methods, such as self-evaluation, evaluation by others, quantitative indexes, and fuzzy evaluation should also be used in the evaluation process to ensure the accuracy of evaluation and the effectiveness of results.

(2) Hierarchy of evaluation systems:

Industry practitioners have different work fields and are required to perform different tasks. Different posts often imply certain differences in competency levels. The required vocational skills are different and have different hierarchical standards. Therefore, the artificial intelligence vocational ability evaluation system should also be hierarchical to meet the needs of artificial intelligence posts at different levels.

(3) Comprehensive evaluation of experts:

From the perspective of the integration of industry and education, the construction of the evaluation system and the selection of evaluation experts should be comprehensive. The professional ability of talent should be evaluated from the perspective of education and employment. One must comprehensively evaluate the professional ability of the evaluated object, make the ability level of employees clear, clarify the ability gap and deepen the learning direction.

4. Methods

4.1. Text Mining. The era of big data has driven an increase in electronic text information, which contains a large amount of valuable intelligence. Text mining refers to the process of applying technologies and algorithms to extract potential and valuable knowledge from a large number of unstructured or semistructured text sets. The emergence and development of its key technologies are based on massive text information. In addition, in the existing research on vocational ability evaluation, the selection of indicators often relies on literature research, summarizes the commonly used indicators in previous research, or adds interviews and consultations with experts in the field. Taking the mining results of enterprise recruitment texts as the basis for the selection of indicators from the perspective of "industry" is more practical and persuasive than making decisions through literature research or expert consultation.

The data volume of enterprise recruitment text is large, but it has a relatively clear text structure. The LDA topic model in the text mining method can project high-dimensional sample data into the optimal classification vector space. The extracted topic subspace has a larger distance between different categories and a smaller distance within the same category to complete the classification of different artificial intelligence positions. The enterprise recruitment text also has a short length and the text has a strong semantic relevance. The Word2Vec word vector model can convert natural language symbols that cannot be directly understood by computers into specific vectors that can be recognized by computers and take into account the semantic links between words. The K-means clustering algorithm has higher efficiency and accuracy in processing short texts. Therefore, this study combines Word2Vec with K-means, which can effectively solve the problem of clustering the information on professional ability needs in the short text of enterprise recruitment represented by the vocabulary.

(1) Latent Dirichlet Allocation (LDA):

The latent Dirichlet allocation (LDA) topic model is a three-layer Bayesian topic model with a "text-topi-words" structure [43]. The bag-of-words method is used to convert the text into word frequency vectors. Assuming that several topics form a document according to a certain probability and several words determine the topic according to a certain probability, document to topic and topic to vocabulary obey a polynomial distribution. The mathematical description of the model is as follows:

Step 1: For each document $d \in D$, the document length is N, and the subject probability distribution of the sampled generated document d is $\theta_d \sim \text{Dirichlet}(\alpha)$.
Step 2: Subject $Z_{d,i}\,(i \in \{1, 2, 3 \ldots N_d\})$ can be obtained according to multinomial distribution $Z_{d,i} \sim \text{Multinomid}(\theta_d)$.
Step 3: For each topic $z \in K$, the lexical probability distribution of the sampled topic z is $\phi Z_{d,i} \sim \text{Dirichlet}(\beta)$.
Step 4: According to the multinomial distribution $W_{d,i} \sim \text{Multinomid}(\phi Z_{d,i})$, the word $W_{d,i}\,(i \in \{1, 2, 3 \ldots N_d\})$ can be obtained.

The joint distribution formula of the whole model is expressed as formula (1):

$$p(w, z, \theta_d, \phi_k | \alpha, \beta) = \prod_{i=1}^{N} p(\theta_d | \alpha) p\left(Z_{d,i} | \theta_d\right) p(\phi_k | \beta) \cdot p\left(W_{d,i} | \phi Z_{d,i}\right). \quad (1)$$

In this formula, α, β and k are preset hyperparameters; normally, α and β are default values, and k is the most appropriate number of topics. θ, φ and z are the latent parameters to be inferred and calculated and w is the observation parameter.

The graphic representation of LDA is shown in Figure 2.

The word segmentation results are converted into structured data, and this is taken as the object to

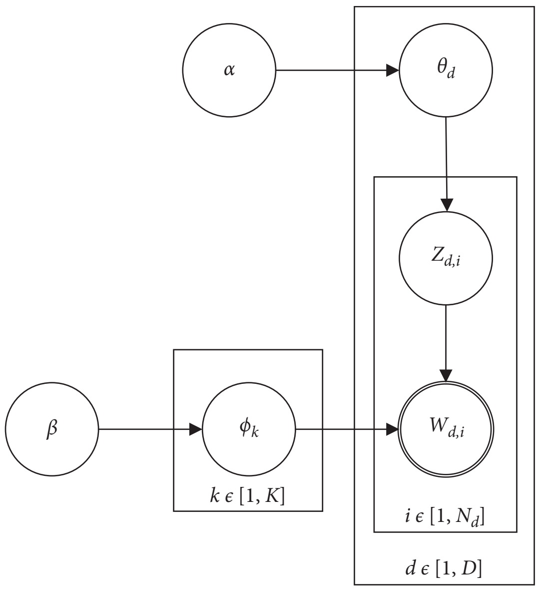


FIGURE 2: Graphic representation of LDA topic model.

build the LDA topic model. Then, the method of calculating perplexity is used to assist in determining the optimal topic number k of the four types of text. The calculation formula of perplexity is shown in:

$$\text{Perplexity}(D) = \exp\left\{-\frac{\sum_{d=1}^{M} \log P(W_d)}{\sum_{d=1}^{M} N_d}\right\}, \tag{2}$$

where D is the test document set, W_d is the vocabulary sequence of document d, P (W_d) is the probability of W_d in document d, and N_d is the vocabulary number of document d.

(2) Combination of Word2Vec and K-means:

Word2Vec is an open-source computing tool based on the idea of deep learning and has a three-layer neural network language model structure of "input layer-hidden layer-output layer." It can convert vocabulary into a high-dimensional vector through the training of a text dataset to realize the feature quantization of vocabulary text [44]. Word2Vec fully considers the relationship between words and context and includes two common models: the CBOW model and the skip-gram model. The CBOW model takes the long following words of the target vocabulary as the input to predict the target vocabulary probability, while the Skip-Gram model takes the target vocabulary as the input to predict the context vocabulary probability. The selected text dataset is the recruitment demand text in the field of artificial intelligence. The semantic tendency of the dataset is relatively concentrated, and the vocabulary frequency of ability demand is high and has a strong connection with the context. Therefore, this study selects the CBOW model to embed high-frequency keywords with a frequency of more than 300. The structure of the CBOW training model is shown in Figure 3, where ω represents the word in the text.

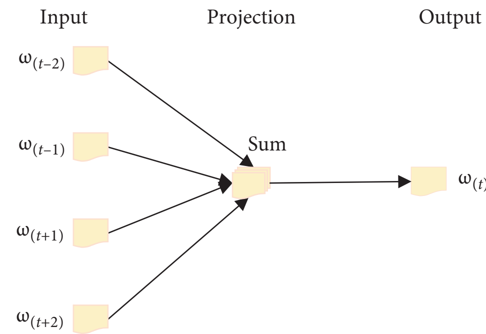


FIGURE 3: The structure of the CBOW training model.

The K-means algorithm is a dynamic clustering algorithm based on sample distance. It reduces the similarity between clusters through iterative optimization of unsupervised learning of clustering rules. Compared with other clustering algorithms, the K-means algorithm is simpler and more effective, with low computational time complexity and strong stability when processing high-dimensional data [45]. In this study, the silhouette coefficient is used as the judgment basis for the selection of the k value of clustering. The calculation formula of the individual silhouette coefficient is shown in:

$$S_i = \frac{b_i - a_i}{\max(a_i, b_i)}, \tag{3}$$

where a_i is the average distance between sample i and other samples in the same category, and b_i is the average distance between sample i and its nearest internal samples in other categories. The average value of the individual silhouette coefficient of all samples is the global silhouette coefficient, and the calculation formula is as follows:

$$S_k = \frac{1}{n}\sum_{i=1}^{n} S_i. \tag{4}$$

The character n represents the number of samples, and k represents the number of clusters. The higher the S_k value is, the more reasonable the sample clustering result.

Word2Vec is combined with K-means to complete the text mining task from vectorization to final clustering to obtain different categories of all words in the text.

4.2. Intuitionistic Fuzzy Analytic Hierarchy Process. The traditional analytic hierarchy process has both significant advantages and disadvantages. While the evaluation process is intuitive and concise, it depends too much on the subjective experience of experts. Based on this, Van Laarhoven and Pedrycz [46] fuzzified the analytic hierarchy process, and Buckley [47] formally proposed the fuzzy analytic hierarchy process on this basis. The fuzzy analytic hierarchy process combines the inclusiveness of the fuzzy set and the

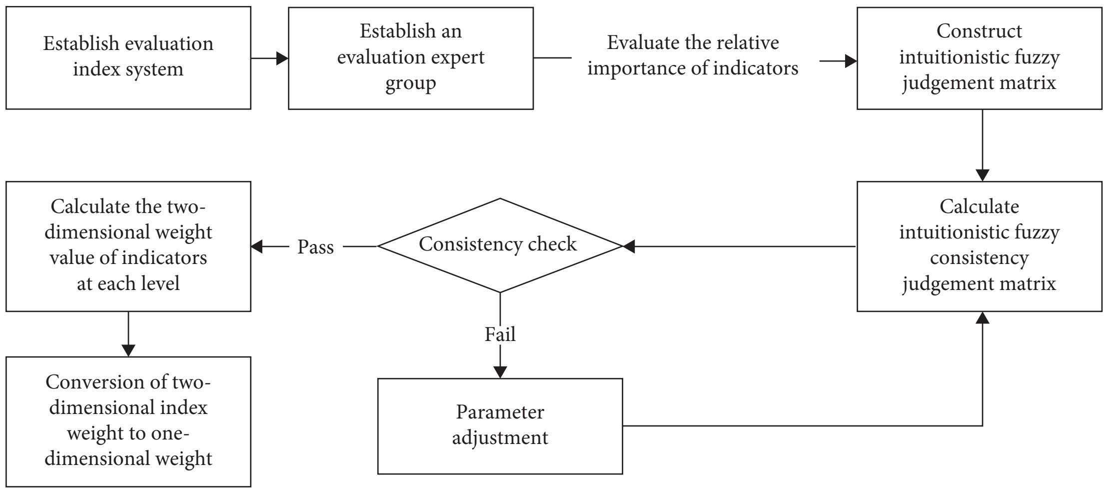


Figure 4: Flow chart of the intuitionistic fuzzy analytic hierarchy process.

quantification of the analytic hierarchy process. This reduces the influence of experts' subjective factors, but its judgment matrix has difficulty achieving the expected results in the consistency test and has limitations on the expression of experts' hesitation in the process of judgment. Atanassov [48] extended the intuitionistic fuzzy set theory to the three states of "subordinate," "nonsubordinate," and "hesitation" in the intuitionistic fuzzy set. This allows the expert evaluation to have the three attitudes of support, opposition, and neutrality. It is integrated into the analytic hierarchy process to form the intuitionistic fuzzy analytic hierarchy process to make up for the shortcomings of the past methods. In this paper, the intuitionistic fuzzy analytic hierarchy process is used to calculate the weight, and the specific process is shown in Figure 4.

5. Construction of the Index System

Based on the characteristics of artificial intelligence professional ability evaluation and following the principles of scientific, comprehensive, feasible, qualitative, and quantitative combination, this paper collects and analyses enterprise recruitment texts, integrates and divides the types of artificial intelligence posts, interprets the professional needs of different posts, analyses the common indicators of professional ability evaluation in combination with expert consultation and literature research, and comprehensively constructs a hierarchical index system of professional ability evaluation of artificial intelligence talent.

5.1. Artificial Intelligence Post Type Extraction Based on LDA. This paper uses a program compiled by Python to obtain 12531 enterprise recruitment texts retrieved with "artificial intelligence" as the keyword from a recruitment website. The program deletes duplicate records, records of posts in the non-artificial intelligence industry, and records with incomplete text content and finally retain 10,138 recruitment texts. In the text preprocessing stage, the keywords "job requirements," "application requirements," "qualification," "position request," "hire requirements," and other keywords are replaced by "position requirements." The program is used to divide the post tasks and post requirements within the boundary of "position requirements." We manually handle text records that do not meet this format as exceptions and find and delete irrelevant information such as workplace, welfare benefits, and company profile through regular expression.

Different AI posts have their own preferences for the educational level of talent. Matching the processed recruitment text with the minimum educational requirements of the post through keywords helps to extract different AI posts from a large number of recruitment texts. After matching, 598 text records indicated positions requiring an associate degree or above, 4,752 records requiring a bachelor's degree or above, 3,217 records requiring a master's degree or above, 202 records requiring a doctoral degree, and 1,369 records not indicating academic requirements are obtained. We then randomly select 500 recruitment information as the dataset, carefully read and built the dictionary of common terms in the recruitment text, enrich the stop word list, segmented the four types of text, and remove the stop words in the word segmentation results.

The value of perplexity generally decreases gradually with the increase in k, the number of potential topics. A lower degree of confusion represents a higher model generation ability. Therefore, the k value of the lower point or inflection point of the curve describing perplexity is selected to substitute into the LDA training model to judge the text topic classification results. The results show that the optimal number of topics for texts with a minimum requirement of an associate degree is 3, the optimal number of topics for texts with a minimum requirement of a bachelor's degree is 5, and the optimal number of topics for texts with a minimum requirement of a master's degree is 5, and the optimal number of topics for texts with a minimum requirement of a doctoral degree is 2. Because different enterprises have different educational requirements for the same post, the text classification

Table 1: Artificial intelligence post topics extracted by LDA topic model.

Order number	Educational requirements	Subject words	Subject	Positions included
A1	Associate degree or above	0.012 * automation + 0.010 * communication + 0.009 * debugging + 0.008 * responsibility + 0.008 * computer + 0.007 * team spirit + 0.007 * project + 0.007 * responsible + 0.006 * relevant + 0.006 * electronic + 0.006 * office + 0.005 * maintenance	Basic application post	Product sales Account manager R & D assistant
A2		0.025 * automation + 0.016 * development + 0.014 * robot + 0.013 * C# + 0.013 * correlation + 0.012 * computer + 0.011 * debugging + 0.011 * algorithm + 0.011 * C++ + 0.010 * programming + 0.009 * business trip + 0.009 * image processing	Technical support post	Test engineer
A3		0.013 * computer + 0.010 * system + 0.009 * technology + 0.009 * related + 0.008 * development + 0.007 * automation + 0.007 * responsibility + 0.006 * algorithm + 0.006 * learning + 0.005 * communication + 0.005 * electronics + 0.005 * attitude	Application development post	Search engine engineer NLP engineer
B1	Bachelor's degree or above	0.026 * ability + 0.018 * experience + 0.015 * technology + 0.012 * communication + 0.011 * computer + 0.011 * software + 0.011 * development + 0.009 * design + 0.009 * product + 0.009 * responsibility + 0.008 * project + 0.007 * work experience	Technical support post	Technical support engineer Product manager
B2		0.024 * algorithm + 0.024 * control + 0.022 * robot + 0.018 * development experience + 0.017 * C++ + 0.015 * ROS + 0.013 * experience + 0.013 * development + 0.012 * computer + 0.011 * mathematics + 0.011 * system + 0.010 * C	Application development post	Search engine engineer NLP engineer Speech recognition engineer
B3		0.031 * development + 0.019 * programming + 0.016 * development experience + 0.015 * Linux + 0.014 * experience + 0.014 * embedded + 0.014 * C++ + 0.013 * C + 0.011 * system + 0.011 * platform + 0.009 * electronics + 0.009 * framework	Application development post	Embedded engineer
B4		0.027 * algorithm + 0.025 * image processing + 0.021 * machine vision + 0.021 * experience + 0.019 * vision + 0.018 * C++ + 0.017 * automation + 0.014 * computer + 0.011 * image + 0.011 * development + 0.010 * capability + 0.010 * C	Application development post	Computer vision engineer
B5		0.031 * algorithm + 0.019 * ability + 0.018 * mathematics + 0.018 * deep learning + 0.017 * machine learning + 0.016 * computer + 0.015 * experience + 0.015 * python + 0.012 * C++ + 0.010 * domain + 0.009 * technology + 0.009 * language	Algorithm research post	Machine learning direction Data mining direction

Table 1: Continued.

Order number	Educational requirements	Subject words	Subject	Positions included
C1	Master's degree or above	0.028 * algorithm + 0.020 * robot + 0.019 * control + 0.015 * C + + + 0.012 * experience + 0.012 * C + 0.012 * automation + 0.011 * planning + 0.011 * capability + 0.010 * development experience + 0.010 * computer + 0.009 * ROS	Application development post	Search engine engineer NLP engineer Speech recognition engineer
C2		0.031 * algorithm + 0.023 * signal processing + 0.017 * experience + 0.013 * C + + + 0.012 * communication + 0.012 * C + 0.012 * electronics + 0.012 * capability + 0.011 * mathematics + 0.010 * simulation + 0.010 * computer + 0.010 * MATLAB	Application development post	Embedded engineer
C3		0.044 * image processing + 0.032 * algorithm + 0.025 * image + 0.023 * C + + + 0.018 * pattern recognition + 0.016 * C + 0.016 * computer + 0.015 * math + 0.015 * vision + 0.014 * detection + 0.014 * OpenCV + 0.014 * programming	Application development post	Computer vision engineer
C4		0.028 * algorithm + 0.020 * machine learning + 0.020 * ability + 0.019 * deep learning + 0.017 * experience + 0.015 * domain + 0.014 * computer + 0.014 * mathematics + 0.013 * Python + 0.011 * technology + 0.009 * model + 0.009 * C++	Algorithm research post	Machine learning direction Data mining direction
C5		0.013 * ability + 0.012 * deep learning + 0.012 * C++ + 0.011 * domain + 0.011 * computer + 0.011 * algorithm + 0.010 * C + 0.010 * python + 0.010 * experience + 0.009 * computer vision + 0.009 * publication + 0.009 * conference	Senior R & D post	Team leader
D1	Doctoral degree	0.029 * algorithm + 0.015 * experience + 0.015 * deep learning + 0.014 * mathematics + 0.012 * calculation + 0.011 * C++ + 0.010 * artificial intelligence + 0.010 * C + 0.009 * computer + 0.009 * machine learning + 0.008 * ability + 0.008 * python	Senior R & D post	Senior algorithm researcher
D2		0.019 * ability + 0.013 * experience + 0.012 * algorithm + 0.011 * field + 0.011 * technology + 0.010 * machine learning + 0.009 * deep learning + 0.008 * computer + 0.008 * project + 0.008 * publication + 0.008 * paper + 0.007 * artificial intelligence	Senior R & D post	Artificial intelligence expert

results of different educational requirements overlap. The combined AI post topics are shown in Table 1.

We carefully read the original text corpus used to refine the topics of different AI posts and integrate five main types of AI posts by analyzing their post name, job description, educational requirements, and ability requirements, including basic application posts, technical support posts, application development posts, algorithm research posts, and senior R&D posts. Among the extracted topics, $A1$ covers most of the basic application posts, including product sales, account managers, and R&D assistants. The topics of $A2$ and $B1$ are technical support posts, of which $A2$ mainly includes test engineers, and $B1$ mainly includes technical support engineers and product managers. The $A3$, $B2$, $B3$, $B4$, $C1$, $C2$, and $C3$ topics are application development posts. Topic $B3$ and $C2$ identify embedded development posts responsible for underlying development and software embedding. Machine vision development direction post requirements are reflected in topics $B4$ and $C3$. Search engine development, natural language processing, speech recognition, and other application development posts are mainly concentrated in the $A3$, $B2$, and $C1$ topics. The algorithm research post is concentrated under the $B5$ and $C4$ topics, mainly in two directions: machine learning and data mining. $C5$, $D1$, and $D2$ topics are classified as senior R&D posts, including the team leader, who leads the enterprise AI team; the senior algorithm researcher responsible for the efficient optimization of existing algorithms and the research and development of core intelligent algorithms and cooperating with the engineering team to realize the engineering implementation of algorithms; and artificial intelligence experts who overcome the technical difficulties in the research and development of core algorithms to maintain the leading position of algorithms and are responsible for the precipitation of scientific research achievements and product transformation. The artificial intelligence posts corresponding to the extracted topics are sorted, and the results are shown in Figure 5.

Among the five types of artificial intelligence posts, the basic application posts in the artificial intelligence industry are mainly responsible for the sales promotion of artificial intelligence products, customer relationship maintenance, and other auxiliary work. Talent is required to have the general basic skills and qualities necessary for all industries. Those in technical support posts need to master customer needs and project progress, complete project installation, commissioning, and training, identify problems in artificial intelligence projects and products and provide solutions, provide professional support for the industry, and have professional knowledge and technology such as data processing and network programming. Artificial intelligence application development posts require skills in search engines, natural language processing [49], speech recognition, embedded development, and computer vision [50]. Accordingly, talent is required to have deep knowledge in one or several directions. The algorithm research posts include two major directions: machine learning or deep learning and data mining. It is required for talent in these posts to master common algorithms and frameworks and to be able to form opinions on the R&D and design of algorithms. The advanced research and development positions of artificial intelligence require the team to carry out project development in the enterprise, plan the development of the AI team, protect the advanced nature of the algorithm, and at the same time be responsible for the enterprise's patent deployment and research output. The demand for talent is the lowest but these posts require the talent to have a deeper development experience and R&D capability.

5.2. Extraction of the Professional Ability Requirements of Artificial Intelligence Posts Combined with Word2Vec and k-Means. We accumulate the four types of text word segmentation results into a complete recruitment demand text dataset, set the "min_count" parameter of Word2Vec to 300, that is, vectorize the words after word segmentation with a frequency greater than 300, and finally, K-means cluster the vectorized dataset. The global silhouette coefficient is iteratively calculated as k takes different values, and a broken line diagram is drawn, as shown in Figure 6.

After the k value reaches 15, the results of word vectorization clustering combined with Word2Vec and K-means, merging synonyms and eliminating irrelevant words, are shown in Table 2.

The clustering results of vocabulary show that most AI enterprises require talent to have a certain computer level, have the ability to quickly read materials in English, communicate well, actively cooperate, and constantly learn and innovate, while bearing the pressure of work. There is also a relatively concentrated professional demand for talent. The posts correspond to different educational levels. At a deeper level, talent is required to have a certain number of years of work experience and project development experience and to produce high-quality academic achievements. On the premise of mastering the basic knowledge of algorithms, theory, and data processing, it is also necessary to flexibly use common programming languages such as C++, C, Python, and Java to be familiar with different operating system development environments. Enterprises also require talent to have skills in product design, software, and hardware testing. Data mining and analysis, common algorithms, and deep learning frameworks are also high-demand professional skills and knowledge. Computer vision and image processing are currently the hot directions for development posts. In addition, there are increasing requirements for digital signal processing and simulation, industrial robot technology, etc.

5.3. Selection of Indicators. To objectively evaluate the professional ability of artificial intelligence talent, we must select the professional ability evaluation indices from many directions and angles and establish the hierarchical structure of multi-index evaluation. There is a certain mismatch between the demand for artificial intelligence talent in enterprises and artificial intelligence teaching in colleges and universities. More precisely, enterprises prefer professional

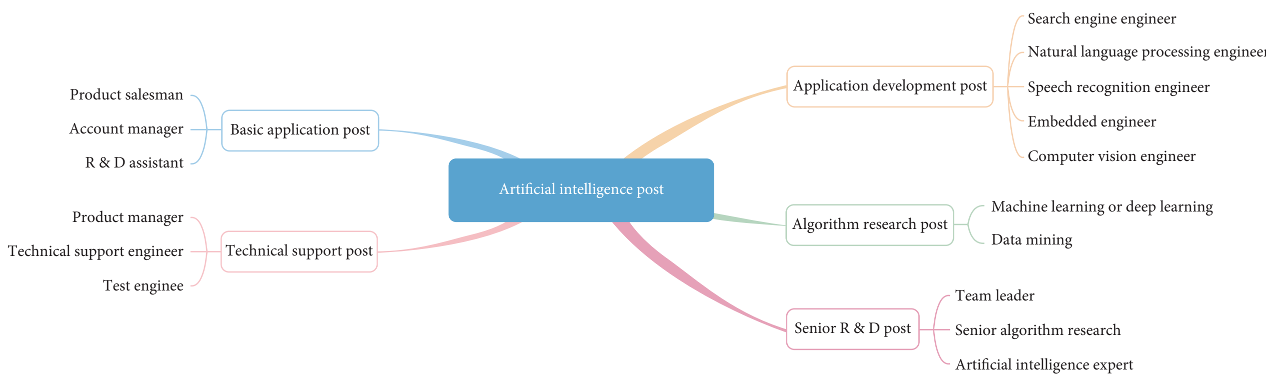


Figure 5: Artificial intelligence job classifications.

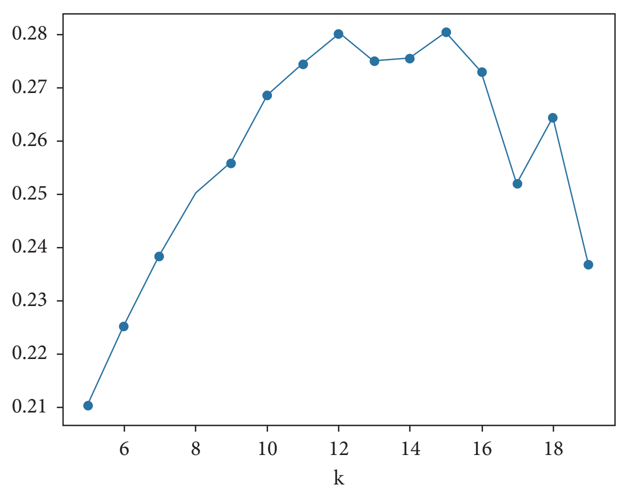


Figure 6: Variation trend of global silhouette coefficient.

skills and experience, while colleges and universities prefer knowledge and quality training. Enterprise recruitment text analysis summarizes the ability needs from the perspective of "industry." Therefore, this paper integrates the perspective of "education" through expert consultations, combined with literature research on talent training in colleges and universities. After this, we finally build an index system along four dimensions: basic quality, data processing, network programming skills, algorithm and design skills, and research and practice skills. Based on expert opinions, the selected indicators are added, deleted, and adjusted, and 32 second-level indicators, as shown in Figure 7, are preliminarily selected.

The Delphi method is then used to determine the final evaluation index system. Twenty-six experts from artificial intelligence enterprise technology R&D personnel, management personnel, and college artificial intelligence teachers were invited as team members, and an expert correspondence questionnaire was designed. The questionnaire included the experts' basic information, the experts' familiarity with the indicators, the Likert scale with the increasing importance of each indicator from 1 to 5 points, and the basis for judging the importance. The experts' familiarity with each index and the judgment basis of expert importance in the questionnaire are quantified to form the expert authority. The calculation formula of authority C_r is

$$C_r = \frac{C_a + C_s}{2}, \tag{5}$$

where C_a represents the basis for expert importance judgment, and the assignment from high to low is practical experience, relevant research, theoretical analysis, and intuitive judgment. C_s represents the arithmetic mean of the value of experts' familiarity with each index. The average authority C_r of the expert object in the first round of correspondence was 3.46, indicating that the selected experts are authoritative. The critical value method was adopted for the screening of indicators, and the indicators with an arithmetic average value of importance lower than 3.8, a full score ratio lower than 55%, and a coefficient of variation higher than 0.3 in the results of the first round of correspondence were deleted. To ensure preciseness, a large sample size questionnaire survey was conducted on the remaining indicators before the second round of Delphi. Taking the managers and technical R&D personnel of enterprises whose business scope involves artificial intelligence as well as the teachers and students of artificial intelligence-related majors in colleges and universities as the objects, 160 questionnaires

Table 2: Demand topics extracted by vocabulary vectorization clustering.

Category	Vocabulary	Refined demand
N1	Computer Automation Electronic engineering Math Software engineering Signal communication Applied mathematics	Professional background
	Undergraduate Master Doctor	Educational level
	More than 1 year More than 2 years More than 3 years More than 5 years Work experience Relevant working experience	Work experience
N2	Algorithm principle Theoretical basis Basic skills Modelling Algorithm development MATLAB Data processing	Professional basic knowledge
N3	Teamwork spirit Teamwork ability Steadfast Responsibility Communication skills Learning ability Innovative consciousness Compressive capacity Problem-solving ability Logical thinking ability Expressive ability	Basic quality
N4	SLAM Robot Motion Control Sensor ROS Plan Navigation	Industrial robot technology
N5	Open-source framework TensorFlow PyTorch Deep learning framework Caffe	Common open-source frameworks
N6	Statistics Data Big data Data mining Data analysis	Data mining and data analysis

Table 2: Continued.

Category	Vocabulary	Refined demand
N7	Debugging Software development Test Hardware equipment	Software testing and hardware debugging
	Design Technological process Product	Product design
N8	Code Network Database Development experience Embedded Development	Development experience
	Operating system Linux Environment	Mainstream operating system development environment
N9	C C++ Python Java Programming language Data structure Programming ability	Programing language
N10	Image processing 3D OpenCV Halcon Machine vision	Computer vision development software and technology
N11	Technology Research and development Artificial intelligence Field Project experience	Research and project experience in the field
N12	Deep learning Machine learning Natural language processing Neural network Recommendation algorithm	Artificial intelligence related algorithms
N13	Object detection Identification tracking Image Video Division	Computer vision industrial automation system
N14	Signal simulation Digital signal processing Radar	Digital signal processing and simulation
N15	English File Fast reading	Foreign language ability
	Paper publication	Academic achievements

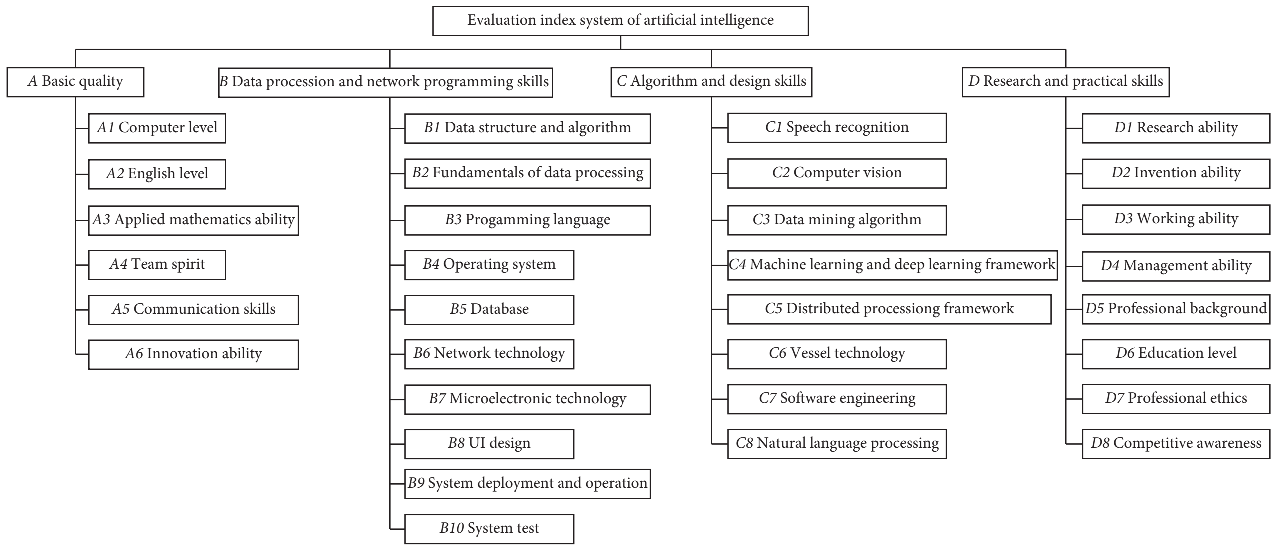


Figure 7: Construction of the preliminary evaluation index system of artificial intelligence professional ability.

were distributed, and 152 valid questionnaires were left after eliminating the questionnaires that were completed too carelessly, with an effective recovery rate of 95%. The Cronbach's α coefficient in the reliability test result of the questionnaire was 0.964, which is greater than 0.8; that is, the reliability of the questionnaire is sufficiently high. The *KMO* value of the validity test was 0.881, greater than 0.8, the chi-square of Bartlett's test of sphericity was 4867.51, and the corresponding p value was 0, less than 0.05, indicating that the data are suitable for factor analysis. The dimensionality reduction results of factor analysis are shown in Table 3, and the accumulative contribution rate is 70.813, indicating that most of the information on the variables is well explained.

According to the factor analysis results of second-level indicators $B1$, $B2$, and $B3$, it is suggested to classify them into the category of first-level indicator A. After adjusting the membership of indicators, we modify the names of first-level indicator A and first-level indicator B as "*A* Basic artificial intelligence vocational skills" and "*B* Database and network skills." In the second round of the Delphi process, 21 experts, whose authority value was greater than 3 in the first round were selected to form a new group, and the overall authority value increased to 3.95. The importance scoring results obtained in the first round of correspondence were attached to the questionnaire to achieve the effect that the opinions of experts were consistent. After the second round of Delphi expert consultation, the value range of the variation coefficient of each index changed from 0.135 to 0.324 in the first round to 0.092~0.220. The maximum value of the coefficient of variation was less than 0.25, and the index coordination coefficient increased from 0.129 to 0.346. This indicated a high degree of coordination of expert opinions. Therefore, the Delphi consultation was terminated. We established the final artificial intelligence professional ability evaluation index system and the connotation of each index, as shown in Table 4.

Table 3: Factor analysis results of the artificial intelligence professional ability evaluation index.

Code	Extracted principal components		Code	Extracted principal components	
	1	2		3	4
A1	0.678		C1	0.550	
A2	0.536		C2	0.714	
A3	0.696		C3	0.536	
A4	0.657		C4	0.762	
A5	0.623		C5	0.736	
A6	0.571		C6	0.732	
B1	0.738		C7	0.726	
B2	0.717		C8	0.700	
B3	0.691		D1		0.669
B4		0.708	D2		0.686
B5		0.552	D3		0.774
B6		0.783	D4		0.684
B7		0.808	D5		0.745
B8		0.750	D6		0.694
			D7		0.701
			D8		0.664

6. Determination of Weight

6.1. Constructing the Intuitionistic Fuzzy Judgement Matrix. We make a pairwise comparison of the relative importance of the four first-level indicators of the constructed index system and use the same method to make a pairwise comparison of the second-level indicators under each first-level indicator. Based on this, we establish the intuitionistic fuzzy judgement matrix $R = (r_{ij})_{n\times n}$ according to the standards in Table 5. The symbols i and j represent the rows and columns of the judgement matrix, respectively. Where $r_{ij} = (\mu_{ij}, \nu_{ij})$ $(i, j = 1, 2, 3, \ldots, n)$, μ_{ij} indicates the degree to which the i-th indicator is more important than the j-th indicator, that is, the degree of membership. The symbol ν_{ij} indicates the degree to which the j-th index is more important than the i-th index, that is, the

Table 4: Evaluation index system and connotation of artificial intelligence professional ability.

First-level index	Second-level index	Standard and connotation of each index
A basic artificial intelligence vocational skills	$A1$ computer level	The ability to use computers to deal with general problems
	$A2$ English level	English listening, speaking, reading and writing ability, especially foreign literature reading and learning ability
	$A3$ applied mathematics ability	Ability to model and solve problems using mathematical methods
	$A4$ team spirit	Have the overall concept, participate in team cooperation and jointly complete the teamwork objectives
	$A5$ communication skills	Be able to communicate effectively with internal and cross teams, respond to the needs of other types of post problems, and realize the business landing of artificial intelligence application scenarios
	$A6$ innovation ability	Can solve more complex application problems through reasonable combination, transformation, and innovation of relevant algorithm models
	$A7$ data structure and algorithm	Master basic algorithms such as recursion, sorting, and binary search, and use data structures flexibly
	$A8$ fundamentals of data processing	Have data processing ability, such as text, image, web page, and other data import, processing, transformation, etc.
	$A9$ programing language	Master C/C++, Python, java, and other programming languages
B database and network skills	$B1$ operating system	Master the development environment of mainstream operating systems such as MAC, Linux, and Windows
	$B2$ database	Master MySQL, Oracle, SQL Server, and other mainstream databases
	$B3$ network technology	Master the network configuration and application of switches and servers
	$B4$ electronic technology	Master the technology related to hardware design and development such as the artificial intelligence chip
	$B5$ UI design	Design and development of interactive interface of artificial intelligence equipment
C algorithm and design skills	$C1$ speech recognition	Master deep learning algorithms and models related to speech recognition
	$C2$ computer vision	Master computer vision-related problems and solutions, such as detection, tracking, classification, semantic segmentation, reinforcement learning, 3D vision and image processing, and master OpenCV, Halcon, VisionPro, and other machine vision development technologies
	$C3$ data mining algorithm	Master the principles of common data mining algorithms such as logistic regression and decision tree, and can apply them to practical scenarios
	$C4$ machine learning and deep learning framework	Master machine learning algorithms and mainstream deep learning frameworks such as Caffe, TensorFlow, and PyTorch
	$C5$ distributed processing framework	Master common distributed open-source frameworks such as Hadoop, Spark, and Storm
	$C6$ Vessel technology	Master Docker, K8S, Mesos, and other container technologies
	$C7$ software engineering	Master software design tools and be able to manage software engineering
	$C8$ natural language processing	Master deep learning algorithms and models related to natural language processing
D research and practical skills	$D1$ research ability	Publish artificial intelligence-related papers in core journals or international conferences
	$D2$ invention ability	Patent achievements related to artificial intelligence as the main inventor
	$D3$ Working ability	Working years and positions related to artificial intelligence
	$D4$ management ability	Experience in the development and implementation of AI products and projects as the main person in charge
	$D5$ professional background	Professional background requirements related to artificial intelligence
	$D6$ education level	Educational level requirements of artificial intelligence-related majors
	$D7$ professional ethics	Abide by laws and industrial ethics, and be vigilant against technical risks and privacy security
	$D8$ Competitive awareness	The ability to use computers to deal with general problems

nonmembership degree. π_{ij} represents the degree of hesitation, $\pi_{ij} = 1 - \mu_{ij} - \nu_{ij}$.

The scoring results of multiple experts are collected through the intuitionistic fuzzy weighted average operator, IFWA. A set of intuitionistic fuzzy numbers is set as $r_i = (\mu_i, \nu_i)$ $(i = 1, 2, \ldots)$, $IFWA{:}\Theta^n \longrightarrow \Theta$, if $IFWA_\omega(r_1, r_2, \ldots, r_n) = \omega_1 r_1 \oplus \omega_2 r_2 \oplus \ldots \oplus \omega_n r_n$. In addition, $\omega = (\omega_1, \omega_2, \ldots, \omega_n)^T$ is the weight vector of r_i $(i = 1, 2, \ldots, n)$, and its value is the normalized authority value of experts participating in intuitionistic fuzzy judgment research, that is, $\sum_{j=1}^{n} \omega_j = 1$. The operation of the intuitionistic fuzzy information set follows the following rules:

Table 5: Comparison between experts' preference and intuitionistic fuzzy numbers.

Experts' preference	Intuitionistic fuzzy numbers
Compared with the two indicators, i is extremely more important than j	(0.90, 0.10, 0.00)
Compared with the two indicators, i is strongly more important than j	(0.80, 0.15, 0.05)
Compared with the two indicators, i is obviously more important than j	(0.70, 0.20, 0.10)
Compared with the two indicators, i is moderately more important than j	(0.60, 0.25, 0.15)
Compared with the two indicators, i and j are equally important	(0.50, 0.30, 0.20)
Compared with the two indicators, j is moderately more important than i	(0.40, 0.45, 0.15)
Compared with the two indicators, j is obviously more important than i	(0.30, 0.60, 0.10)
Compared with the two indicators, j is strongly more important than i	(0.20, 0.75, 0.05)
Compared with the two indicators, j is extremely more important than i	(0.10, 0.90, 0.00)

Table 6: Intuitionistic fuzzy judgement matrix (taking first-level index as an example).

Judgement matrix	Intuitionistic fuzzy set			
	(0.5, 0.5)	(0.4310, 0.3820)	(0.5152, 0.4321)	(0.3768, 0.5102)
X	(0.3820, 0.4310)	(0.5, 0.5)	(0.4166, 0.4553)	(0.4830, 0.4040)
	(0.4321, 0.5152)	(0.4553, 0.4166)	(0.5, 0.5)	(0.4633, 0.3952)
	(0.5102, 0.3768)	(0.4040, 0.4830)	(0.3952, 0.4633)	(0.5, 0.5)

$$\alpha_1 \oplus \alpha_2 = \left(\mu_{\alpha_1} + \mu_{\alpha_2} - \mu_{\alpha_1} * \mu_{\alpha_2}, \nu_{\alpha_1} * \nu_{\alpha_2}\right), \tag{6}$$

$$\alpha_1 \otimes \alpha_2 = \left(\mu_{\alpha_1} * \mu_{\alpha_2}, \nu_{\alpha_1} + \nu_{\alpha_2} - \nu_{\alpha_1} * \nu_{\alpha_2}\right), \tag{7}$$

$$\lambda\alpha_1 = \left(1 - \left(1 - \mu_1\right)^{\lambda}, \nu_1^{\lambda}\right), \tag{8}$$

$$\alpha_1^{\lambda} = \left(\mu_1^{\lambda}, 1 - \left(1 - \nu_1\right)^{\lambda}\right). \tag{9}$$

This research used the senior management of artificial intelligence companies established for more than 10 years, professors and associate professors of artificial intelligence-related majors in higher vocational colleges and universities directly under the Ministry of Education, and a total of 5 experts. They scored the relative importance of indicators at all levels of the constructed system. The questionnaire also required experts to judge their familiarity with each index and indicate the judgment basis of relative importance to quantify the authority of experts. The authority value C_r is normalized to obtain the expert weight vector $\omega = (0.19, 0.19, 0.22, 0.21,$ and $0.19)$ T. The intuitionistic fuzzy weighted average operator $\mathrm{IFWA}_{\omega}(r_1, r_2, \ldots, r_6) = 0.19r_1 \oplus 0.19r_2 \oplus\ 0.22r_3 \oplus 0.21r_4 \oplus 0.19r_5$. Combined with Formulas (6) and (8), the intuitionistic fuzzy judgment matrix of indices at all levels is obtained. The relative importance of the first-level index to the total index is taken as an example, and the intuitionistic fuzzy judgment matrix X is shown in Table 6.

6.2. Consistency Check

6.2. Consistency Check. The consistency check can be used to establish whether there is conflict in the evaluation of the relative importance of indicators by expert;ls. If the consistency check fails, formula iteration can be carried out by setting parameters without the second scoring by experts. The formula of the consistency check is [51]:

$$d(\overline{R}, R) = \frac{1}{2(n-1)(n-2)} \cdot \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\left|\overline{\mu}_{ij} - \mu_{ij}\right| + \left|\overline{\nu}_{ij} - \nu_{ij}\right| + \left|\overline{\pi}_{ij} - \pi_{ij}\right|\right), \tag{10}$$

where R represents the intuitionistic fuzzy judgement matrix $R = (r_{ij})_{n\times n}$, $\overline{R}$ represents the intuitionistic fuzzy consistency judgement matrix $\overline{R} = (\overline{r}_{ij})_{n\times n}$, and the calculation formula of $\overline{R}$ is [50]:

(1) When $j > i + 1$, let $\overline{r}_{ij} = (\overline{\mu}_{\mathrm{ij}}, \overline{\nu}_{ij})$, where:

$$\overline{\mu}_{ij} = \frac{\sqrt[j-i-1]{\prod_{t=i+1}^{j-1}\mu_{it}\mu_{tj}}}{\sqrt[j-i-1]{\prod_{t=i+1}^{j-1}\mu_{it}\mu_{tj}} + \sqrt[j-i-1]{\prod_{t=i+1}^{j-1}\left(1-\mu_{it}\right)\left(1-\mu_{tj}\right)}}, j > i + 1,$$
$$\overline{\nu}_{ij} = \frac{\sqrt[j-i-1]{\prod_{t=i+1}^{j-1}\nu_{it}\nu_{tj}}}{\sqrt[j-i-1]{\prod_{t=i+1}^{j-1}\nu_{it}\nu_{tj}} + \sqrt[j-i-1]{\prod_{t=i+1}^{j-1}\left(1-\nu_{it}\right)\left(1-\nu_{tj}\right)}}, j > i + 1. \tag{11}$$

(2) When $j = i + 1$, let $\overline{r}_{ij} = (\mu_{ij}, \nu_{ij})$,

(3) When $j < i$, let $\overline{r}_{ij} = (\overline{\nu}_{ji}, \overline{\mu}_{ji})$.

We substitute the calculated intuitionistic fuzzy consistency judgment matrix $\overline{R}$ into (10) for consistency check calculation. If the calculated $d(R, R) < 0.1$, the consistency check passes. If $\overline{R}$ fails to pass the consistency check, the iterative parameter σ is introduced. We then adjust the value of the iterative parameters and calculate and correct the intuitionistic fuzzy consistency judgment matrix according to Formula [50]:

$$\tilde{\mu}_{ij} = \frac{(\mu_{ij})^{1-\sigma}(\overline{\mu}_{ij})^{\sigma}}{(\mu_{ij})^{1-\sigma}(\overline{\mu}_{ij})^{\sigma} + (1-\mu_{ij})^{1-\sigma}(1-\overline{\mu}_{ij})^{\sigma}}, \quad i, j = 1, 2 \ldots n, \tag{12}$$

$$\tilde{\nu}_{ij} = \frac{(\nu_{ij})^{1-\sigma}(\overline{\nu}_{ij})^{\sigma}}{(\nu_{ij})^{1-\sigma}(\overline{\nu}_{ij})^{\sigma} + (1-\nu_{ij})^{1-\sigma}(1-\overline{\nu}_{ij})^{\sigma}} i, \quad j = 1, 2 \ldots n. \tag{13}$$

The changed intuitionistic fuzzy consistency judgment matrix $\tilde{R} = (\tilde{r}_{ij})_{n\times n}$, where $\tilde{r}_{ij} = (\tilde{\mu}_{ij}, \tilde{\nu}_{ij})$, is substituted into Formula (14) for the calculation check until the consistency check is passed.

$$d(\bar{R}, \tilde{R}) = \frac{1}{2(n-1)(n-2)} \cdot \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\left|\tilde{\mu}_{ij} - \bar{\mu}_{ij}\right| + \left|\tilde{\nu}_{ij} - \bar{\nu}_{ij}\right| + \left|\tilde{\pi}_{ij} - \bar{\pi}_{ij}\right|\right). \tag{14}$$

The consistency of five intuitionistic fuzzy judgment matrices is checked according to Formula (10). For the judgment matrix R with $d(\overline{R}, R) > 0.1$, the iterative parameter $\sigma_A \in (0, 1)$ is introduced, taking 0.1 as the starting point and a step size of 0.1. According to the iterative test of Formulas (13) to (15), the adjusted value is calculated until $d(\bar{R}, \tilde{R}) < 0.1$. The results of the parameter adjustment and consistency check are shown in Table 7.

Taking the relative importance of the first-level index to the total index as an example, the adjusted and corrected intuitionistic fuzzy judgment matrix $\tilde{X}$ is shown in Table 8.

Table 7: Parameter adjustment and consistency check results.

Judgement matrix	$d(\overline{R}, R)$	Adjust parameters σ	$d(\overline{R}, \tilde{R})$
X	0.2012	0.5	0.098
A	0.1818	0.5	0.0849
B	0.1394	0.3	0.0929
C	0.1538	0.4	0.0879
D	0.1390	0.3	0.0891

Table 8: Intuitionistic fuzzy judgement matrix after adjustment and correction (taking first-level index as an example).

Judgement matrix	Intuitionistic fuzzy set			
$\tilde{X}$	(0.5, 0.5)	(0.4310, 0.3820)	(0.4310, 0.3854)	(0.4111, 0.4081)
	(0.3820, 0.4310)	(0.5, 0.5)	(0.4166, 0.4553)	(0.4315, 0.3783)
	(0.3854, 0.4312)	(0.4553, 0.4166)	(0.5, 0.5)	(0.4633, 0.3952)
	(0.4081, 0.4111)	(0.3783, 0.4315)	(0.3952, 0.4633)	(0.5, 0.5)

6.3. Weight Calculation. The index weight value of each level is calculated by the modified intuitionistic fuzzy consistency judgment matrix that has passed the consistency check, and the calculation formula is [50]:

$$\omega_i = \left(\frac{\sum_{j=1}^{n}\mu_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{n}(1-\nu_{ij})}, 1 - \frac{\sum_{j=1}^{n}(1-\nu_{ij})}{\sum_{i=1}^{n}\sum_{j=1}^{n}\mu_{ij}}\right) i, \quad j = 1, 2 \ldots n. \tag{15}$$

The calculated weights are two-dimensional. To meet the more intuitive evaluation requirements, the two-dimensional index weights need to be fuzzily transformed from a vague set to a fuzzy set [52].

Let U be a universe, A be a vague set in U, and $A = \{<t_A(x), f_A(x), \pi_A(x) > |x \in U\} \in V(U)$. Then, $A^{(n)} = \{<t_A^{(n)}(x), f_A^{(n)}(x), \pi_A^{(n)}(x) > |x \in X\}$ is called the n-th transformation from A to the fuzzy set. If $t_A(x)$ and $f_A(x)$ are not all 0, then when $n \longrightarrow \infty$, the limit state of A transformation to a fuzzy set is:

$$A^{(\infty)} = \left\{ < t_A^{(\infty)}(x), f_A^{(\infty)}(x), \pi_A^{(\infty)}(x) > | x \in X \right\}, \tag{16}$$

$$t_A^{(\infty)}(x) = t_A(x)\prod_{k=1}^{\infty}(1+\pi_A(x))^{2^{k-1}} = \frac{t_A(x)}{1-\pi_A(x)} = \frac{t_A(x)}{t_A(x)+f_A(x)}, \tag{17}$$

$$f_A^{(\infty)}(x) = f_A(x)\prod_{k=1}^{\infty}(1+\pi_A(x))^{2^{k-1}} = \frac{f_A(x)}{1-\pi_A(x)} = \frac{f_A(x)}{t_A(x)+f_A(x)}, \quad \pi_A^{(\infty)}(x) = 0. \tag{18}$$

At this time, $t_A^{(\infty)}(x) + f_A^{(\infty)}(x) = 1$, and $A^{(\infty)}$ is a fuzzy set. Then, the transformed fuzzy set of vague set A is $A^F(x) = \{<x, A^F(x) > x \in U\} = t_A^{(\infty)}(x)$. We normalize all $A^F(x)$ at the same level to obtain the final weight of indicators at each level.

After the five modified intuitionistic fuzzy matrices passing the consistency check are calculated by Formula (15) to aggregate the weights, the fuzzy set is transformed by Formula (17), the two-dimensional weights are mapped to one-dimensional weights, and the obtained $A^F(x)$ is normalized to obtain the one-dimensional comprehensive weights of indicators at all levels. The weight calculation results are shown in Table 9.

6.4. Analysis of the Results. From the weight assignment results, it can be seen that the weights of the four first-level indicators are almost the same. However, the weight of “C algorithm and design skills” is 0.2564, which is relatively

Table 9: Weight of artificial intelligence professional ability evaluation index.

First-level index	Second-level index	Final weight of the second-level index
A (0.1968, 0.6674) 0.2559	*A*1 (0.1012, 0.8164) 0.1431	0.0366
	*A*2 (0.0734, 0.8554) 0.1025	0.0262
	*A*3 (0.1028, 0.8158) 0.1452	0.0372
	*A*4 (0.0689, 0.8566) 0.0966	0.0247
	*A*5 (0.0685, 0.8565) 0.0961	0.0246
	*A*6 (0.0760, 0.8455) 0.1070	0.0274
	*A*7 (0.0825, 0.8356) 0.1166	0.0298
	*A*8 (0.0657, 0.8618) 0.0919	0.0235
	*A*9 (0.0716, 0.8489) 0.1009	0.0258
B (0.1920, 0.6801) 0.2474	*B*1 (0.1788, 0.7155) 0.2319	0.0574
	*B*2 (0.2010, 0.6886) 0.2621	0.0648
	*B*3 (0.1395, 0.7561) 0.1807	0.0447
	*B*4 (0.1350, 0.7611) 0.1747	0.0432
	*B*5 (0.1166, 0.7816) 0.1506	0.0373
C (0.2002, 0.6771) 0.2564	*C*1 (0.0952, 0.8132) 0.1332	0.0342
	*C*2 (0.0925, 0.8211) 0.1287	0.0330
	*C*3 (0.1158, 0.7940) 0.1618	0.0415
	*C*4 (0.1164, 0.7916) 0.1630	0.0418
	*C*5 (0.0925, 0.8242) 0.1283	0.0329
	*C*6 (0.0733, 0.8522) 0.1007	0.0258
	*C*7 (0.0632, 0.8653) 0.0865	0.0222
	*C*8 (0.0708, 0.8487) 0.0979	0.0251
D (0.1866, 0.6860) 0.2403	*D*1 (0.1440, 0.7793) 0.1898	0.0456
	*D*2 (0.1275, 0.7996) 0.1674	0.0402
	*D*3 (0.0971, 0.8313) 0.1272	0.0306
	*D*4 (0.0941, 0.8358) 0.1232	0.0296
	*D*5 (0.0898, 0.8426) 0.1172	0.0281
	*D*6 (0.0901, 0.8403) 0.1179	0.0283
	*D*7 (0.0675, 0.8728) 0.0874	0.0210
	*D*8 (0.0542, 0.8895) 0.0699	0.0168

high. This shows that as one of the three cornerstones of artificial intelligence, the mastery of algorithms is an important standard for investigating the professional ability of artificial intelligence talent, and is an important link between the college training plan and industry talent development. The second is "*A* Basic artificial intelligence vocational skills," with a weight of 0.2559. The basic vocational skills of artificial intelligence are the basic requirements for engaging in artificial intelligence-related work, which is composed of general basic skills and basic data processing and programming skills of front-line employees in various industries. "*B* Database and network skills" is the third place, with a weight of 0.2474. Database and network skills are professional skills and knowledge highly related to artificial intelligence. Databases store massive data in computers to complete the management and sharing of data, while computer networks can establish connections between decentralized independent computers to achieve the purpose of information resource sharing. As a high-tech industry, artificial intelligence attracts a considerable number of young practitioners. Therefore, the requirements for practical experience are not strict, and the professional background is highly inclusive. The weight value of "*D* Research and practical skills" is thus lower, which is 0.2403.

Under the first-level indicator "*A* Basic artificial intelligence vocational skills," the weight of applied mathematics ability and computer level far exceeds that of other second-level indicators, followed by the data structure, algorithm, and innovation ability. Under "*B* Database and network skills," mastering the development environment of the mainstream database and the mainstream operating system is also the most important indicator. The high professional threshold of the artificial intelligence industry makes the evaluation of the basic quality and skills of talent focus on the professional skills and knowledge of the industry and requires innovation ability on this basis. The three cornerstones of artificial intelligence are algorithms, data, and computing power. Data mining, machine learning, and deep learning algorithms are the fundamental ways to realize artificial intelligence, simulate human learning behavior, and endow computers with intelligence. Therefore, in the first-level index "*C* Algorithm and design skills," mastering the data mining algorithms, machine learning, and deep learning frameworks have a great impact on the evaluation results of professional ability, while speech recognition and computer vision are the most popular practical application scenarios of artificial intelligence in recent years. These are not only the hot topics of algorithm research in colleges and universities but also the technical opportunity directions of industry development. They also have a high weight in the evaluation system. The second-level index weight data under "*D* Research and practical skills" shows that research and invention ability is also an important evaluation dimension, which reflects the employees' research depth and innovation consciousness.

7. Conclusion

Based on the mining and content analysis of the recruitment text of artificial intelligence posts on a recruitment website, this paper extracts five main types of artificial intelligence posts: basic application posts, technical support posts, application development posts, algorithm research posts, and senior R&D posts. Based on the work responsibilities and corresponding ability needs of various posts, combined with the talent training plan of colleges and universities obtained from expert consultation and literature research, a multidimensional evaluation index system of the professional ability of artificial intelligence talent is constructed. Then, the index weights at all levels are calculated according to the intuitive fuzzy judgment of experts under the two backgrounds of "industry" and "education." By analyzing the index weight calculation results of the intuitive fuzzy analytic hierarchy process in Section 6.4, it is found that the weight of the "C Algorithm and design skills" is 0.2564, which is the highest among the four first-level indicators. This shows that the industry and education expert groups surveyed in this study agree that the algorithms and design skills from the perspective of the integration of industry and education are the most important evaluation dimension of the professional ability of artificial intelligence talent. Talent is also required to master data mining and machine learning and deep learning algorithms and have the ability to solve practical problems of computer vision and speech recognition. On the one hand, in terms of the need for learning and innovation in colleges and universities, the mastery and research of algorithms is the cornerstone of the continuous updating of AI's cutting-edge technology. On the other hand, the design skills of AI talent are an important driving force for the implementation of cutting-edge research on algorithms and the continuous development of the AI industry. In addition, the basic abilities of the evaluated object are of great concern. Among the basic abilities, professional skills such as applied mathematics, computers, data structures, and algorithms are more important, and the basic quality is mainly innovation ability. The weight of the database and network skill dimension ranks third, and talent is mainly required to be familiar with the mainstream database and mainstream operating system development environment. The last is the dimension of research and practical skills. The results of papers published in core journals or international conferences and artificial intelligence patents are all bonus items of professional ability evaluation.

Human-job match theory points out that human personality differences are common, and any individual has his or her own characteristics. The fundamental problem is how to put talent in appropriate positions to give full play to their maximum effectiveness. The final analysis is to overcome the problem of information asymmetry between positions and the workforce. The integration of industry and education is an effective way to solve the serious problem of talent training and demand mismatch. In the process of talent training, the cultivation method of industry and education integration takes projects as the link and deepens the cooperation between schools and enterprises, which is an effective way to cultivate high-level applied talent. In addition to artificial intelligence posts, the professional ability evaluation of various industries in urgent need of applied talent is applicable to the perspective of integration of industry and education, especially in industries that have high requirements for the technical level and innovation consciousness of employees. This paper studies the vocational ability evaluation system based on artificial intelligence from the perspective of the integration of production and education, which has a certain reference significance for the vocational ability evaluation of applied talent in other industries in the future.

Based on the abovementioned results, we believe that strengthening the integration of industry and education and promoting the training of artificial intelligence talent is still an effective means to comply with the rapid development of artificial intelligence skills and alleviate the contradiction between talent supply and industry demand. The selection of indicators in this study is based on the demanded text of artificial intelligence talent, combined with the actual business scenarios of the industry and theoretical research in the literature. In the process of index establishment and weight calculation, the cognition of experts from both "industry" and "education" was consulted and integrated. The integration of these factors could lead to a more multidimensional and unified evaluation system and a beneficial supplement to ease the lack of talent skill certification in today's artificial intelligence industry. However, the evaluation index system is only one way to test the professional talent ability. If the talent training of the artificial intelligence industry wants to achieve the "integration of industry and education" and shorten the distance between higher education and the actual needs of industry development, it is more important to pay attention to the mutual penetration and support of the roles of universities and industry. In this context, one could consider a cooperative teaching mode between schools and enterprises, encourage innovative research based on enterprise R&D projects, and provide more practical opportunities in talent education. One could also explore the training mode of "artificial intelligence +" compound talent and implement the training strategy of professional differentiation to promote an effective connection between the paths of talent education and industry development.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the special key project "research on vocational skill level evaluation system of artificial intelligence integrated with industry and education" of the Modern Educational Technology Research Smart Campus in Jiangsu Province (2020-R-84366).

References

[1] J. Evans, "Social computing unhinged," *Journal of Social Computing*, vol. 1, no. 1, pp. 1–13, 2020.

[2] Y. S. Su, Y. Ruan, S. Sun, and Y. T. Chang, "A pattern recognition framework for detecting changes in Chinese internet management system," *Journal of Social Computing*, vol. 1, no. 1, pp. 28–39, 2020.

[3] X. Gao, J. D. Luo, K. Yang, X. Fu, L. Liu, and W. Gu, "Predicting tie strength of Chinese guanxi by using big data of social networks," *Journal of Social Computing*, vol. 1, no. 1, pp. 40–52, 2020.

[4] Y. Chen, W. Gu, and K. Li, "Dynamic task offloading for internet of things in mobile edge computing via deep reinforcement learning," *International Journal of Communication Systems*, 2022.

[5] Y. Xiao, G. Niu, L. Xiao, Y. Ding, S. Liu, and Y. Fan, "Reinforcement learning based energy-efficient internet-of-things video transmission," *Intelligent and Converged Networks*, vol. 1, no. 3, pp. 258–270, 2020.

[6] T. Li, C. Li, J. Luo, and L. Song, "Wireless recommendations for internet of vehicles: recent advances, challenges, and opportunities," *Intelligent and Converged Networks*, vol. 1, no. 1, pp. 1–17, 2020.

[7] X. Zhou, W. Liang, K. I. K. Wang, and L. T. Yang, "Deep correlation mining based on hierarchical hybrid networks for heterogeneous big data recommendations," *IEEE*

Transactions on Computational Social Systems, vol. 8, no. 1, pp. 171–178, Feb. 2021.

[8] R. Gu, K. Zhang, Z. Xu et al., “Fluid: dataset abstraction and elastic acceleration for cloud-native deep learning training jobs,” in *Proceedings of the 38th IEEE International Conference on Data Engineering*, pp. 2183–2196, Kuala Lumpur, Malaysia, May 2022.

[9] W. Zhang, X. Chen, and J. Jiang, “A multi-objective optimization method of initial virtual machine fault-tolerant placement for star topological data centers of cloud systems,” *Tsinghua Science and Technology*, vol. 26, no. 1, pp. 95–111, 2021.

[10] J. Huang, Z. Tong, and Z. Feng, “Geographical poi recommendation for internet of things: a federated learning approach using matrix factorization,” *International Journal of Communication Systems*, 2022.

[11] Y. Chen, Z. Liu, Y. Zhang, Y. Wu, X. Chen, and L. Zhao, “Deep reinforcement learning-based dynamic resource management for mobile edge computing in industrial internet of things,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 4925–4934, 2021.

[12] A. K. Sandhu, “Big data with cloud computing: discussions and challenges,” *Big Data Mining and Analytics*, vol. 5, no. 1, pp. 32–40.

[13] J. Huang, B. Lv, Y. Wu, Y. Chen, and X. Shen, “Dynamic admission control and resource allocation for mobile edge computing enabled small cell network,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 2, pp. 1964–1973, 2022.

[14] C. Ying, C. Xu, and Z. Zhi, “Dynamic task offloading for mobile edge computing with hybrid energy supply,” *Tsinghua Science and Technology*, .

[15] Y. Bie and Y. Yang, “A multitask multiview neural network for end-to-end aspect-based sentiment analysis,” *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 195–207, 2021.

[16] G. Zhai, Y. Yang, H. Wang, and S. Du, “Multi-attention fusion modeling for sentiment analysis of educational big data,” *Big Data Mining and Analytics*, vol. 3, no. 4, pp. 311–319, 2020.

[17] Q. Wang, X. Yang, D. Pu, and Y. Fan, “Sustainable investment forecasting of power grids based on the deep restricted Boltzmann machine optimized by the lion algorithm,” *Computer Modeling in Engineering and Sciences*, vol. 130, no. 1, pp. 269–286, 2022.

[18] Q. Shao, R. Yu, H. Zhao et al., “Toward intelligent financial advisors for identifying potential clients: a multitask perspective,” *Big Data Mining and Analytics*, vol. 5, no. 1, pp. 64–78, 2022.

[19] J. Xu, D. Li, W. Gu, and Y. Chena, “Uav-assisted Task Offloading for Iot in Smart Buildings and Environment via Deep Reinforcement Learning,” *Building and Environment*, vol. 222, 2022.

[20] Y. Chen, F. Zhao, X. Chen, and Y. Wu, “Efficient multi-vehicle task offloading for mobile edge computing in 6g networks,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 4584–4595, 2022.

[21] R. Gu, Y. Chen, S. Liu et al., “Liquid: intelligent resource estimation and network-efficient scheduling for deep learning jobs on distributed GPU clusters,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, p. 1, 2021.

[22] Y. Lu, X. Chen, Y. Zhang, and Y. Chen, “Cost-efficient resources scheduling for mobile edge computing in ultra-dense networks,” *IEEE Transactions on Network and Service Management*, p. 1, 2022.

[23] G. M. S. Rahman, T. Dang, and M. Ahmed, “Deep reinforcement learning based computation offloading and resource allocation for low-latency fog radio access networks,” *Intelligent and Converged Networks*, vol. 1, no. 3, pp. 243–257, 2020.

[24] S. Nath and J. Wu, “Deep reinforcement learning for dynamic computation offloading and resource allocation in cache-assisted mobile edge computing systems,” *Intelligent and Converged Networks*, vol. 1, no. 2, pp. 181–198, 2020.

[25] L. Qi, C. Hu, X. Zhang et al., “Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4159–4167, 2021.

[26] C. Catlett, P. Beckman, N. Ferrier et al., “Measuring cities with software-defined sensors,” *Journal of Social Computing*, vol. 1, no. 1, pp. 14–27, 2020.

[27] X. Zhou, Y. Li, and W. Liang, “CNN-RNN based intelligent recommendation for online medical pre-diagnosis support,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 3, pp. 912–921, May-Jun. 2021.

[28] X. Guo, Y. D. Zhang, S. Lu, and Z. Lu, “A survey on machine learning in COVID-19 diagnosis,” *Computer Modeling in Engineering and Sciences*, vol. 130, no. 1, pp. 23–71, 2022.

[29] Y. Chen, H. Xing, Z. Ma, and X. Chen, “Cost-efficient edge caching for noma-enabled iot services,” China Communications, 2022.

[30] H. Dai, Y. Xu, G. Chen et al., “ROSE: robustly safe charging for wireless power transfer,” *IEEE Transactions on Mobile Computing*, vol. 21, no. 6, pp. 2180–2197, 2022.

[31] H. Dai, X. Wang, X. Lin et al., “Placing wireless chargers with limited mobility,” *IEEE Transactions on Mobile Computing*, p. 1, 2021.

[32] C. B. Frey and M. A. Osborne, “Osborne, “the future of employment: how susceptible are jobs to computerisation,” *Technological Forecasting and Social Change*, vol. 114, no. 1, pp. 254–280, 2017.

[33] D. Acemoglu and P. Restrepo, “Robots and jobs: evidence from US labor markets,” *Journal of Political Economy*, vol. 128, no. 6, pp. 2188–2244, 2020.

[34] T. Wang and Y. Ren, “Talent strategy in the age of artificial intelligence: interpretation of the ‘innovative action plan for artificial intelligence in colleges and universities,” *The Journal of Distance Education*, vol. 36, no. 5, pp. 52–59, 2018.

[35] Y. Li and X. Chen, “Talent profiles of artificial intelligence and its training countermeasures based on recruitment data,” *Chongqing Higher Education Research*, vol. 9, no. 5, pp. 55–68, 2021.

[36] H. Huang, “Understanding and thinking of talent training of artificial intelligence specialty under the background of new engineering,” *China University Teaching*, vol. 39, no. 02, pp. 20–25, 2019.

[37] C. Wang and S. She, “The heterogeneity of registered residence professional-education mismatching and income gaps,” *Comparative Economic & Social Systems*, vol. 33, no. 3, pp. 10–20, 2017.

[38] L. Chen, H. T. Lian, X. Y. Sun, and B. Liu, “Sensitive detection of L-5-hydroxytryptophan based on molecularly imprinted polymers with graphene amplification,” *Analytical Biochemistry*, vol. 526, no. 6, pp. 58–65, 2017.

[39] Y. Li, F. Yang, and F. Wang, “Systematic thinking on the trinity of ‘artificial intelligence + higher education,” *China Educational Technology*, vol. 42, no. 9, pp. 88–96, 2021.

[40] W. Dong and Y. Huang, “Vocational education reform and model construction under the background of artificial intelligence,” *China Educational Technology*, vol. 40, no. 7, pp. 1–7+45, 2019.

[41] M. Hu, “Exploration and practice of new engineering education mode integrating industry and education,” *China University Teaching*, vol. 41, no. 6, pp. 7–11, 2019.
[42] Z. Y. Ren and J. N. Liu, “Reflections on the universities’ talent cultivation reform in the era of artificial intelligence,” *University Education Science*, vol. 37, no. 4, pp. 114–121, 2019.
[43] M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of Machine Learning Research*, vol. 3, no. 1, pp. 993–1022, 2003.
[44] T. Mikolov, I. Sutskever, and K. Chen, “Distributed representations of words and phrases and their compositionality,” *Advances in Neural Information Processing Systems*, pp. 3111–3119, MIT Press, Cambridge, MA, USA, 2013.
[45] L. Jing, M. K. Ng, and J. Z. Huang, “Knowledge-based vector space model for text clustering,” *Knowledge and Information Systems*, vol. 25, no. 1, pp. 35–55, 2010.
[46] P. J. M. Van Laarhoven and W. Pedrycz, “A fuzzy extension of saaty’s priority theory,” *Fuzzy Sets and Systems*, vol. 11, no. 1-3, pp. 229–241, 1983.
[47] J. J. Buckley, “Fuzzy hierarchical analysis,” *Fuzzy Sets and Systems*, vol. 17, no. 3, pp. 233–247, 1985.
[48] K. T. Atanassov, “Intuitionistic fuzzy sets,” *Fuzzy Sets and Systems*, vol. 20, no. 1, pp. 87–96, 1986.
[49] C. Peng, C. Zhang, X. Xue, J. Gao, H. Liang, and Z. Niu, “Cross-modal complementary network with hierarchical fusion for multimodal sentiment classification,” *Tsinghua Science and Technology*, vol. 27, no. 4, pp. 664–679, 2022.
[50] Q. Cao, W. Zhang, and Y. Zhu, “Deep learning-based classification of the polar emotions of ‘moe’ -style cartoon pictures,” *Tsinghua Science and Technology*, vol. 26, no. 3, pp. 275–286, 2021.
[51] Z. Xu and H. Liao, “Intuitionistic fuzzy analytic hierarchy process,” *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 4, pp. 749–761, 2014.
[52] C. Y. Zhang and H. Y. Zhou, “Transformation and entropy for vague sets,” *Computer Engineering and Applications*, vol. 43, no. 33, pp. 20-21, 2006.

Hindawi
Scientific Programming
Volume 2022, Article ID 8626882, 14 pages
https://doi.org/10.1155/2022/8626882

Review Article

Challenges in Integration of Heterogeneous Internet of Things

Muhammad Noaman,[1] Muhammad Sohail Khan,[1] Muhammad Faisal Abrar,[2] Sikandar Ali,[3] Atif Alvi,[4] and Muhammad Asif Saleem[5]

[1]*Department of Computer Software Engineering, University of Engineering and Technology Mardan, Khyber Pakhtunkhwa, Mardan 23200, Pakistan*
[2]*Department of Computer Science, University of Engineering and Technology Mardan, Mardan 23200, Khyber Pakhtunkhwa, Pakistan*
[3]*Department of Information Technology, The University of Haripur, Haripur 22620, Khyber Pakhtunkhwa, Pakistan*
[4]*Department of Computer Science, University of Management and Technology, Lahore, Pakistan*
[5]*Department of Software Engineering, Lahore Garrison University, Lahore, Pakistan*

Correspondence should be addressed to Sikandar Ali; sikandar@uoh.edu.pk

Received 12 December 2021; Revised 11 May 2022; Accepted 28 June 2022; Published 16 August 2022

Academic Editor: Xu Zhang

Internet of Things (IoT) is considered the upcoming industrial and academic revolution in the technological world having billions of things and devices connected to the Internet. These connected devices are heterogeneous. They have different standards and technologies which communicate through different protocols. Therefore, the implementation of IoT on a large scale is difficult due to these heterogeneity challenges. This motivated us to overcome the scaling problem of IoT by identifying the challenges from the literature and providing solutions. This study is based on the identification of the heterogeneous challenges with solutions via a systematic literature review. A total of 81 primary sources were selected. After extracting and synthesizing the data, we identified 14 different IoT heterogeneity challenges. Some of the identified challenges are "heterogeneity of devices," "heterogeneity in formats of data," "heterogeneity in communication," and "interoperability issue due to heterogeneity." The identified challenges have been analyzed from digital libraries and timeframe perspectives. Furthermore, we have found a total of 81 solutions for those challenges, with at least 5 unique solutions for each challenge. In the future, we will categorize the challenges and prioritize the solutions by using a multi-criteria decision-making problem.

1. Motivation

IoT is the expansion of current Internet services to provide connectivity to each object of this world. IoT has become the most prominent technology across the globe. It is an emerging technology that is under development process where everyone is trying to interpret it according to their needs. The implementation and interpretation of IoT face some serious challenges like security, virtualization, and heterogeneity. Heterogeneity itself is a multifaceted challenge hindering the large-scale implementation of IoT vision. It is due to these challenges that so far only limited implementations of IoT systems have become a reality. This motivated us to perform a systematic literature review to identify those IoT heterogeneity challenges and their solutions. Another contribution of this study is conducting a depth analysis of those challenges using the chi-square test based on digital libraries and timeframe.

2. Introduction

In today's technological world, IoT is considered an important advancement among the trending technologies. The term IoT can be simply defined as the devices that can be connected with sources of the Internet [1]. In past years, these devices have been constantly growing. Reference [2] reported that around 500 billion devices will be connected to the Internet by 2030. In the physical as well as in the virtual world, these IoT devices will further interconnect with other devices in a large number, which will give new birth to the forms of interaction. This will enable us to connect all objects of our surroundings in every corner of the world in a single period. These objects can be sensors, smartphones, automobiles, industrial robots, refrigerators, thermostats, tablets, etc. The IoT is widespread in both academia and industry. It is producing business opportunities in multiple fields of industrial markets, in both public and private sectors in a very broad range. The industrial revolution of IoT will have billions of heterogeneous devices on the Internet of Things in the near future.

At the other extreme, the IoT vision of a large-scale implementation faces serious challenges across many dimensions. One of the main obstructions in IoT is its inclination toward heterogeneity, and the heterogeneous nature might be in form of protocols, device data format, communication capabilities of the devices, technologies, hardware, etc. [3, 4]. It is due to these types of challenges that so far only limited implementations of IoT systems have become a reality. For IoT to evolve toward its vision of global implementation, these obstructions need to be reduced on different levels. To activate and provide the service, devices must be connected to the Internet. The identification of heterogeneity-based challenges that exist at different levels is needed, and the current solutions adopted and/or implemented by different studies for handling heterogeneity in IoT systems need to be highlighted. The objective of this study is to conduct a systematic literature review to identify those IoT heterogeneity challenges and find out the solutions implemented by different studies to handle IoT heterogeneity challenges. The significance of this study is that it will provide the identification and analysis of heterogeneity challenges in IoT systems and provide a summary of different studies that implemented various solutions to handle the heterogeneity of IoT systems. Another significant contribution is that it will provide a future direction to researchers to make a better stand-alone architecture aiming to tackle the heterogeneity challenges at different levels of IoT systems. As a result, IoT systems can be utilized and implemented in a wide range of industrial fields.

The organization of this paper is as follows: Section 3 presents a literature review related to IoT history, challenges, and heterogeneity concerns. Section 4 explains the research methodology used to achieve the objectives of this research. Section 5 provides the result and discussion of IoT heterogeneity challenges and the solutions to those challenges found in this study. Section 6 concludes this review and suggests future work.

3. Literature Review

Because of the broad and complex nature of IoT, it has not yet got a single unique definition that is acceptable to the whole global community of users. Many researchers, practitioners, academicians, developers, and corporate people have defined IoT in their terms, but the credit must be attributed to Kevin Ashton, an expert on digital innovation who for the first time used and defined it. According to [5], IoT could be nicely defined as follows: a very comprehensive and accessible network of intelligent devices which can act and react in accord with situations; self-organize; share information, data, and resources; and be subject to change in the environment.

IoT is growing and maturing day by day. It is the latest, most fine, and excellent concept in information technology. It is a new paradigmatic shift in information technological advancement. The expression "Internet of Things," concisely shortened to IoT, is comprised of two words, "Internet" and "Things." Internet uses a standard set of Internet protocols (TCP/IP) to connect and serve a large number of users around the globe [6, 7]. It is a global system of interconnected computer networks. Internet is interconnected networks that includes a large number of local to regional commercial networks that may be private, government-owned, public, or academic networks. These networks are connected through a wide range of electronic, wireless, and optical network technologies [8]. The Internet is generally defined as a global network that connects millions of computers. About 190 states of the world are linked through the Internet that constantly shares data, opinions, and news [9]. According to [10], there is an estimated 5,080,388,142 Internet users around the world. This large sum of users indicates that about 40% of the world's total population uses the Internet.

The word "Things" in the "Internet of Things" can be any object or person identifiable in the real world. Daily necessities include electronic devices that we come across as well as daily used advanced technological items like equipment and gadgets. In the near future, other certain everyday objects are expected to connect with the Internet, which will lead to a period of extreme expansion of the Internet known as the Internet of Things. IoT system is based on devices to sense, actuate, control, and monitor activities [11]. IoT devices that are connected to other devices and applications can exchange data with each other, they may receive information from some other IoT devices. To process data, they may send the data either to centralized serves locally or it may send data for processing to cloud.

The National Intelligence Council (NIC) of the United Nations (UN) has considered IoT as one of the six "Disruptive Civil Technologies" [12]. In this respect, we can list many fields that are already benefiting from the services of different architectural forms of IoT like transportation, e-governance, smart city, smart health, life support, education, retail, logistics, agriculture, automation, manufacturing of industrial products, and management of businesses.

Ericsson [13] and Evans [14] have conducted surveys and estimated that the use of the Internet will further increase, grow, and be boosted tenfold in the coming days. According to their estimation, about 50 billion devices would have connected by 2020. This expected number of new Internet devices shall be supposedly called constricted devices [15]. These devices are small in size, are enclosed in nature, and have a low cost. They are specifically designed for the purpose of executing specific tasks like monitoring the physical environment. IoT devices are limited in terms of communication capabilities, processing power, and energy consumption due to their low cost. That is why these constrained devices are very heterogeneous in terms of their essential communication protocols, device data formats, and technologies.

Due to the heterogeneity of IoT, devices on the market nowadays have diversity in communication protocols, methods of network connectivity, and resulting models of application. It is not feasible to support such kinds of diversities in IoT, because developers typically lack the proper resources required to have a grip on the specifics of the constrained devices and network [16]. The main objective is to decrease, hide, or eradicate such a broad range of diversity of the technologies, applications models, and protocols from the users of IoT [17]. Because of the heterogeneous nature of IoT, it is one of the newly emerging research areas which has robust potential to bring a paradigm shift in the understanding of fundamental computer science principles and standards of our future living [18]. The demand for constrained IoT devices is expected to increase; this problem is expected to get worse in the future. Therefore, there will be a need to improve the integration of a large number of constrained devices in IoT. In this study, we have performed SLR aiming to identify the heterogeneity challenges and provide a summary of the solutions adopted for those challenges.

4. Research Methodology

A systematic literature review (SLR) methodology is adopted to identify the heterogeneity challenges in IoT systems, hindering a global IoT vision, and to find solutions to the identified heterogeneity challenges.

4.1. Research Questions. The first step of a systematic literature review is to define research questions. The research questions of this study are mentioned in Table 1.

4.2. Search String. The second phase of a systematic literature review is to find relevant studies on the research topic. We identified digital libraries in which primary search was carried out: IEEE Xplore, SpringerLink, Google Scholar, ScienceDirect, and ACM. We then defined a set of keywords related to our research topic: “Internet of Things,” “IoT,” “heterogeneity,” “heterogeneous,” and “challenges.” Finally, search strings were defined and used to collect published articles related to the research topic. Search strings are provided in Table 2.

4.3. Study Selection. The research selection process is to perform search in digital libraries based on the tollgate approach considering the search strings. Figure 1 shows a selection of articles using the tollgate approach.

In snowballing process, we selected 9 papers from journals (IEEE TMC, TPDS, JSAC, ToN, TWC) and conferences (SIGCOMM, MobiCom, MobiSys, INFOCOM), the content of which is analyzed and discussed in Section.

Initially, 3854 papers were selected by applying search protocol to the selected digital libraries. A selection process has been applied based on keywords, titles, duplication removal, abstracts, and full text of selected papers. We excluded the papers of the following types:

(1) Studies published in sources other than conferences, journals, patents, and technical reports
(2) Research papers not published in the English language
(3) Studies published before 2010
(4) Studies that are not related to the defined search strings

To evaluate the quality of the included research papers, we assessed the following aspects:

(1) The study provides information about any challenge related to IoT heterogeneity
(2) The study represents a clear solution to the identified challenge of heterogeneity
(3) The published study is from a stable and recognized publication source

Figure 2 shows a summary of paper selection based on digital libraries, and Figure 3 shows the paper selection on a yearly basis.

5. Analysis and Discussion

In this section, we have discussed the results relevant to our RQs. For the analysis, we used the linear chi-square test of association. For categorical values of predictor and outcome variables, the chi-square test is counted as more significant than other statistical tests. To answer RQ1, identified challenges/critical issues through the SLR are presented in Table 3. We have found a significant difference in heterogeneity challenges.

We set an occurrence percentage threshold of 30%. Accordingly, “Heterogeneity of devices,” “heterogeneity in formats of data,” and “heterogeneity in communication” are the most critical identified challenges.

5.1. Comparison of Challenges Based on Digital Libraries. Table 4 shows the analysis of the identified challenges based on digital libraries. We have Google Scholar, IEEE Xplore, SpringerLink, ScienceDirect, and ACM as digital libraries. From the analysis, we have found the following:

(1) Heterogeneity in communication is critical in Google Scholar and SpringerLink
(2) Heterogeneity of devices is critical in Google Scholar, IEEE Xplore, and SpringerLink

Table 1: Research questions.

No.	Research questions
RQ1	What are the challenges in heterogeneous IoT in the literature?
RQ2	What are the solutions to these challenges in heterogeneous IoT in the literature?

Table 2: Search strings.

Sources	Search string	Context
Google Scholar	(“Heterogeneous Internet of Things”) AND (“issues OR challenges”)	
IEEE Explore	“Heterogeneous” AND (“IoT” OR “Internet of Things”) AND “challenges”	
ScienceDirect	(“IoT” OR “Internet of Things”) AND (“heterogenous”) AND (“challenges”)	IoT heterogeneity
ACM	(“IoT”) AND (“heterogenous”) AND (“challenges”)	
SpringerLink	(“IoT”) AND (“heterogeneity”) AND (“challenges”)	

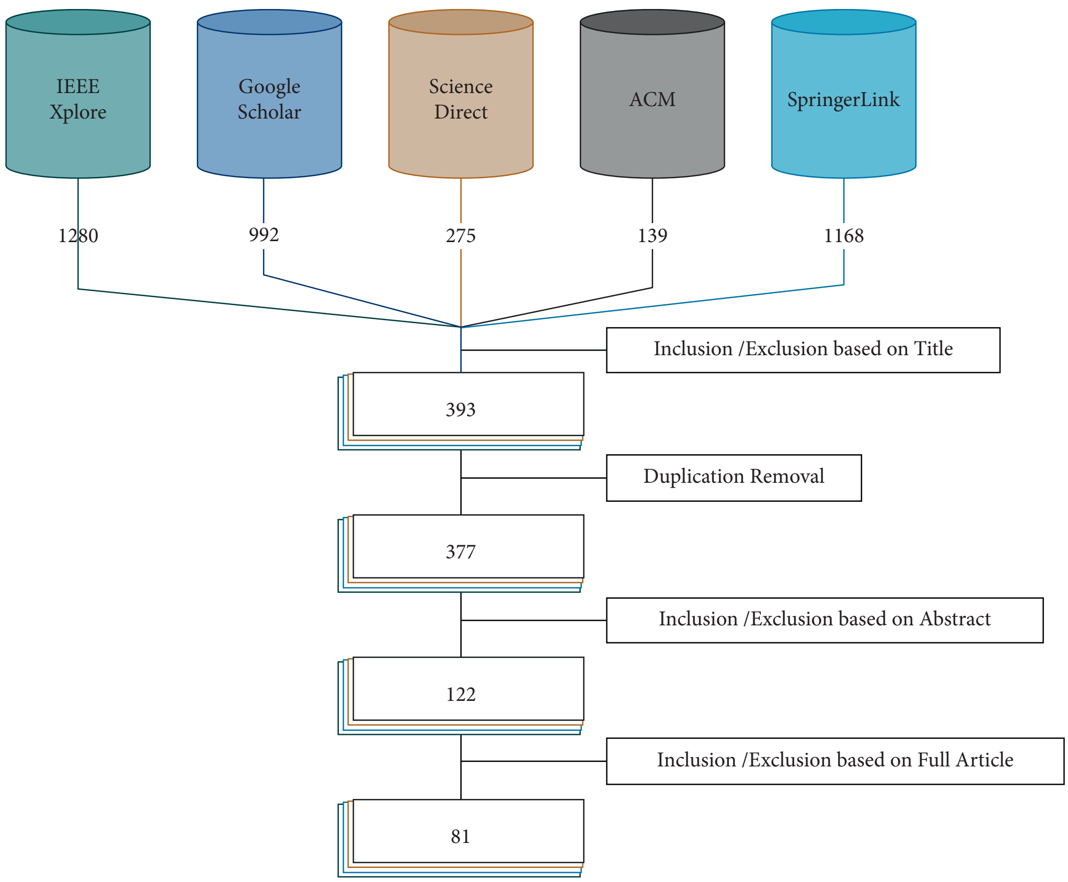


Figure 1: Selection of articles using the tollgate approach.

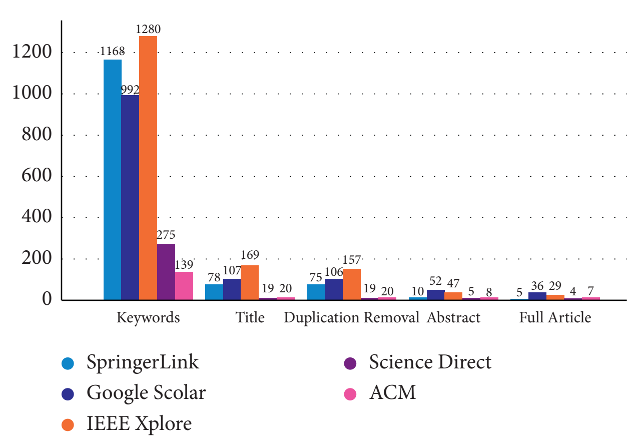


Figure 2: Paper selection based on digital libraries.

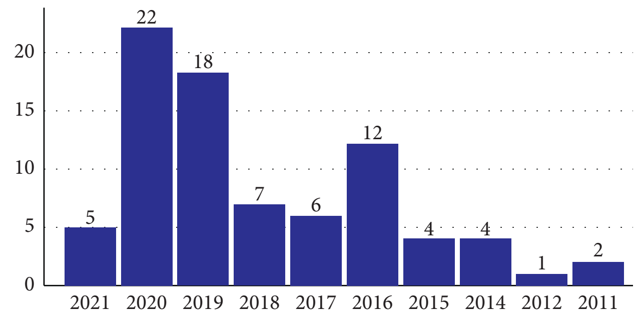


Figure 3: Paper selected on a yearly basis.

Table 3: Challenges identified through SLR.

S. No	Challenges	Frequency, 81	Percentage (%)	Papers ID
1	Fragmentation in connectivity, protocols	10	14	PiD4, PiD2, PiD4, PiD5, PiD36, PiD54, PiD60, PiD66, PiD74, PPiD78
2	Diversity in network technologies	9	13	PiD3, PiD6, PiD7, PiD8, PiD9, PiD10, PiD46, PiD54, PiD63
3	Management of networks	6	8	PiD8, PiD11, PiD13, PiD14, PiD15, PiD16
4	Heterogeneous communication issues	26	36	PiD3, PiD10, PiD11, PiD13, PiD15, PiD17, PiD20, PiD21, PiD26, PiD28, PiD29, PiD49, PiD57, PiD58, PiD61, PiD62, PiD64, PiD65, PiD68, PiD70, PiD72, PiD73, PiD75, PiD80, PiD81
5	Heterogeneity of devices issues	35	49	PiD2, PiD3, PiD8, PiD11, PiD12, PiD14, PiD16, PiD19, PiD20, PiD21, PiD22, PiD23, PiD24, PiD25, PiD26, PiD27, PiD28, PiD45, PiD46, PiD47, PiD50, PiD51, PiD56, PiD57, PiD58, PiD62, PiD63, PiD65, PiD68, PiD70, PiD71, PiD72, PiD77
6	Communication between heterogeneous devices	15	21	PiD3, PiD7, PiD10, PiD18, PiD28, PiD29, PiD30, PiD31, PiD32, PiD33, PiD47, PiD50, PiD53, PiD69, PiD71
7	Management and configuration of devices	13	18	PiD11, PiD14, PiD16, PiD30, PiD33, PiD34, PiD35, PiD36, PiD37, PiD54, PiD61, PiD62
8	Heterogeneous data/data formats	26	36	PiD1, PiD6, PiD10, PiD18, PiD25, PiD26, PiD27, PiD28, PiD32, PiD36, PiD38, PiD39, PiD40, PiD41, PiD42, PiD57, PiD58, PiD62, PiD64, PiD65, PiD68, PiD69, PiD70, PiD72, PiD79
9	Data security	12	17	PiD23, PiD31, PiD43, PiD44, PiD45, PiD46, PiD49, PiD52, PiD53, PiD76, PiD77
10	Communication security	16	22	PiD19, PiD23, PiD31, PiD32, PiD35, PiD43, PiD44, PiD47, PiD48, PiD49, PiD50, PiD51, PiD52, PiD53, PiD54, PiD55
11	Device security	14	20	PiD8, PiD19, PiD23, PiD24, PiD31, PiD43, PiD48, PiD49, PiD52, PiD53, PiD54, PiD55, PiD67
12	Heterogeneity in standards, platform	10	14	PiD1, PiD34, PiD36, PiD56, PiD57, PiD58, PiD59, PiD60, PiD66, PiD72
13	Integration of devices and data	14	20	PiD12, PiD18, PiD21, PiD23, PiD38, PiD41, PiD53, PiD56, PiD60, PiD61, PiD62, PiD63, PiD64, PiD65
14	Interoperability issue	16	22	PiD5, PiD9, PiD23, PiD27, PiD30, PiD37, PiD51, PiD59, PiD65, PiD66, PiD67, PiD68, PiD69, PiD70, PiD71, PiD72

(3) Heterogeneity in formats of data is critical in ScienceDirect, SpringerLink, and IEEE Xplore

(4) Interoperability issue is critical in ACM and SpringerLink

5.2. Comparison of Challenges Based on Timeframe. Table 5 shows the analysis of the identified challenges based on timeframe. We have divided the duration into two timeframes: Timeframe I from 2011 to 2016 and Timeframe II from 2017 to 2021. From the analysis, we have found the following:

Heterogeneity in communication is critical in Timeframe I from 2011 to 2016 as shown in Table 5.

(1) Heterogeneity of devices is critical in Timeframe I and Timeframe II

(2) Heterogeneity in formats of data is critical in Timeframe I and Timeframe II

5.3. Proposed Solutions. To answer RQ2, solutions to the identified challenges are presented in Table 6. We have found a total of 81 solutions for those challenges, with at least 5 unique solutions for each challenge.

Table 4: Summary of challenges based on digital libraries.

Challenges	Google Scholar, $n=36$		IEEE Xplore, $n=29$		ScienceDirect, $n=4$		ACM, $n=7$		SpringerLink, $n=5$		Chi-square test, $\alpha=0.05$	
	f	%	f	%	f	%	f	%	f	%	x^2	p
Fragmentation in connectivity, protocols	5	14	3	10	0	0	2	29	0	0	4.1393	0.0419
Diversity in network technologies	4	11	3	10	1	25	0	0	1	20	1.6701	0.1962
Management of networks	3	8	3	10	0	0	0	0	0	0	1.6683	0.1965
Heterogeneity in communication	12	33	7	24	1	25	1	14	2	40	0.3876	0.5336
Heterogeneity of devices	16	44	10	34	1	25	2	29	5	100	1.6342	0.2011
Communication between heterogeneous devices	7	19	6	21	1	25	0	0	1	20	1.4601	0.2269
Management and configuration of devices	8	22	3	10	0	0	1	14	0	0	2.9163	0.0877
Heterogeneity in formats of data	9	25	10	34	3	75	2	29	2	40	4.4138	0.0357
Data security	4	11	5	17	0	0	0	0	1	20	1.9809	0.1593
Communication security	8	22	5	17	1	25	1	14	1	20	0.5067	0.4766
Device security	5	14	6	21	0	0	1	14	1	20	1.3884	0.2387
Heterogeneity in standards, platform	7	19	1	2	0	0	1	14	1	20	3.4381	0.0637
Integration of devices and data	6	17	6	21	0	0	1	14	1	20	1.0421	0.3073
Interoperability issue	6	17	4	14	1	25	2	29	3	60	3.4303	0.0640

Table 5: Summary of challenges based on timeframe.

Challenges	Timeframe I (2011–2016), $n = 23$		Timeframe II (2017–2021), $n = 58$		Chi-square test, $\alpha = 0.05$	
	f	%	f	%	x^2	p
Fragmentation in connectivity, protocols	2	9	8	14	0.4918	0.4831
Diversity in network technologies	5	22	4	7	2.7502	0.0972
Management of networks	3	13	3	5	1.1185	0.2902
Heterogeneity in communication	8	35	18	31	0.0042	0.9478
Heterogeneity of devices	13	57	22	38	0.8052	0.3695
Communication between heterogeneous devices	4	17	11	19	0.0878	0.7669
Management and configuration of devices	3	13	10	17	0.3112	0.5769
Heterogeneity in formats of data	8	35	18	31	0.0042	0.9478
Data security	2	9	10	17	1.0399	0.3078
Communication security	3	13	13	22	0.9920	0.3192
Device security	2	9	12	20	1.6784	0.1951
Heterogeneity in standards, platform	3	13	7	12	0.0001	0.9901
Integration of devices and data	6	26	8	14	1.0677	0.3015
Interoperability issue	5	22	11	19	0.0086	0.9257

Table 6: Summary of solutions for identified challenges.

Challenge addressed	Proposed solutions			
	Ref	Year	Approach	Solution
Fragmentation in connectivity, data formats, and protocols	[4]	2015	Framework	ConnectOpen, providing a flexible communication agent deployed at the gateway
	[19]	2016	Platform	SPOT, a smartphone-based platform that makes use of open device driver models using XML
	[15]	2015	Platform	Cloud-based platform to integrate the services with communication models and constrained devices
	[20]	2020	Architecture	The recursive Inter-network architecture-based approach that reduces the protocol complexity and improves standardization
	[21]	2019	Protocol	Coexistent routing and flooding (CRF) using unique features of the physical layer technology-to-physical communication method for concurrent routing
	[22]	2020	Platform	iArk, a universal tracking platform for all types of IoT devices operating in the very high-frequency band
	[23]	2020	Mechanism	New roaming mechanism for LoRaWAN (low bandwidth wide area network) protocol based on reliable 5G network
Diversity in network technologies	[24]	2019	Architecture	Fog computing-based, multi-technology service architecture for IoT devices
	[25]	2011	Framework	IDRA, reconfigurable network framework to directly connect the devices that are correlated to each other
	[26]	2019	Framework	SDN-IoT, a framework that provides the functionality of converting m heterogeneous controllers to n homogeneous controllers
	[27]	2017	Middleware	A smartphone-based mobile gateway that provides an interface between devices and the Internet, being flexible and transparent
	[28]	2020	Proposed system	A decentralized, blockchain-based cloud solution for creating complex services of the network at the edge using IoT devices
Management of networks	[29]	2016	Architecture	A solution based on the utilization of dockers implemented on devices
	[30]	2016	Architecture	Combining both direct and indirect current management approaches
	[31]	2020	Model	Message-based communication model consists of a dictionary of services for devices and servers to interact
	[32]	2019	Platform	M4DN.IoT, a platform for the management of IoT networks with a user-friendly interface
	[33]	2014	Architecture	Extending the multi-network information architecture (MINA) middleware with SDN multilayer IoT controller
	[34]	2014	Framework	Framework for managing and configuring the network dynamically based on SDN

Table 6: Continued.

Challenge addressed	Proposed solutions			
	Ref	Year	Approach	Solution
Heterogeneity in communication	[35]	2016	Algorithm	Hierarchal clustering algorithm for dynamic and heterogeneous IoT
	[36]	2016	Framework	Relying on a device and distributed SDN connectivity to overcome the issue of heterogeneous communication methods used in IoT
	[37]	2016	Proposed system	TACIoT, a flexible and reliable IoT access control system
	[38]	2020	Framework	Knowledge-based framework using edge computing for heterogeneous connectivity in the Internet of Things networks
	[39]	2021	Algorithm	Distributed online optimization algorithm based on game theory and optimization theory. The algorithm works online and jointly decides to offload heterogeneous tasks, allocate computing resources, and manage battery power
	[40]	2020	Model	Optimal geographic distribution across heterogeneous networks with caching support. Extending optimization to heterogeneous networks using simulated user distributions
	[41]	2020	Framework	Elastic zoom algorithm for cells based on the end-user quality of service and traffic load
	[42]	2019	Platform	MINOS, multi-protocol software that defines a networking platform
Heterogeneity of devices	[43]	2017	Platform	ThingsJS, a JavaScript-based middleware platform and runtime environment that bypasses system-specific complexities
	[44]	2014	Architecture	Architecture, combined with cognitive capabilities, that supports intelligent decision-making and automates service creation
	[45]	2017	Platform	IoTOne, software platform to support heterogeneous Internet of Things devices and allow robust control of devices
	[46]	2020	Middleware	Cuttlefish, lightweight and flexible middleware having unified APIs for application development for heterogeneous device utilization
	[47]	2020	Middleware	MSOAH-IoT is based on a service-oriented architecture that handles various networking interfaces and collects data using REST API
	[48]	2017	Architecture	A middleware architecture and edge-based protocol that enables heterogeneous edge devices to dynamically exchange data and resources to improve application performance and privacy
	[49]	2019	Framework	A novel communication framework that enables simultaneous N-Way communication between Wi-Fi and bluetooth low energy (BLE) devices
	[50]	2020	Mechanism	eWoT, a semantically interactive ecosystem of IoT devices that provides SPARQL query-based mechanism for transparent discovery and access to IoT devices
Communication between heterogeneous devices	[51]	2012	Framework	Resource-oriented middleware framework using blockchain technology
	[52]	2018	Architecture	Use of a multimodal *d* employing a variety of heterogeneous wireless networks
	[53]	2019	Model	Ontology-based device semantic web rule language between multiple devices in a heterogeneous system
	[54]	2020	Protocol	A device-to-device lightweight security protocol based on a symmetric key scheme to ensure secure communication between devices
	[55]	2019	Middleware	PICO, REST web service-based data-centric middleware for real-time communication and storage of data
	[56]	2018	Architecture	A fully decentralized IoT access control system, based on blockchain technology architecture
Management and configuration of devices	[32]	2019	Platform	M4DN.IoT, a platform for the management of IoT networks with a user-friendly interface
	[57]	2016	Framework	EC-IoT, making use of an open standard upon IoT communication protocol (COAP)
	[58]	2019	Architecture	An improved architecture for managing, monitoring, and configuring IoT devices, based on private blockchain
	[59]	2020	Platform	An open and scalable IoT platform having edge computing characteristics
	[60]	2020	Framework	DIAM-IoT, a framework for IoT device decentralized identification and access management

Table 6: Continued.

Challenge addressed	Proposed solutions			
	Ref	Year	Approach	Solution
Heterogeneity in formats of data	[61]	2016	Framework	SIGHTED, a framework based on the semantic web and connected data principles
	[62]	2017	Framework	A solution consisting of Internet gateway device functions, NoSQL database, web services, and IoT application
	[63]	2011	Framework	SeaCloudDM, a novel sea-cloud-based heterogeneous data management framework
	[64]	2020	Proposed system	MusQ, a solution that provides a multi-storing query system for IoT data using a formal unified query language (MQL)
	[68]	2020	Mechanism	Ethereum blockchain consisting of generic and constrained devices that connect to the blockchain via a wired and wireless heterogeneous network
	[65]	2020	Architecture	Software architecture for processing and analyzing data from heterogeneous sources with different structures in IoT scopes
Data security	[66]	2021	Framework	Distributed multiparty secure computing framework for data authentication of devices
	[67]	2018	Framework	Framework for access control in IoT using block chain technology
	[69]	2020	Proposed system	PEIoT, a data streaming system with enhanced privacy (i) Providing users with the ability to subscribe to/unsubscribe from data (ii) Enabling data controllers to invoke various privacy-enhancing technologies
	[70]	2019	Model	Two modules: (i) Secure data processing system to maintain IoT data security and integrity (ii) Drone-based data analytic system using edge computing and on-device computing
	[71]	2016	Architecture	A novel architecture consisting of a distributed interface having e-nodes—inexpensive, simple, and embedded nodes
Communication security	[72]	2019	Protocol	Multigroup key management protocol to ensure upstream and downstream secrecy, recovery from collision attacks, and network security
	[73]	2019	Mechanism	A physical layer security mechanism to validate the single source for heterogeneous IoT
	[74]	2020	Protocol	Using proxy re-signature, a privacy-preserving authentication protocol for heterogeneous system
	[75]	2019	Method	SMER, a method for exchanging resources between heterogeneous devices securely
	[76]	2015	Algorithms	Optimized elliptic curve cryptography (ECC) algorithms for the devices based on NXP/Jennic JN5148
Device security	[77]	2019	Proposed system	Novel and lightweight authentication and key agreement scheme for heterogeneous IoT devices
	[78]	2018	Algorithm	ECC-based algorithm to authenticate and authorize new devices in a network
	[79]	2021	Framework	MECshield, a DDoS prevention framework, based on mobile edge computing (MEC)
	[80]	2020	Proposed system	Novel decentralized authentication mechanism based on blockchain technology
Heterogeneity in standards, platform	[81]	2018	Method	Smart governance approach for heterogeneous IoT system management
	[82]	2019	Framework	BRAIN-IoT, a framework and methodology for interconnecting heterogeneous platforms and automation
	[83]	2016	Model	The data model used in the VITAL project (open source IoT system of systems) for rapid development of IoT-based systems
	[84]	2021	Platform	Data Spine, federated platform for bridging interoperability gaps between heterogeneous IoT platforms
	[85]	2017	Model	A concept of generic driver injection for developing mobile applications that can be deployed in a variety of environments using different middleware

Table 6: Continued.

Challenge addressed	Ref	Year	Approach	Solution
				Proposed solutions
Integration of devices and data	[86]	2019	Proposed system	SensPnP, a novel plug-and-play solution which combines hardware with firmware
	[87]	2020	Architecture	Blockchain-based architecture to improve integration and reduce computation overhead and energy consumption
	[88]	2015	Proposed system	A solution based on management distribution of devices among gateways and making use of web service delegation
	[89]	2016	Architecture	Architecture for data integration from heterogeneous data sources like government agencies and unreliable sources
	[90]	2020	Model	Point-to-point integration model in IoT applications, with three layers: hardware, communication, and integration
Interoperability issue	[91]	2019	Model	Three internetworking models based on the status of the city
	[92]	2020	Framework	AFaaS, authorization framework as a services to support interoperability challenge
	[93]	2018	Framework	SHIOT, an SDN-based framework which is based on ontology and applies SDN controller
	[94] [95]	2018	Platform	New design of decentralized IoT platform having capabilities of edge computing, fog computing, and cloud computing
	[96] [90] [91] [92] [93]	2014 2022 2022 2022 2019	Framework	A lightweight, middleware-independent development framework for interoperability monitoring
	[97]	2021	Framework	Trust-based middleware framework for managing interoperability challenges in heterogeneous IoT
	[98]	2016 2020 2021 2022	Method	Model-based engineering methods to ensure those complex software systems are interoperable with each other

6. Conclusion and Future Work

This research is a systematic literature review that reviews the literature in the domain of IoT heterogeneity. The review has been implemented using a systematic methodology to select different studies addressing the challenges faced by heterogeneous IoT. To conduct this study, a total of 81 research papers that are published in different digital libraries from 2011 to 2021 were selected. For analysis purposes, we divided this duration into two timeframes. One is from 2011 to 2016, and the other is from 2017 to 2021. In this SLR, we have identified 14 different heterogeneity challenges that need to be addressed to implement IoT on a large scale. Challenges with occurrence percentage of more than 30% are defined as the most critical ones. In this study, we analyze the occurrence of those challenges based on digital libraries as well as timeframe. In this analysis, we found out that some challenges were more critical in the earlier timeframe than in the recent timeframe.

We also found out that some challenges are still critical in both timeframes. After identifying those challenges, we found at least 5 solutions for each identified challenge. The summary of those solutions is given in Table 6. In future work, we want to categorize the challenges and prioritize the solutions by using a multi-criteria decision-making problem.

Data Availability

The data collected during the data collection phase will be provided by the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

References

[1] I. Within and C. Farook, "IoT generic frameworks: what needs to improve," in *Proceedings of the 2019 7th International Conference on Smart Computing & Communications (ICSCC)*, pp. 1–5, IEEE, Sarawak, Malaysia, June 2019.

[2] A. H. Mohd Aman, E. Yadegaridehkordi, Z. S. Attarbashi, R. Hassan, and Y.-J. Park, "A survey on trend and classification of internet of things reviews," *IEEE Access*, vol. 8, pp. 111763–111782, 2020.

[3] D. C. Yacchirema Vargas and C. E. Palau Salvador, "Smart IoT gateway for heterogeneous devices interoperability," *IEEE Latin America Transactions*, vol. 14, no. 8, pp. 3900–3906, 2016.

[4] N. Pazos, M. Müller, M. Aeberli, and N. Ouerhani, "ConnectOpen-automatic integration of IoT devices," in *Proceedings of the 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pp. 640–644, IEEE, Milan, Italy, December 2015.

[5] S. Madakam, R. Ramaswamy, S. Tripathi, and V. Lake, "Internet of things (IoT): a literature review," *Journal of Computer and Communications*, vol. 03, no. 05, pp. 164–173, 2015.

[6] J. Huang, B. Lv, Y. Wu, Y. Chen, and X. Shen, "Dynamic admission control and resource allocation for mobile edge computing enabled small cell network," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 2, pp. 1964–1973, 2021.

[7] Y. Chen, F. Zhao, Y. Lu, and X. Chen, "Dynamic task offloading for mobile edge computing with hybrid energy supply," *Tsinghua Science and Technology*, vol. 10, 2021.

[8] G. Nunberg, *The Advent of the Internet, 12th April, Courses*, 2012.

[9] "Internet," 1996, V. Beal, https://www.webopedia.com/definitions/internet/.

[10] I. L. Stats, "Internet world Statistics," 2021, https://www.internetlivestats.com/.

[11] X. Zhang, H. Chen, Y. Zhao et al., "Improving cloud gaming experience through mobile edge computing," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 178–183, 2019.

[12] P. P. Ray, "A survey on Internet of Things architectures," *Journal of King Saud University - Computer and Information Sciences*, vol. 30, no. 3, pp. 291–319, 2018.

[13] L. Ericsson, *More than 50 Billion Connected Devices—Taking Connected Devices to Mass Market and Profitability*, Technology report, 2011.

[14] D. Evans, "The internet of things: How the next evolution of the internet is changing everything 2011," 2015, http://www.cisco.com/web/about/ac79/docs/innov/IoTIBSG0411FINAL.pdf.

[15] F. Van den Abeele, J. Hoebeke, I. Moerman, and P. Demeester, "Integration of heterogeneous devices and communication models via the cloud in the constrained internet of things," *International Journal of Distributed Sensor Networks*, vol. 2015, no. 10, pp. 1–16, 2015.

[16] J. Huang, C. Zhang, and J. Zhang, "A multi-queue approach of energy efficient task Scheduling for sensor Hubs," *Chinese Journal of Electronics*, vol. 29, no. 2, pp. 242–247, 2020.

[17] X. Zhang, Z. Qi, G. Min, W. Miao, Q. Fan, and Z. Ma, "Cooperative edge Caching based on Temporal Convolutional networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 9, pp. 2093–2105, 2022.

[18] T. Qiu, N. Chen, K. Li, M. Atiquzzaman, and W. Zhao, "How can heterogeneous internet of things build our future: a survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2011–2027, 2018.

[19] M.-M. Moazzami, G. Xing, D. Mashima, W.-P. Chen, and U. Herberg, "SPOT: a smartphone-based platform to tackle heterogeneity in smart-home IoT systems," in *Proceedings of the 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pp. 514–519, IEEE, Reston, VA, USA, December 2016.

[20] R. Herrero, "6LoWPAN fragmentation in the context of IoT based media real time communication," *Internet Technology Letters*, vol. 3, no. 2, p. e144, 2020.

[21] W. Wang, X. Liu, Y. Yao, Y. Pan, Z. Chi, and T. Zhu, "Crf: Coexistent routing and flooding using wifi packets in heterogeneous iot networks," in *Proceedings of the IEEE INFOCOM 2019-IEEE conference on computer communications*, pp. 19–27, IEEE, Paris, France, May 2019.

[22] Z. An, Q. Lin, P. Li, and L. Yang, "General-purpose deep tracking platform across protocols for the internet of things," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, pp. 94–106, ACM, 2020.

[23] E. M. Torroglosa-Garcia, J. M. A. Calero, J. B. Bernabe, and A. Skarmeta, "Enabling roaming across heterogeneous IoT wireless networks: LoRaWAN MEETS 5G," *IEEE Access*, vol. 8, pp. 103164–103180, 2020.

[24] H. Le, N. Achir, and K. Boussetta, "Fog computing architecture with heterogeneous Internet of Things technologies," in *Proceedings of the 2019 10th International Conference on Networks of the Future (NoF)*, pp. 130–133, IEEE, Rome, Italy, October 2019.

[25] E. De Poorter, I. Moerman, and P. Demeester, "Support for heterogeneous dynamic network environments through a reconfigurable network service platform," in *Proceedings of the 2011 1st International Symposium on Access Spaces (ISAS)*, pp. 174–179, IEEE, Yokohama, Japan, June 2011.

[26] K. Sood, K. K. Karmakar, S. Yu, V. Varadharajan, S. R. Pokhrel, and Y. Xiang, "Alleviating heterogeneity in SDN-IoT networks to Maintain QoS and Enhance security," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5964–5975, 2020.

[27] G. Aloi, G. Caliciuri, G. Fortino et al., "Enabling IoT interoperability through opportunistic smartphone-based mobile gateways," *Journal of Network and Computer Applications*, vol. 81, pp. 74–84, 2017.

[28] I. Al Ridhawi, M. Aloqaily, A. Boukerche, and Y. Jaraweh, "A blockchain-based decentralized composition solution for iot services," in *Proceedings of the Icc 2020-2020 ieee international conference on communications (icc)*, pp. 1–6, IEEE, Dublin, Ireland, June 2020.

[29] I. Bedhief, M. Kassar, and T. Aguili, "SDN-based architecture challenging the IoT heterogeneity," in *Proceedings of the 2016 3rd Smart Cloud Networks & Systems (SCNS)*, pp. 1–3, IEEE, Dubai, United Arab Emirates, December 2016.

[30] C. Pham, Y. Lim, and Y. Tan, "Management architecture for heterogeneous IoT devices in home network," in *Proceedings of the 2016 IEEE 5th Global Conference on Consumer Electronics*, pp. 1–5, IEEE, Kyoto, Japan, October 2016.

[31] B. Oniga, L. Denis, V. Dadarlat, and A. Munteanu, "Message-based communication for heterogeneous internet of things systems," *Sensors*, vol. 20, no. 3, p. 861, 2020.

[32] J. D. C. Silva, J. J. P. C. Rodrigues, K. Saleem, S. A. Kozlov, and R. A. L. Rabelo, "M4DN.IoT-A networks and devices management platform for internet of things," *IEEE Access*, vol. 7, pp. 53305–53313, 2019.

[33] Z. Qin, G. Denker, C. Giannelli, P. Bellavista, and N. Venkatasubramanian, "A software defined networking architecture for the internet-of-things," in *Proceedings of the 2014 IEEE network operations and management symposium (NOMS)*, pp. 1–9, IEEE, Krakow, Poland, May 2014.

[34] H. Huang, J. Zhu, and L. Zhang, "An SDN_based management framework for IoT devices," in *Proceedings of the 25th IET Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014)*, pp. 175–179, IET, Limerick, June 2014.

[35] J. S. Kumar and M. A. Zaveri, "Hierarchical clustering for dynamic and heterogeneous internet of things," *Procedia Computer Science*, vol. 93, pp. 276–282, 2016.

[36] A. Desai, K. Nagegowda, and T. Ninikrishna, "A framework for integrating IoT and SDN using proposed OF-enabled management device," in *Proceedings of the 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, pp. 1–4, IEEE, Nagercoil, India, March 2016.

[37] J. Bernal Bernabe, J. L. Hernandez Ramos, and A. F. Skarmeta Gomez, "TACIoT: multidimensional trust-aware access control system for the Internet of Things," *Soft Computing*, vol. 20, no. 5, pp. 1763–1779, 2016.

[38] R. Xu, W. Jin, and D. H. Kim, "Knowledge-based edge computing framework based on CoAP and HTTP for enabling heterogeneous connectivity," *Personal and Ubiquitous Computing*, vol. 26, no. 2, pp. 329–344, 2020.

[39] S. Xia, Z. Yao, Y. Li, and S. Mao, "Online distributed offloading and computing resource management with energy harvesting for heterogeneous MEC-enabled IoT," *IEEE Transactions on Wireless Communications*, vol. 20, no. 10, pp. 6743–6757, 2021.

[40] J. Yang, C. Ma, B. Jiang, G. Ding, G. Zheng, and H. Wang, "Joint optimization in cached-enabled heterogeneous network for efficient industrial IoT," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 831–844, 2020.

[41] H. Jiang, Z. Xiao, Z. Li, J. Xu, F. Zeng, and D. Wang, "An energy-efficient framework for internet of things underlaying heterogeneous small cell networks," *IEEE Transactions on Mobile Computing*, vol. 21, no. 1, pp. 31–43, 2022.

[42] T. Theodorou, G. Violettas, P. Valsamas, S. Petridou, and L. Mamatas, "A multi-protocol software-defined networking solution for the Internet of Things," *IEEE Communications Magazine*, vol. 57, no. 10, pp. 42–48, 2019.

[43] J. Gascon-Samson, M. Rafiuzzaman, and K. Pattabiraman, "Thingsjs: towards a flexible and self-adaptable middleware for dynamic and heterogeneous iot environments," in *Proceedings of the 4th Workshop on Middleware and Applications for the Internet of Things*, pp. 11–16, December 2017.

[44] C. Sarkar, A. U. Nambi, R. V. Prasad, A. Rahim, R. Neisse, and G. Baldini, "DIAT: a scalable distributed architecture for IoT," *IEEE Internet of Things Journal*, vol. 2, no. 3, pp. 230–239, 2015.

[45] N. Gyory and M. Chuah, "IoTOne: Integrated platform for heterogeneous IoT devices," in *Proceedings of the 2017 International Conference on Computing, Networking and Communications (ICNC)*, pp. 783–787, IEEE, Silicon Valley, CA, USA, January 2017.

[46] A. Pamboris, C. Kozis, and H. Herodotou, "Cuttlefish: a flexible and lightweight middleware for combining heterogeneous IoT devices," in *Proceedings of the 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–6, IEEE, Las Vegas, NV, USA, January 2020.

[47] Y. Mesmoudi, M. Lamnaour, Y. El Khamlichi, A. Tahiri, A. Touhafi, and A. Braeken, "A middleware based on service oriented architecture for heterogeneity issues within the internet of things (MSOAH-IoT)," *Journal of King Saud University - Computer and Information Sciences*, vol. 32, no. 10, pp. 1108–1116, 2020.

[48] A. Essameldin and K. A. Harras, "The hive: an edge-based middleware solution for resource sharing in the internet of things," in *Proceedings of the 3rd Workshop on Experiences with the Design and Implementation of Smart Objects*, pp. 13–18, ACM, October 2017.

[49] Z. Chi, Y. Li, H. Sun, Y. Yao, and T. Zhu, "Concurrent cross-technology communication among heterogeneous IoT devices," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 932–947, 2019.

[50] A. Cimmino, M. Poveda-Villalón, and R. García-Castro, "ewot: a semantic interoperability approach for heterogeneous iot ecosystems based on the web of things," *Sensors*, vol. 20, no. 3, p. 822, 2020.

[51] Y. Hong, "A resource-oriented middleware framework for heterogeneous internet of things," in *Proceedings of the 2012 international conference on cloud and service computing*, pp. 12–16, IEEE, Shanghai, China, November 2012.

[52] J. Famaey, R. Berkvens, G. Ergeerts et al., "Flexible multimodal sub-gigahertz communication for heterogeneous internet of things applications," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 146–153, 2018.

[53] S. Suman, T. Perumal, N. Mustapha, and R. Yaakob, "Device Verification and Compatibility for heterogeneous semantic IoT systems," in *Proceedings of the 2019 4th International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE)*, pp. 1–3, IEEE, Kedah, Malaysia, November 2019.

[54] X. Luo, L. Yin, C. Li et al., "A lightweight privacy-preserving communication protocol for heterogeneous IoT environment," *IEEE Access*, vol. 8, pp. 67192–67204, 2020.

[55] J. Chen, E. Cañete, D. Garrido, M. Díaz, and K. Piotrowski, "PICO: a platform independent communications middleware for heterogeneous devices in smart grids," *Computer Standards & Interfaces*, vol. 65, pp. 1–14, 2019.

[56] O. Novo, "Blockchain meets IoT: an architecture for scalable access management in IoT," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184–1195, 2018.

[57] E. Dalipi, F. Van den Abeele, I. Ishaq, I. Moerman, and J. Hoebeke, "EC-IoT: an easy configuration framework for constrained IoT devices," in *Proceedings of the 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pp. 159–164, IEEE, Reston, VA, USA, December 2016.

[58] K. Košťál, P. Helebrandt, M. Belluš, M. Ries, and I. Kotuliak, "Management and monitoring of IoT devices using blockchain," *Sensors*, vol. 19, no. 4, p. 856, 2019.

[59] A. Javed, A. Malhi, T. Kinnunen, and K. Framling, "Scalable IoT platform for heterogeneous devices in smart environments," *IEEE Access*, vol. 8, pp. 211973–211985, 2020.

[60] X. Fan, Q. Chai, L. Xu, and D. Guo, "DIAM-IoT: a decentralized identity and access management framework for internet of things," in *Proceedings of the 2nd ACM International Symposium on Blockchain and Secure Critical Infrastructure*, pp. 186–191, ACM, October 2020.

[61] A. M. Nagib and H. S. Hamza, "SIGHTED: a framework for semantic integration of heterogeneous sensor data on the internet of things," *Procedia Computer Science*, vol. 83, pp. 529–536, 2016.

[62] E. S. Pramukantoro, W. Yahya, G. Arganata, A. Bhawiyuga, and A. Basuki, "Topic based IoT data storage framework for heterogeneous sensor data," in *Proceedings of the 2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA)*, pp. 1–4, IEEE, Lombok, Indonesia, October 2017.

[63] Z. Ding, Q. Yang, and H. Wu, "Massive heterogeneous sensor data management in the Internet of Things," in *Proceedings of the 2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing*, pp. 100–108, IEEE, Dalian, China, October 2011.

[64] H. Ramadhan, F. I. Indikawati, J. Kwon, and B. Koo, "MusQ: a Multi-store query system for iot data using a datalog-like language," *IEEE Access*, vol. 8, pp. 58032–58056, 2020.

[65] D. Corral-Plaza, I. Medina-Bulo, G. Ortiz, J. Boubeta-Puig, and User Group, "A stream processing architecture for heterogeneous data sources in the Internet of Things," *Computer Standards & Interfaces*, vol. 70, p. 103426, 2020.

[66] Y. D. Al-Otaibi, "Distributed multi-party security computation framework for heterogeneous internet of things (IoT) devices," *Soft Computing*, vol. 25, no. 18, pp. 12131–12144, 2021.

[67] C. Dukkipati, Y. Zhang, and L. C. Cheng, "Decentralized, blockchain based access control framework for the

heterogeneous internet of things," in *Proceedings of the Third ACM Workshop on Attribute-Based Access Control*, pp. 61–69, ACM, March 2018.

[68] S. Misra, A. Mukherjee, A. Roy, N. Saurabh, Y. Rahulamathavan, and M. Rajarajan, "Blockchain at the edge: Performance of resource-constrained IoT networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 1, pp. 174–183, 2021.

[69] S. Ghayyur, P. Pappachan, G. Wang, S. Mehrotra, and N. Venkatasubramanian, "Designing privacy preserving data sharing middleware for internet of things," in *Proceedings of the Third Workshop on Data: Acquisition To Analysis*, pp. 1–6, ACM, November 2020.

[70] M. S. Islam, H. Verma, L. Khan, and M. Kantarcioglu, "Secure real-time heterogeneous iot data management system," in *Proceedings of the 2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, pp. 228–235, IEEE, Los Angeles, CA, USA, December 2019.

[71] S. Sicari, C. Cappiello, F. De Pellegrini, D. Miorandi, and A. Coen-Porisini, "A security-and quality-aware system architecture for Internet of Things," *Information Systems Frontiers*, vol. 18, no. 4, pp. 665–677, 2016.

[72] M. A. Kandi, H. Lakhlef, A. Bouabdallah, and Y. Challal, "An efficient multi-Group Key management protocol for heterogeneous IoT devices," in *Proceedings of the 2019 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, IEEE, Marrakesh, Morocco, April 2019.

[73] S. Yu, X. Zhang, P. Huang, and L. Guo, "Secure authentication in cross-technology communication for heterogeneous IoT," in *Proceedings of the 2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 1-2, IEEE, Newark, NJ, USA, November 2019.

[74] H. Xiong, Y. Wu, C. Jin, and S. Kumari, "Efficient and privacy-preserving authentication protocol for heterogeneous systems in IIoT," *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11713–11724, 2020.

[75] Y. Zhang, Y. Han, and J. Wen, "SMER: a secure method of exchanging resources in heterogeneous internet of things," *Frontiers of Computer Science*, vol. 13, no. 6, pp. 1198–1209, 2019.

[76] L. Marin, M. P. Pawlowski, and A. Jara, "Optimized ECC implementation for secure communication between heterogeneous IoT devices," *Sensors*, vol. 15, no. 9, pp. 21478–21499, 2015.

[77] J. Liu, A. Ren, L. Zhang, R. Sun, X. Du, and M. Guizani, "A novel secure authentication scheme for heterogeneous internet of things," in *Proceedings of the ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, Shanghai, China, May 2019.

[78] S. Sasirekha, S. Swamynathan, and S. Suganya, "An ECC-based algorithm to handle secure communication between heterogeneous IoT devices," in *Advances in Electronics, Communication and Computing*, pp. 351–362, Springer, Singapore, 2018.

[79] N.-N. Dao, T. V. Phan, U. Sa'ad et al., "Securing heterogeneous iot with intelligent ddos attack behavior learning," *IEEE Systems Journal*, vol. 16, no. 2, pp. 1974–1983, 2022.

[80] A. Mostafa, "Blockchain-based distributed authentication Mechanism for internet-of-things devices," in *Proceedings of the 2020 9th International Conference on Software and Information Engineering (ICSIE)*, pp. 159–164, November 2020.

[81] A. Kazmi, M. Serrano, and A. Lenis, "Smart governance of heterogeneous internet of things for smart cities," in *Proceedings of the 2018 12th International Conference on Sensing Technology (ICST)*, pp. 58–64, IEEE, Limerick, Ireland, December 2018.

[82] D. Conzon, M. R. A. Rashid, X. Tao, A. Soriano, R. Nicholson, and E. Ferrera, "BRAIN-IoT: model-based framework for dependable sensing and actuation in intelligent decentralized IoT Systems," in *Proceedings of the 2019 4th International Conference on Computing, Communications and Security (ICCCS)*, pp. 1–8, IEEE, Rome, Italy, October 2019.

[83] A. Kazmi, Z. Jan, A. Zappa, and M. Serrano, "Overcoming the heterogeneity in the internet of things for smart cities," in *International Workshop on Interoperability and Open-Source Solutions*, pp. 20–35, Springer, Cham, 2016.

[84] R. A. Deshmukh, D. Jayakody, A. Schneider, and V. Damjanovic-Behrendt, "Data spine: a federated interoperability enabler for heterogeneous IoT platform ecosystems," *Sensors*, vol. 21, no. 12, p. 4010, 2021.

[85] K. Saatkamp, U. Breitenbücher, F. Leymann, and M. Wurster, "Generic driver injection for automated IoT application deployments," in *Proceedings of the 19th International Conference on Information Integration and Web-based Applications & Services*, pp. 320–329, ACM, Salzburg, Austria, December 2017.

[86] S. K. Roy, S. Misra, and N. S. Raghuwanshi, "SensPnP: Seamless integration of heterogeneous sensors with IoT devices," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 2, pp. 205–214, 2019.

[87] L. Tseng, L. Wong, S. Otoum, M. Aloqaily, and J. B. Othman, "Blockchain for managing heterogeneous internet of things: a perspective architecture," *IEEE network*, vol. 34, no. 1, pp. 16–23, 2020.

[88] A. C. Olivieri and G. Rizzo, "Scalable approaches to integration in heterogeneous IoT and M2M scenarios," in *Proceedings of the 2015 9th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pp. 358–363, IEEE, Santa Catarina, Brazil, July 2015.

[89] F. Montori, L. Bedogni, and L. Bononi, "On the integration of heterogeneous data sources for the collaborative Internet of Things," in *Proceedings of the 2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*, pp. 1–6, IEEE, Bologna, Italy, September 2016.

[90] D. Alulema, J. Criado, L. Iribarne, A. J. Fernández-García, and R. Ayala, "A model-driven engineering approach for the service integration of IoT systems," *Cluster Computing*, vol. 23, no. 3, pp. 1937–1954, 2020.

[91] J. Hwang, J. An, A. Aziz, J. Kim, S. Jeong, and J. Song, "Interworking models of smart city with heterogeneous internet of things standards," *IEEE Communications Magazine*, vol. 57, no. 6, pp. 74–79, 2019.

[92] S.-R. Oh and Y.-G. Kim, "AFaaS: Authorization framework as a service for Internet of Things based on interoperable OAuth," *International Journal of Distributed Sensor Networks*, vol. 16, no. 2, Article ID 155014772090638, 2020.

[93] H. A. Tran, D. Tran, L. G. Nguyen, Q. T. Ha, V. Tong, and A. Mellouk, "SHIOT: a novel SDN-based framework for the heterogeneous internet of things," *Informatica*, vol. 42, no. 3, 2018.

[94] B. Skrbic, D. Radovanovic, S. Tomovic, L. Lazovic, Z. Zecevic, and I. Radusinovic, "A decentralized platform for heterogeneous IoT networks management," in *Proceedings of the 2018 23rd International Scientific-Professional Conference on Information Technology (IT)*, pp. 1–4, IEEE, Zabljak, Montenegro, February 2018.

[95] J. Huang, B. Lv, Y. Wu, Y. Chen, and X. Shen, "Dynamic admission control and resource allocation for mobile edge computing enabled small cell network," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 2, pp. 1964–1973, 2022.

[96] P. Grace, J. Barbosa, B. Pickering, and M. Surridge, "Taming the interoperability challenges of complex iot systems," in *Proceedings of the 1st ACM Workshop on Middleware for Context-Aware Applications in the IoT*, pp. 1–6, ACM, Bordeaux, France, December 2014.

[97] M. A. Abbasi, Z. A. Memon, N. M. Durrani, W. Haider, K. Laeeq, and G. A. Mallah, "A multi-layer trust-based middleware framework for handling interoperability issues in heterogeneous IoTs," *Cluster Computing*, vol. 24, no. 3, pp. 2133–2160, 2021.

[98] P. Grace, B. Pickering, and M. Surridge, "Model-driven interoperability: engineering heterogeneous IoT systems," *Annals of telecommunications*, vol. 71, no. 3-4, pp. 141–150, 2016.

Hindawi
Scientific Programming
Volume 2022, Article ID 2673753, 8 pages
https://doi.org/10.1155/2022/2673753

Review Article

An Integration of IoT, IoC, and IoE towards Building a Green Society

Sultan Ahmad,[1] Sudan Jha,[2] Hikmat A. M. Abdeljaber,[3] Mohammad Khalid Imam Rahmani,[4] Mohammad Maqbool Waris,[5] Ajit Singh,[6] and Muhammad Yaseen[7]

[1]*Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Alkharj 11942, Saudi Arabia*
[2]*Department of Computer Science and Engineering, School of Engineering, Kathmandu University, Banepa, Kathmandu, Nepal*
[3]*Department of Computer Science, Faculty of Information Technology, Applied Science Private University, Amman, Jordan*
[4]*College of Computing and Informatics, Saudi Electronic University, Riyadh 11673, Saudi Arabia*
[5]*Department of Mechanical Engineering, Adama Science and Technology University, Adama, Ethiopia*
[6]*Patna Women's College, Patna University, Patna, India*
[7]*Faculty of Information Technology, University of Central Punjab, Lahore, Pakistan*

Correspondence should be addressed to Mohammad Maqbool Waris; mohammad.waris@astu.edu.et

Received 8 April 2022; Revised 1 June 2022; Accepted 14 June 2022; Published 5 July 2022

Academic Editor: Jiwei Huang

Energy waste altogether adds to expanded expenses in the car fabricating industry, which is liable to energy use limitations and tax assessment from national and global strategy creators and confinements and charges from national energy suppliers. This checking is essential for energy sparing since it empowers organizations to roll out operational improvements to diminish energy utilization and expenses. The primary test to energy observation is the need to incorporate assembling and energy checking and control gadgets that help diverse correspondence conventions and are generally dispersed over a wide region. One of the most significant challenges in the advancement of the Internet of Things (IoT) has been the powering of billions of connected devices. Evaluation of digital services considering an energy impression of the Internet normally requires models of the energy intensity of the Internet. A typical way to deal with the display of the energy intensity is to consolidate assessments of market studies of introduced gadgets on a national or worldwide scale and their related power utilization with the aggregate information volume transported at a similar scale. Energy sources are a fundamental part of society development, and a steady power supply is essential for today's progress. End-use energy is transferred to various consumers via power transmission and circulation networks after being transformed to optional energy as electricity by various power facilities. The power grid serves as the physical stage for both wide-area electric power sharing and display exchanges, and it is at the heart of auxiliary energy sources. In this manner, it attempts to connect the part of a center point between essential energy and end-use energy. With the bidirectional power stream given by the Energy Internet, different techniques are elevated to enhance and increase the energy usage between Energy Internet and Main-Grid. Energy proficiency and, in addition, quick information transmission are fundamental to green correspondences-based applications for IoT. Here, we are trying to provide a state-of-the-art survey over various Internet of Energy techniques along with IoT.

1. Introduction

The Internet of Computer (IoC) has been widely used since 1991, which is utilized for people's interaction for a long time. New energy advancements for co-created warmth and power and expanded utilization of renewable ones, for example, biomass, sun-powered energy, and wind power, should be incorporated in a clever, data-based worldwide energy framework. In that sense, we have a worldview change from existing aloof and data-poor systems to

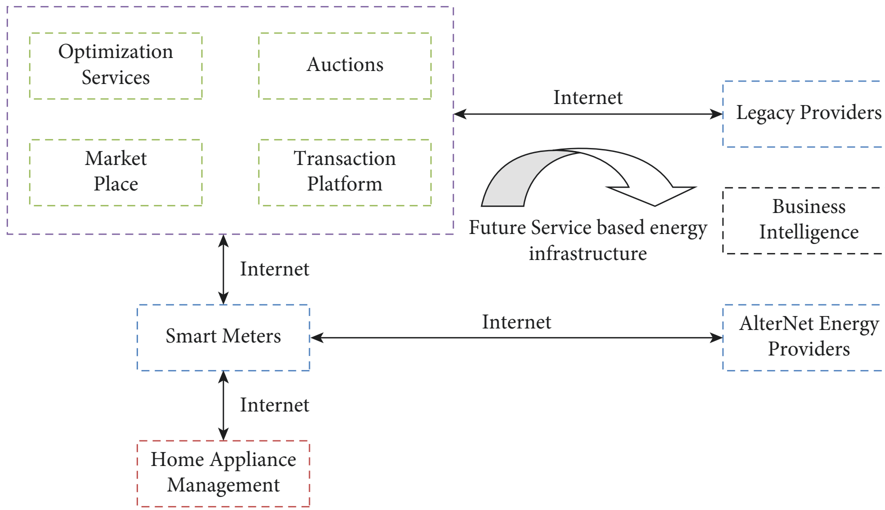


Figure 1: Internet of Energy.

dynamic data-rich energy systems, which alters the course of one-path stream since the power systems were started. The integration of little much conveyed energy creation sources and their coupling with cutting edge data-driven administrations will offer ascent to another foundation that we allude to as Internet of Energy as shown in Figures 1 and 2.

In this way, the Internet-based foundation will be firmly combined with the energy area and is used to help the improvement of new components for exchange in view of free market activity in the power advertise. New data subordinate astute energy administration systems will be required for a framework equipped for supporting the deregulated energy advertise. In such an intricate and dynamic system, it is normal that disseminated energy makers and devouring substances will be much interconnected likewise by means of data streams. The expectation is to build up a free and focused market for energy generation and circulation by separating the esteemed chain creation, exchange, and appropriation of electrical power. ICT will make it feasible for future dispersed energy systems to act naturally and automatically in more better ways and will empower dynamic revamping and coordination of administration markets. Diverse models and situations for an exceptionally dispersed data-based energy framework will be developed [1].

IoE utilizes the bidirectional stream of energy and information inside the smart grid to increase profound bits of knowledge on power use and predicts future activities to build energy effectiveness and low general cost [2]. The combination of WSNs, actuators, smart meters, and different parts of the power grid together with information and communication technology (ICT) is alluded to as the Internet of Energy (IoE) [3, 4]. As of now, the utilization of renewable energy (counting hydroelectric energy) in the United States is just 6.8% of the aggregate energy expended, which is a level much lower than other created countries, where petroleum product costs are generally higher than the United States [5]. Therefore, to empower across the board usage of long haul, secure, feasible, and naturally benevolent

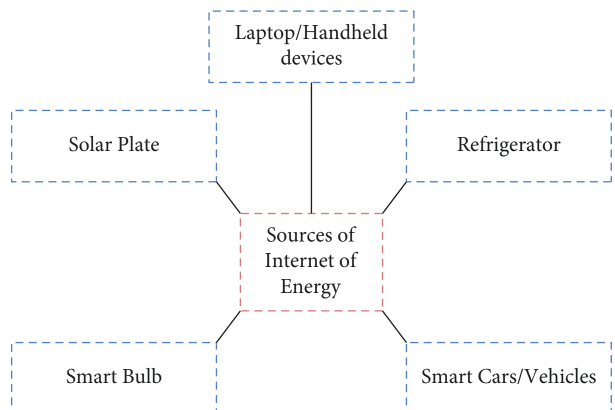


Figure 2: Source of Internet of Energy.

energy, the future electric distribution grid must address the issues of capacity and complex control.

According to the US Energy Information Administration, the US consumed more than 27 trillion kilowatt hours (kWh) in 2007, with electric energy accounting for about 11 trillion kWh and transportation accounting for 8.4 trillion kWh. The capacity to properly harness these assets must be evaluated in preparation for the United States' advancement to a greater level of renewable energy consumption. Renewable energy assets must be implemented in two ways to achieve this goal: large-scale centralized setups (such as wind or solar-based farms) and large-scale DRERs [6].

As nations around the globe put more efforts in green energy and renewable resources, the wasteful aspects of existing power frameworks around the globe are regularly neglected. This implies renewable energy cannot be given at its ideal level of proficiency on the grounds that the grid cannot completely bolster it. For instance, China is one of the world's biggest makers of renewable energy; however, despite everything, it encounters deficiencies and energy emergencies since it cannot convey that energy at a level that can support its populace [7]. This outcome results in power blackouts and holes. The energy exists yet the framework

Table 1: Comparative analysis of previous works [9].

S. No.	Authors	Advantages of the work	Limitations of work
1	Kumar and Sukumar (2018) [10]	Proposed a novel scalar point multiplication	ECC method cannot handle the DoS assaults
2	Yau et al. (2018) [11]	Provided a review of smart devices in heterogeneous environments and analysed the energy consumption of IoT devices	It did not address the major challenging issues
3	Wang et al. (2017) [12]	Analysed the development from smart grid to EI	It did not address the security of electrical systems and correspondence systems
4	Zhou et al. (2016) [13]	Analysed IoE (Internet of Energy) with the business perspective	It did not address various aspects like behavioural, security, and administrative issues
5	Kamalinejad et al. (2015) [14]	Reviewed on productive WEH and life-time of WEH-empowered IoT gadgets	It only outlined how to empower WEH for IoT frameworks but no proposed solutions were given
6	Kaur and Sood (2015) [15]	Defined three layers Detecting and control Data handling and presentation	Calculation of accelerating the calculation time is not enough for energy design
7	Moness et al. (2015) [16]	A review for cutting edge layers was exhibited	It did not address the social impact
8	Tao et al. (2014) [17]	A new ESER assessment was analysed	Lack of design, development, and use of ESER
9	Nieminen et al. (2014) [18]	Technique to connect bluetooth LE devices with the IoT	It did not focus on smart devices administration and their improvement
10	Shrouf et al. (2014) [19]	Focused on energy administration; in addition to this, it also addressed how to enhance energy	It does not focus on recovery of information
11	Cao and Yang (2013) [20]	Focuses on energy "web" issues	It does not address the upcoming problems like energy production, consumption, and sharing
12	Krenge et al. (2013) [21]	Introduced energy name service (ENS) inside the IoE	But it did not address how ENS prohibits security issues and how the objects are recognized
13	Kelly et al. (2013) [22]	Proposed techniques to check easiness universal detecting framework	It does not focus on the next network technologies including IPv6

does not. Correspondingly, the nation delivers an enormous number of electric vehicles; however, it does not have adequate charging stations, so the vehicles cannot work. In 2014, China's energy misfortune because of wasteful aspects of foundations was greater than the energy utilized every year by numerous nations around the globe.

2. Future of IoE

One potential answer to the issue of energy inefficiency is ultra-high voltage transmission (UHV), a framework that enables very fast energy transmission over long distances with minimal power loss. Adoption has been relatively slow over the years after all the benefits of UHV. China initially actualized UHV in 2009, yet its advancement is continually extending to take care of demand. China is attempting to computerize appropriation and add more resources to take care of demand, including all the more charging stations for electric autos. It is likewise building stockpiling destinations, especially in those urban areas that utilize the most energy, keeping in mind the end goal to store extra energy productively and near where it will be required. This will have included financial advantages for the organizations providing renewable energy, for example, sun powered and twists, because of the way that more energy will be held and sold, notwithstanding moderately low stockpiling costs.

In the future as the world moves in the direction of dependence on and reaping of renewable energy sources, the utilization of nonrenewable resources relied can fall, which will decrease the requirement for obsolete foundations that handle resources, for example, coal and oil. The IoE is the consequence of the execution of IoT innovation with distributed energy systems. Its motivation is to upgrade the productivity of the age, transmission, and usage of electricity. IoT innovation empowers the IoE by making networks of sensors that have various smart grid applications. These incorporate power observation, request side energy administration, distributed capacity, and renewable energy combination among others. The IoT is estimated to add $14 trillion to the worldwide economy by 2030, and the market for computerized gadgets that empower the IoE is probably going to develop to $89.4 billion by 2030. Smart sensor networks are generally cheap so they can be comprehensively sent at scale bringing about a huge measure of information which can be examined to uncover approaches to upgrade grid proficiency [8].

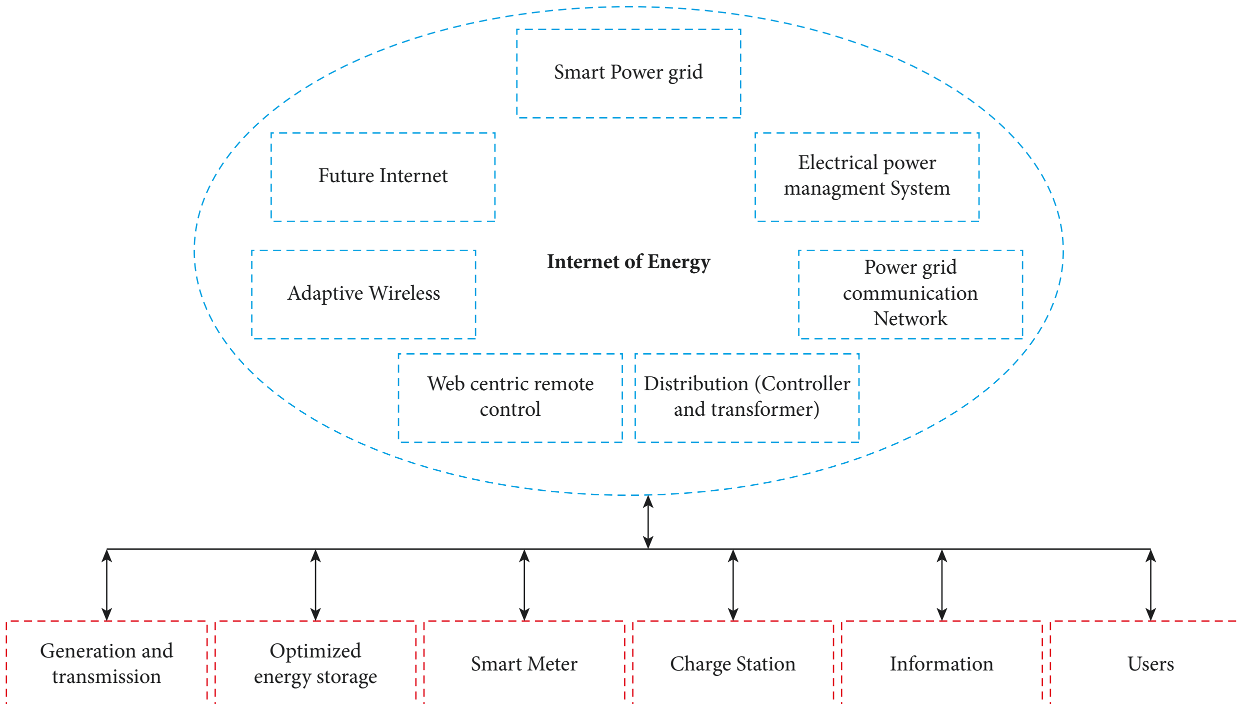


Figure 3: Concepts of Internet of Energy.

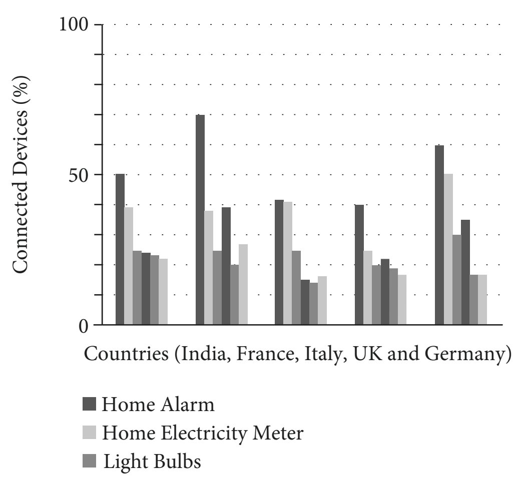


Figure 4: Devices that consumers want to connect.

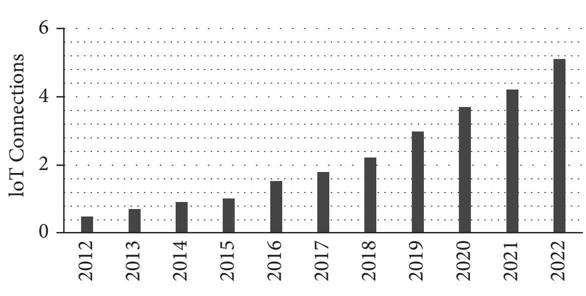


Figure 5: IoT connections.

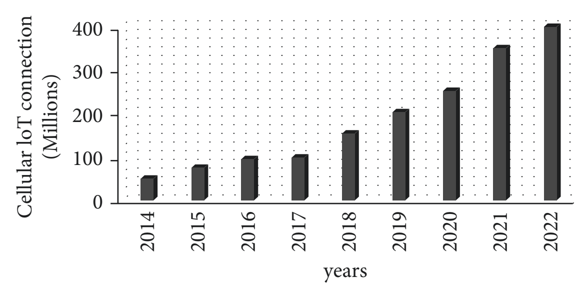


Figure 6: IoT consumer perspective.

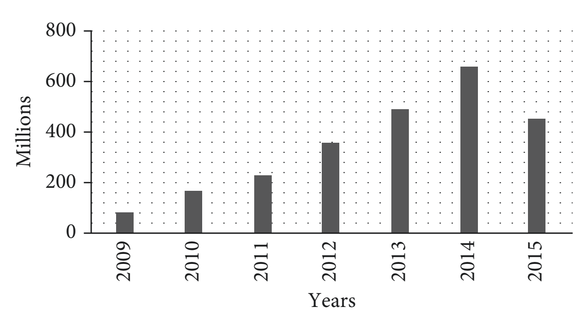


Figure 7: Statistic of I3 connect platform.

3. Related Work

In the last decade, there are many potential works have been done on green society and Internet of Energy. These works have their own advantages and limitations. Biswaranjan et al.

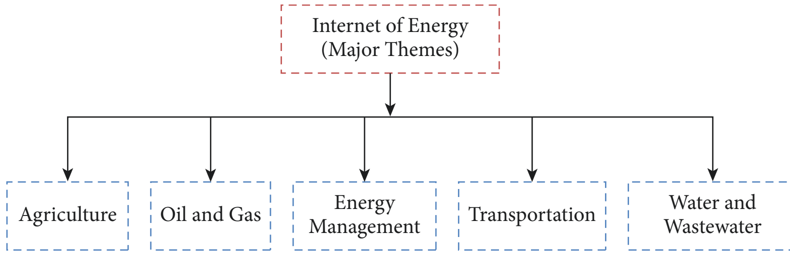


Figure 8: i3Connect platform.

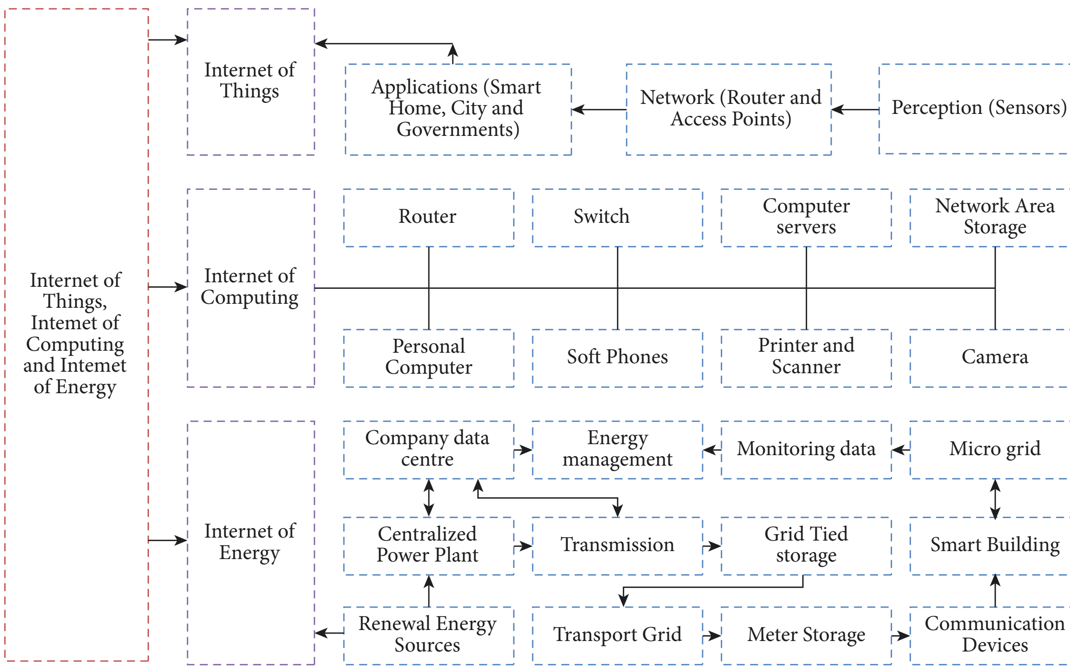


Figure 9: Taxonomy IoT, IoC, and IoE.

have done a very comprehensive comparative study and included cloud integration with Lora watermeter network in their paper [9] available as preprint on Research Square. In Table 1, the brief comparative analysis of these works is being referred.

4. Internet of Energy for Electric Mobility

The task will address reference plans and installed framework designs for exceptionally proficient, imaginative smart network frameworks with respect to prerequisites of similarity, organization, security, power, conclusion, support, coordinated asset administration, and self-association [23]. Figure 3 shows the concept of Internet of Energy in graphical form.

Productive, spotless, sheltered, and consistent portability: IoE proposes inventive answers for interfacing the Internet with the power grid with applications for electric versatility, making transport more maintainable, proficient, perfect, protected, and consistent [24, 25].

Finally, from the perspective of customers, IoT is beginning to expand beyond entertainment and communication devices. Phones, laptops, televisions, and game consoles are all connected to wearable devices [26]. For example, 8% of western Europeans possesses a smart watch, with additional 8% expecting to get one within the next year.

Figures 4 and 5 demonstrate a strong desire to connect additional household gadgets and appliances. As seen in Figure 6, this interest is closely related to the utility industry's growth around smart meters and consumer energy management. Georgina Penfold, Director of Spraga Ltd and winner of Energy Industry Expert of the Year at the 2016 Energy Live Consultancy Awards expressed the following: "A current survey of substantial vitality end-clients (£500 K per annum power spend or more noteworthy) affirmed that about 20% of vitality supervisors trust suspicion of the innovation among the chiefs will be a huge boundary to the appropriation of keen resources inside the business".

As the following outline represents, our i3Connect (Figure 7) stage has followed five years of consistent development of wander venture into IoE, finishing in \$657 million put crosswise over 106 arrangements in 2014. We trust this could be a monstrous open door for financial esteem creation; however, the vitality business will require greater security and unwavering quality in contrast with what is satisfactory for buyer applications [27].

Table 2: List of different areas and the works.

Domain	Area	Work
Energy efficiency and Savings	Architectural flexibility	Integration of HVAC with service oriented architecture
	Optimization using big data	Big data and cloud
	Big data	New platform 'open Fridge' created to collect data.
	User satisfaction from services	Dynamic power allocation to optimize user satisfaction
Networking and security	Low power wide-area network	Wide coverage of network (low cost and adaptive data rate)
	Reduction in packet size	Used 6LoWPAN over IPv6 along with CoAP
	Developed security framework	The problem of multimedia heterogeneity is addressed and new framework proposed
	New system architecture	New architecture is proposed to handle large-scale data based on CoAP
	Introduction of Fog computing in IoT (indirectly leading to low latency and wide coverage)	Thought of Fog computing as a unifying platform for new services
Safety	Safe disposal of dam tailings	The tailings dam monitoring and prealarm system along with cloud for real-time monitoring and safe disposal
	Transportation	Smart-eye technology developed for real-time monitoring of vehicle data
	Healthcare	Real-time monitoring of patient data using CoAP and integration of different low-cost devices

Note that these numbers exclude consumer (and other) IoT applications to concentrate on Industrial IoE over various distinctive parts. For instance, organizations like WellAware and PetroFeed have brought up capital in 2014 to convey IoE to the oil and gas industry (Figure 8).

With the presentation of associated sensors and cloud-based information logical frameworks, customers and districts are currently ready to change conventional sprinkling frameworks into dynamic savvy watering structures to accomplish the most ideal water utilization [28]. Organizations like aWhere, CropX, and Granular are helping ranchers better comprehend climate information, water utilization, and vitality utilization. Organizations, for example, Enlighted, Blue Pillar, and Simple Energy, are helping undertakings to enhance lighting effectivenessand turn out as more vital. Zenodys, the Internet of Things platform, solution is based on the Zenodys visual IoT platform that handles most of the tasks without any programming: ZenoConnect provides connectivity of various smart energy vendors (houses, windmills, PV panels, batteries, etc.) and connects them to the microgrid.

The solution is fully independent and can run offline, but for demonstration purposes, we also enabled remote solution delivery and Docker-based service deployments. Zenobox physically connects all the equipment and services and provides a runtime environment.

The vision of OSI model is integration of all power systems and power grid apparatus. It contains procedures of generation, transport and conveyance, utility store network, buyers and customers' energy-utilizing contraptions, and machines or any systems that add to the creation or utilization of the electron. The request reaction will use the virtual power plant as far as oversaw stacks and oversaw supplies, for example, electric vehicle battery stockpiling, etc. With IoT-E, purchasers without bounds can turn into a virtual small-scale power plant, offering their overabundance energy, put-away energy, or sustainable age into the grid. Here are some fundamental intriguing ideas: "IoT-E permits the flip of a solitary light switch to put into impact a procedure that flags all procedures upstream to account, alter for and begin a chain response including all parts of the energy creation, conveyance, and upkeep and production network."

5. Taxonomy of Different Technologies

In general, IoT systems make use of a three-layered architecture consisting of the application, network, and perception layer [29]. The generation of data is solely handled by the perception layer [30]. It chiefly consists of sensors fitted onto various devices. Other sources of information such as computers, mobile devices, social media, forms, audio data, speech data, image data, and power generation data may also be part of this layer. The taxonomy of IoT, IoC, and IoE is shown in Figure 9.

This layer plays a crucial role in data transfer from sensors to the user applications because data are of crucial importance and this is the only medium of communication (wired or wireless) between a user and a device. The data generated from smart energy devices can be used to gain insights and predict outcomes that can be useful for optimization. This huge influx of data can overwhelm existing systems, and incorporation of cloud computing technology might be beneficial to address this issue [31]. Table 2 shows it in detail:

5.1. Combination of the Three Fields: A Prophecy. The integration of cloud computing and the architecture for heterogeneous devices can give a pathway to increased security and scalability in the context of smart grids. Pervasiveness of IoT-based devices and smart appliances has led to the problem of continuous authentication which may be addressed by the different security frameworks established. Seemingly enough, as mentioned before, there has been work done on the integration of real-time monitoring systems with existing ones in order to ensure optimized performance and

Table 3: List of related problems, possible solutions, and outcomes.

Area	Problem	Possible solution	Outcome
Smart grid	Scalability and mobility	Integration of new heterogeneous cloud architecture, different IoT devices, and the security framework (Perles et al. 2018).	Data security and ease of real-time monitoring; scalability achieved
User satisfaction	Companies not able to gain trust and bad user experience	Integration of CoAP with cloud with security architecture mentioned [12]	Low latency and high throughput achieved leading to better performance
Environment	Needless consumption of natural resources to generate energy	Integration of cloud and real-time monitoring systems (Wanyama 2018)	Optimized consumption of natural resources for energy production

Table 4: Comparison of different domains, related problems, and solutions.

Domain	Cloud	Enhanced network	Real-time monitoring	Sensors
Smart grid	Yes	Yes	Yes	Yes
User Satisfaction	Yes	Yes	Yes	Optional
Environment	Yes	Optional	Yes	Yes

reduce losses [32, 33]. These are listed in Table 3. This higher performance along with new security architectures are more robust and less prone to cyber-attacks, making people put their trust on such organizations [34, 35].

5.2. Parametric Comparison. Table 4 shows a parametric comparison of different domains of problem addressed in Table 4. The fields that are having 'YES' represent the availability of the required parameter for optimal performance and 'OPTIONAL' represent that the inclusion of the parameter is not of utmost importance. However, one's aim should be to include the integration of all the technologies into an application to achieve good cooperation among the three technologies.

6. Conclusion

IoT innovation is developed alongside the savvy power grid to make our day by day life more brilliant and less demanding. Energy supply, as well as monetary and natural security, is all interwoven and influences people's development. Far-reaching energy security, based on the security of power grids, should be emphasized in the electric power business. As a result, essential and end-use energy conversion is tightly linked, demanding a large-scale energy strategy. We have reviewed innovations and plans to empower IoE for IoT frameworks. The IoE worldview with its guarantee of more effective energy administration and appropriation speaks to the future for all on-screen characters in the energy advertise. Accordingly, a market incorporates energy providers, affiliates, and innovation suppliers and additionally clients of various types (residential, modern, business), and it is a principal need to construct the savvy grid idea with respect to institutionalized and interoperable apparatuses. Standard arrangements will build advertise infiltration: it is very simple for a client to utilize another innovation on the off chance that it is worked over (and available through) all around trying and generally utilizing apparatuses, for example, plain Web 2.0-empowered programs. for example, plain Web 2.0-empowered programs. In any case, the savvy grid will presumably be an exceptionally complex framework, made of various elements, likely passage-level expenses, however perhaps constrained abilities. It is thus key that empowering advances be standard, as well as adequately lightweight to take a shot at these gadgets. We have attempted to give a short outline of IoE over IoT stage. In future, we will try to implement something to work into this huge IoE and IoT field.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] G. Pepermans, D. H. Driesen, R. Belmans, and W. D. Haeseleer, "Distributed generation: definition, benefits and issues," *Energy Policy*, vol. 33, no. 6, pp. 85–94, 2005.

[2] S. Karnouskos and O. Terzidis, "Towards an information infrastructure for the future internet of energy," in *Proceedings of the 2007 ITG-GI Conference Communication in Distributed Systems (KiVS)*, pp. 1–6, VDE, Bern, Switzerland, 2007 February.

[3] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart grid - the new and improved power grid: a survey," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 944–980, 2012.

[4] J. Huang, B. Lv, Y. Wu, Y. Chen, and X. Shen, "Dynamic admission control and resource allocation for mobile edge computing enabled small cell network," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 2, pp. 1964–1973, 2022.

[5] V. Gungor, B. Lu, and G. Hancke, "Opportunities and challenges of wireless sensor networks in smart grid," *IEEE*

Transactions on Industrial Electronics, vol. 57, no. 10, pp. 3557–3564, 2010.
[6] A. Q. Huang, M. L. Crow, G. T. Heydt, J. P. Zheng, and S. J. Dale, "The future renewable electric energy delivery and management (FREEDM) system: the energy internet," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 133–148, 2011.
[7] Z. Xie, "Energy insecurity and renewable energy policy: comparison between the People's Republic of China and Japan," 2020, https://policycommons.net/artifacts/392394/energy-insecurity-and-renewable-energy-policy/1357112/.
[8] S. Karnouskos, "The cooperative internet of things enabled smart grid," in *Proceedings of the 14th IEEE international symposium on consumer electronics (ISCE2010)*, pp. 07–10, Braunschweig, Germany, June 2010.
[9] B. Bhola, R. Kumar, H. Long, I. Priyadarshini, and N. Son, "Cloud integrated with LoRa watermeter network: a water expense repository," 2021, https://assets.researchsquare.com/files/rs-835166/v1_covered.pdf?c=1635431660.
[10] K. S. Kumar and R. Sukumar, "Achieving energy efficiency using novel scalar multiplication based ECC for android devices in Internet of Things environments," *Cluster Computing*, vol. 22, pp. 1–8, 2018.
[11] Y. Yang, M. Zhong, H. Yao, F. Yu, X. Postolache, and O. Postolache, "Internet of things for smart ports: technologies and challenges," *IEEE Instrumentation and Measurement Magazine*, vol. 21, no. 1, pp. 34–43, 2018.
[12] K. Wang, X. Hu, Y. Sun, D.-J. Deng, A. Vinel, and Y. Zhang, "Wireless big data computing in smart grid," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 58–64, 2017.
[13] C. Zhou, M. Zelinka, and S. Klein, "Impact of decadal cloud variations on the Earth's energy budget," *Nature Geoscience*, vol. 9, no. 12, pp. 871–874, 2016.
[14] P. Kamalinejad, C. Mahapatra, Z. Sheng, S. Mirabbasi, V. C. M. Guan, and Y. L. Guan, "Wireless energy harvesting for the internet of things," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 102–108, 2015.
[15] N. Sood and S. K. Sood, "An energy-efficient architecture for the internet of things (IoT)," *IEEE Systems Journal*, vol. 11, no. 2, pp. 796–805, June 2017.
[16] M. Moustafa and A. M. Moustafa, "A survey of cyber-physical advances and challenges of wind energy conversion systems: prospects for internet of energy," *IEEE Internet of Things Journal*, vol. 3, no. 2, pp. 134–145, 2016.
[17] W. Kim, T. He, D. Wang, C. Liang, and S. Liang, "Assessment of long-term sensor radiometric degradation using time series analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 5, pp. 2960–2976, 2014.
[18] J. Nieminen, C. Gomez, M. Isomaki et al., "Networking solutions for connecting bluetooth low energy enabled machines to the internet of things," *IEEE network*, vol. 28, no. 6, pp. 83–90, 2014.
[19] F. Shrouf, J. Ordieres, and G. Miragliotta, "Smart factories in Industry 4.0: a review of the concept and of energy management approached in production based on the Internet of Things paradigm," in *Proceedings of the 2014 IEEE International Conference onIndustrial Engineering and Engineering Management (IEEM)*, pp. 697–701, IEEE, Selangor Darul Ehsan, Malaysia, 2014 December.
[20] J. Cao and M. Yang, "Energy internet--towards smart grid 2.0," in *Proceedings of the 2013 4th International Conference onNetworking and Distributed Computing (ICNDC)*, pp. 105–110, IEEE, Los Angeles, CA, USA, 2013 December.
[21] J. Krenge, M. Scheibmayer, and M. Deindl, "Identification scheme and name service in the Internet of Energy," in *Proceedings of the 2013 IEEE PES Innovative Smart Grid Technologies (ISGT)*, pp. 1–6, IEEE, Washington DC, USA, 2013 February.
[22] S. D. T. Kelly, N. K. Suryadevara, and S. C. Mukhopadhyay, "Towards the implementation of IoT for environmental condition monitoring in homes," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3846–3853, 2013.
[23] J. Huang, C. Zhang, and J. Zhang, "A multi-queue approach of energy efficient task scheduling for sensor hubs," *Chinese Journal of Electronics*, vol. 29, no. 2, pp. 242–247, 2020.
[24] H. Shahinzadeh, J. Moradi, G. B. Gharehpetian, H. Abedi, and M. Abedi, "Internet of energy (IoE) in smart power systems," in *Proceedings of the 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*, pp. 627–636, Tehran, Iran, March 2019.
[25] S. Jha, S. Ahmad, D. Prashar et al., "A proposed waiting time algorithm for a prediction and prevention system of traffic accidents using smart sensors," *Electronics*, vol. 11, no. 11, p. 1765, 2022.
[26] Xu Zhang, H. Chen, Y. Zhao et al., "Improving cloud gaming experience through mobile edge computing," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 178–183, 2019.
[27] S. Ahmad, M. M. Afzal, and A. Alharbi, "Big data analytics with fog computing in integrated cloud fog and IoT architecture for smart devices," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 06, pp. 171–177, 2020.
[28] Y. Chen, Z. Liu, Y. Zhang, Y. Wu, X. Chen, and L. Zhao, "Deep reinforcement learning-based dynamic resource management for mobile edge computing in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 4925–4934, 2021.
[29] Xu Zhang, Z. Qi, G. Min, M. Miao, Q. Fan, and Ma Ma, "Cooperative edge caching based on temporal convolutional networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 9, pp. 2093–2105, 2022.
[30] S. Ahmad, "A review on edge to cloud: paradigm shift from large data centers to small centers of data everywhere," in *Proceedings of the 2020 International Conference on Inventive Computation Technologies (ICICT)*, pp. 318–322, IEEE, Lalitpur, Nepal, February 2020.
[31] S. Ahmad, A. Alharbi, A. Zamani, and M. Yousufuddin, "Implementation of fusion and filtering techniques in IoT data processing: a case study of smart healthcare," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 20, no. 12, pp. 131–137, 2020.
[32] Y. Chen, W. Gu, and K. Li, "Dynamic task offloading for internet of things in mobile edge computing via deep reinforcement learning," *International Journal of Communication Systems*, 2022.
[33] Y. Chen, F. Zhao, Y. Lu, and X. Chen, "Dynamic task offloading for mobile edge computing with hybrid energy supply," *Tsinghua Science and Technology*.
[34] S. Ahmad and M. M. Afzal, "Deployment of fog and edge computing in IoT for cyber-physical infrastructures in the 5G era" in sustainable communication networks and application," in *Proceedings of the International Conference on Sustainable Communication Networks and Application (ICSCN 2019)*, vol. 39, Erode, India, July 2020.
[35] J. Huang, Z. Tong, and Z. Feng, "Geographical POI recommendation for internet of things: a federated learning approach using matrix factorization," *International Journal of Communication Systems*, 2022.

Hindawi
Scientific Programming
Volume 2022, Article ID 8520447, 10 pages
https://doi.org/10.1155/2022/8520447

Research Article

Two FCA-Based Methods for Reducing Energy Consumption of Sensor Nodes in Wireless Sensor Networks

Yuxia Lei [1], **Meiyan Qu**,[1] **Chen Lei**,[1] **Zhiqiang Kong**,[1] **Jingying Tian**,[2] **and Shi Wang**[3]

[1]*School of Computer Science, Qufu Normal University, Rizhao 276800, Shandong, China*
[2]*School of Architectural Engineering, Rizhao Polytechnic, Rizhao 276800, Shandong, China*
[3]*Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China*

Correspondence should be addressed to Yuxia Lei; yx_lei@126.com

Received 8 April 2022; Revised 4 June 2022; Accepted 13 June 2022; Published 5 July 2022

Academic Editor: Ying Chen

In the design of wireless sensor networks (WSNs), it is important to reduce energy consumption and extend the service life of the sensors. Selecting one of the minimum sensor combinations (MSCs) that can monitor all areas, while the other MSCs are asleep, can effectively extend the lifetime of WSNs. This paper proposes two algorithms based on Formal Concept Analysis (FCA) for extracting some MSCs, to minimize the energy consumption and meet the coverage requirement. These two methods firstly extract the concept lattice from a monitor-areas context, and then it is simple to extract sensor nodes monitoring the overlapping area from the concept lattice. The algorithms consist of three steps as follows: the first step is to transform sensors and monitoring areas into a context, the second is to extract the concept lattice and implications among areas, and the third is to extract some different MSCs that can monitor all areas. Thus, some strategies are designed to awaken different MSCs to achieve the purpose of reducing energy consumption. Experimental results show that these methods have played a positive effect on extracting different MSCs and extending the lifetime of sensor networks.

1. Introduction

With the popularity and rapid development of the fields of Internet of Things, there are many meaningful research directions such as point of interest recommendation [1], reducing the energy consumption of WSNs [2, 3], dynamic task offloading [4], and admission control for edge computing [5]. In recent years, WSNs have been more and more widely used in economic life, which are usually made up of a large number of sensor nodes, and are scalable [6, 7]. But because of their limited power and computing power, they are often placed in areas that are hard for people to reach [2]. Energy is a valuable resource in WSNs, and the energy consumption of sensor nodes is the main energy consumption. Therefore, sensor power supply capability is the main obstacle that limits the application of sensor network technology [8]. However, the power of sensor nodes is limited and difficult to replace [9–11], how to reduce the energy consumption of sensor nodes becomes the core problem of WSNs [12–14].

FCA is proposed by Wille R. in 1982, which focuses on the lattice structure induced by a binary relation between a pair of sets (respectively called objects and attributes), known as the Galois lattice or the concept lattice of the relation [15–18]. A node of concept lattice is a pair of objects and attributes, called a formal concept. It consists of two parts: the extent and the intent. The extent is the set of objects which have all attributes in the intent. Similarly, the intent is the set of attributes common to the objects in the extent. A process of generating the concept lattice from a binary relation is a process of clustering, and the line diagram corresponding to a concept lattice vividly shows the generalization and specialization relationship among formal concepts. FCA can be combined with techniques in many fields like big data [19], Internet of Things [20–22], data mining [23], and social computing. If it is combined with deep learning methods to apply to the field of big data and edge computing [24–26], the accuracy of data mining and resource allocation will be optimized. If it is applied to social networks [27, 28], the relationship between nodes can be strengthened to a certain extent and the structure of social networks can be improved. To reduce the energy consumption of WSNs and achieve optimal deployment of nodes [29], this paper uses FCA to extract the minimum sensor combinations to monitor the entire area.

According to the research of existing methods, this paper proposes two algorithms based on FCA for extracting some MSCs. One is called implications-based algorithm for extracting MSCs. This method can find the minimum sensor combination, but the number of rules affects the efficiency of this algorithm. To reduce the rule set, we propose the second method for extracting the stem base of region implications from the concept lattice. The implementation process of these two algorithms can be roughly divided into three parts. Firstly, transform sensors and monitoring areas into a context. Secondly, extract the concept lattice from the context and implications among areas. Lastly, extract sensor nodes that monitor the overlapping area from the concept lattice. Thus, some strategies are designed to awaken different MSCs to reduce energy consumption and extend the lifetime of the network. Experimental results show that different MSCs can be extracted. These two methods have achieved good results in practical applications.

At present, the mechanisms for reducing energy consumption are broadly classified into the following three categories:

(1) Sleeping mechanism: This mechanism mainly solves the problem of energy waste caused by idle monitoring [30]. When there is no event of interest occurring around the node, the computing and communication module will be in the idle module, turning them off or turning them to a lower power, that is, the dormant state [31]. This mechanism is important for extending the lifetime of sensor nodes. For example, dynamic voltage scaling (DVS) and dynamic power management (DPM) manage system power in a power-aware manner to reduce energy consumption and extend the lifetime of the nodes.

(2) Data compression and data fusion mechanism: In WSNs with high coverage, there is redundancy in the information passed by neighbor nodes. And each node consumes too much energy to transmit data alone, shortening the lifetime of the network accordingly [32]. Data compression and fusion can effectively reduce the amount of raw data and the number of transmissions of sensor nodes. The data compression method reduces the energy consumption of nodes by compressing the transmitted data in advance [33]. Data fusion technology can merge the data information collected by multiple sensors and eliminate redundant and useless information [34, 35].

(3) Self-contained energy replenishment device: The sensor node is equipped with the energy harvesting and conversion device to collect energy like solar energy [36–38]. However, renewable energy fluctuates with the environment and the energy density is low. Therefore, this energy supplementation method has great limitations in practical applications. To solve these problems, the relevant research proposes to add the rechargeable sensor node of the wireless charging device to form a wireless rechargeable sensor network [39, 40]. And wireless power transmission (WPT) is used to replenish energy for rechargeable sensor nodes [41, 42].

However, there are still many disadvantages in the existing mechanisms. The sleeping mechanism requires an effective scheduling algorithm, and unreasonable arrangement of the dormant and working state of nodes will lead to idle waste. Besides, the transition between the dormant and working state of nodes also consumes some energy. Thus, frequent state transitions can also lead to excessive energy consumption. The data compression process requires more powerful processing power to handle compression algorithms [43]. Although data fusion can substantially reduce energy consumption, it usually needs specific nodes to collect data before proceeding to the next stage of transmission, which increases the network delay. Moreover, the process of data compression and fusion may cause the loss of detailed information and reduce the quality of data transmission. For wireless rechargeable networks, the current wireless charging technologies are mostly used in a short range. The farther the transmission distance, the lower the energy transmission efficiency. And the charging speed and efficiency of wireless charging still need to be further improved.

The main contributions of our work in this paper are summarized as follows: we propose two algorithms based on FCA for extracting some MSCs and waking up a group of sensor nodes by periodic loop mechanism and prediction mechanism to minimize the energy consumption on condition that the sensors monitor all areas. The specific idea is to extract MSCs by proposed algorithms and sort the combinations, and then put a group of sensor nodes in quasi-wake state by using the periodic loop mechanism. Finally, wake up the sensor nodes that are predicted to have a task schedule. The motivation diagram of this study is shown in Figure 1.

The remainder of this paper is organized as follows. Section 2 presents the related work, followed by some basic notations in formal concept analysis in Section 3. Section 4 provides two FCA-based algorithms called implications-based algorithm and the stem base-based algorithm for extracting different MSCs. Section 5 provides some experimental results to verify the validity and feasibility of the algorithms. Finally, we conclude and present the future work in Section 6.

2. Related Work

In recent years, with the rapid transformation and development of wireless communications in various fields [44, 45], a large number of researchers have been committed to studying various technologies that reduce the energy consumption of WSNs to extend the service life. In [46], the

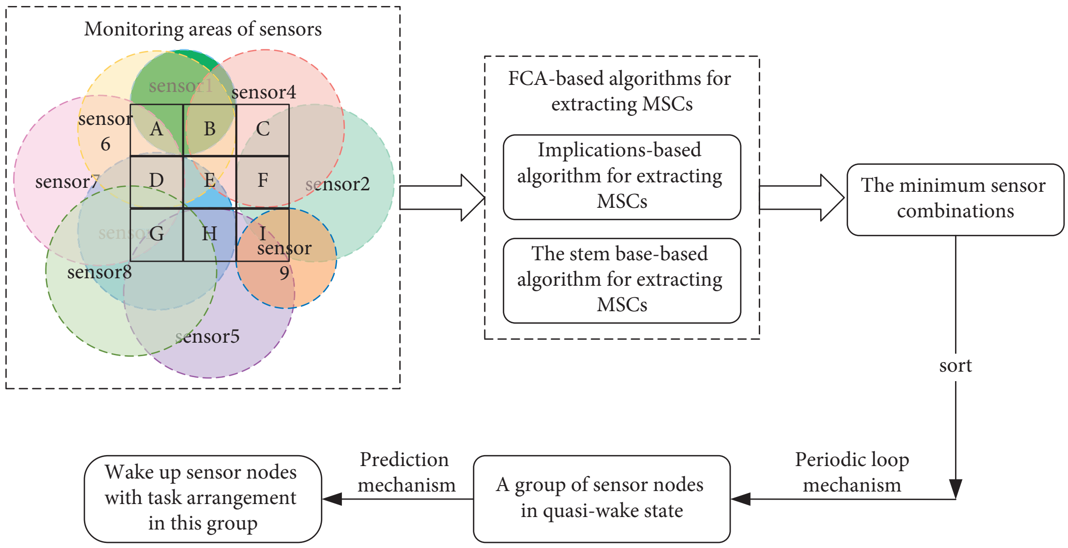


Figure 1: The framework of this study.

authors studied a data compression algorithm that determines the compression level of each sensor node to reduce total energy consumption based on the average energy level of neighboring sensor nodes. This technique reduces the energy consumption to transmit and receive packets and ultimately extends the entire network life. The literature [47] proposed a mean filtering algorithm based on node data images, which divides nodes into active and inactive nodes. This method clusters nodes and analyzes data obtained from sensor networks to eliminate data redundancy of sleep nodes. In [3], the authors introduced an area segmentation model based on optimal energy-saving constraints. According to the energy attribute of the sensor nodes, it optimally segments the coverage area of the wireless sensor network to improve its coverage, reduce the energy cost of a single node and realize the optimal networking of WSNs. To fully balance the energy, the literature [48] proposed to use power supply lines to connect nodes. Based on energy balance, a data transmission method with optimal hop count is proposed, which fully reduces the power consumption of data transmission. To determine the optimal monitoring sensor nodes and the information flow paths to the destination and receivers to reduce the energy consumption between the node and the receiver, Bat algorithm (BA) is proposed in literature [49]. In [2], the authors developed two different system models that use optimal node placement strategies compared with traditional equidistant placement strategies to minimize energy consumption in linear wireless sensor networks (LWSNs). Based on improved sparrow search algorithm (ISSA) optimized self-organizing maps (SOM), a cluster head selection strategy used in heterogeneous wireless sensor network (HWSN) is proposed in the literature [7]. This strategy comprehensively considers the residual energy, distance, and the times the node becomes a cluster head. It uses the adaptive learning of improved competitive neural networks to optimize the cluster head node and extend the life cycle of the network. The literature [50] proposed a power management method to reduce energy consumption in an idle state. Moreover, they studied a fine-grained power mode (FGPM) with five states. This mode can adjust the power consumption according to the communication state of the sensor nodes, reducing the power consumption of each node.

The strategies to wake up nodes will also affect the energy consumption of the sensor nodes, and an effective wake-up strategy can reasonably switch the sleep and working state of the sensors and reduce the energy loss of each sensor. In [51], a low duty cycle energy-efficient MAC protocol is proposed, which can adaptively update based on the prediction wake-up time of nodes. In this method, each node has a neighbor node information table including the target node address. By this table, the sending node will predict the wake-up time of receiving node. In [52], the authors proposed a self-adaptive sleep/wake-up scheduling approach based on the reinforcement learning technique. Each node can decide its working state independently in each time slot. The literature [53] found a multitarget sensing wake-up control method based on swarm intelligence, which studies the optimization of the wake-up strategy of dynamic targets to reduce the number of wake-up nodes when multiple targets are crossing. This method avoids waking up excessive nodes and extends the service life of WSNs.

3. Basic Notations in Formal Concept Analysis

In FCA [15], a (formal) context is defined as a triple $K = (G, M, I)$, where G and M are sets and $I \subseteq G \times M$ is a binary relation. The elements of G and M are respectively called objects and attributes. For any $r \in G$ and $m \in M$, $(r, m) \in I\,(\text{or}\, rIm)$ implies that the object r possesses the attribute m. For a set $X'IG$ of objects, X' is the set of attributes common to the objects in X. It is defined as fd1.

$$X' = \{m \in M: \quad \forall r \in X\,(rIm)\}. \tag{1}$$

Similarly, for a set $Y'IM$, Y' is the set of objects having all attributes in Y. It is defined as fd2.

$$Y' = \{r \in G: \quad \forall m \in Y\,(rIm)\}. \tag{2}$$

Given a context $K = (G, M, I)$, for any XG and Y M, the pair (X, Y) is called a (formal) concept if $X' = Y$ and $Y' = X$, where X and Y are respectively called the extent and the intent of the concept. There are two kinds of special concepts: object concept and property concept. Given an object $r \in G$, the pair (r'', r') is a concept, called the object concept of r. The object concept is the smallest concept having r in its extent. Correspondingly, given an attribute $a \in M$, the pair (a', a'') is also a concept, called the attribute concept of a, which is the greatest concept having a in its intent.

The set of all concepts in the context K is represented as $L(K)$. If (X_1, Y_1), $(X_2, Y_2) \in L(K)$ are concepts, then (X_1, Y_1) is called a subconcept of (X_2, Y_2), provided that $X_1 \subseteq X_2$ (which is equivalent to $Y_2 \subseteq Y_1$). The comparison expression is represented as $(X_1, Y_1) \leq (X_2, Y_2)$. In this case, (X_2, Y_2) is a superconcept of (X_1, Y_1). Therefore, we can get fd3.

$$(X_1, Y_1) \leq (X_2, Y_2) \Leftrightarrow X_1 \subseteq X_2 \Leftrightarrow Y_2 \subseteq Y_1. \tag{3}$$

The relation '$\leq$' is an order on $L(K)$, which is called the hierarchical order of the concepts. The hierarchical order produces a lattice structure in $L(K)$ called the concept lattice of the context $K = (G, M, I)$, which is also represented as $L(K)$. $L(K)$ is also a complete lattice in which infimum and supremum are given by: if T is an index set and for every $t \in T$, (A_t, B_t) is a concept, then the infimum of the set $\{(A_t, B_t): t \in T\}$ is defined as $\wedge_{t \in T}(A_t, B_t)$, which represents the largest common subconcept of the concept (A_t, B_t) $(t \in T)$. Accordingly, the supremum of the set $\{(A_t, B_t): t \in T\}$ represents the smallest common subconcept.

The labelling can be simplified considerably by putting down each object and attribute only once, namely at the circle for the respective object or attribute concept. Thus, the concept of lattices can be described by the line diagrams with reduced labelling. In a line diagram, the name of an object g is always attached to the circle that represents the smallest concept with g in its extent. And the name of an attribute a is always attached to the circle that represents the largest concept with a in its intent. This allows us to read the map I from the diagram: an object g has an attribute a if and only if there is an ascending path from the circle labeled by g to the circle labeled by a. The extent of a concept consists of all tuples whose labels are below in the diagram. The intent consists of all properties attached to the concepts above in the hierarchy. For example, in Figure 2 in section 5, the intent of the concept labeled by the attribute h is MSC, and its extent is {sensor – 4}. Similarly, the extent of the concept labeled by the sensor – 1 is {sensor – 1, sensor – 2, sensor – 3}, and its intent is $\{a, f\}$.

4. FCA-Based Algorithms for Extracting Different MSCs

In this section, we propose two FCA-based algorithms for extracting MSCs. Moreover, we will explain the specific algorithm flow of these two algorithms and strategies to reduce the energy consumption of sensors in detail. The algorithms consist of three steps as follows: the first step is to transform sensors and monitoring areas into a context, the second is to extract the concept lattice and implications among areas, and the third is to extract the minimum sensor combinations that can monitor all areas. The core idea of these two algorithms is that when the sensor set we have found covers all areas, end loop and output the minimum sensor set.

4.1. Transforming Sensors and Their Coverage Areas into a Context. In a wireless sensor network with many nodes, each sensor node can be considered as an object, and each area monitored by sensors can be considered as an attribute of the corresponding object. Assuming that some sensors are scattered in one area, we divide the area into some smaller areas. According to the positioning and operating radius of these sensors, the areas that can be monitored by each sensor can be obtained. Thus, we can transform sensors and monitoring areas into a table. The row represents the sensor number, the column represents the divided monitoring area, and the symbol "+" at the intersection indicates that the sensors can monitor the corresponding areas. For example, there are eight sensors, and the area is divided into eight smaller areas. Each sensor and corresponding monitoring area are as shown in Table 1, which is a context.

By constructing a concept lattice from the context, it is simple to see which sensor nodes are monitoring the overlapping area. Each sensor can perceive and record the surrounding sensors. Communication between sensors is achieved by finding all the sensor nodes that can be reached through the path on the concept lattice. Then we can make some strategies to awaken the corresponding sensors to achieve the purpose of reducing energy consumption and extending network life.

4.2. Implications-Based Algorithm for Extracting MSCs. The basic idea of this algorithm is to extract the region implications among regions from the concept lattice, and then extract different minimum sensor sets according to the rules. To describe this algorithm, we provide several definitions such as implication [15–18].

Definition 1. Given a context $K = (G, M, I)$, for any $A, B \subseteq M$, if every object with attributes in A also has the attributes in B, then $A \longrightarrow B$ is called an implication of K. A and B are respectively called antecedent and consequence of the implication.

Definition 2. Given a concept C, if $A \subseteq \text{Intent}(C)$, it matches an implication $A \longrightarrow B$.

Definition 3. Given a concept C, if C is both an attribute concept and an object concept, then it is called 1 – concept.

Definition 4. Given a concept C, if C is only an attribute concept, then it is called 2 – concept.

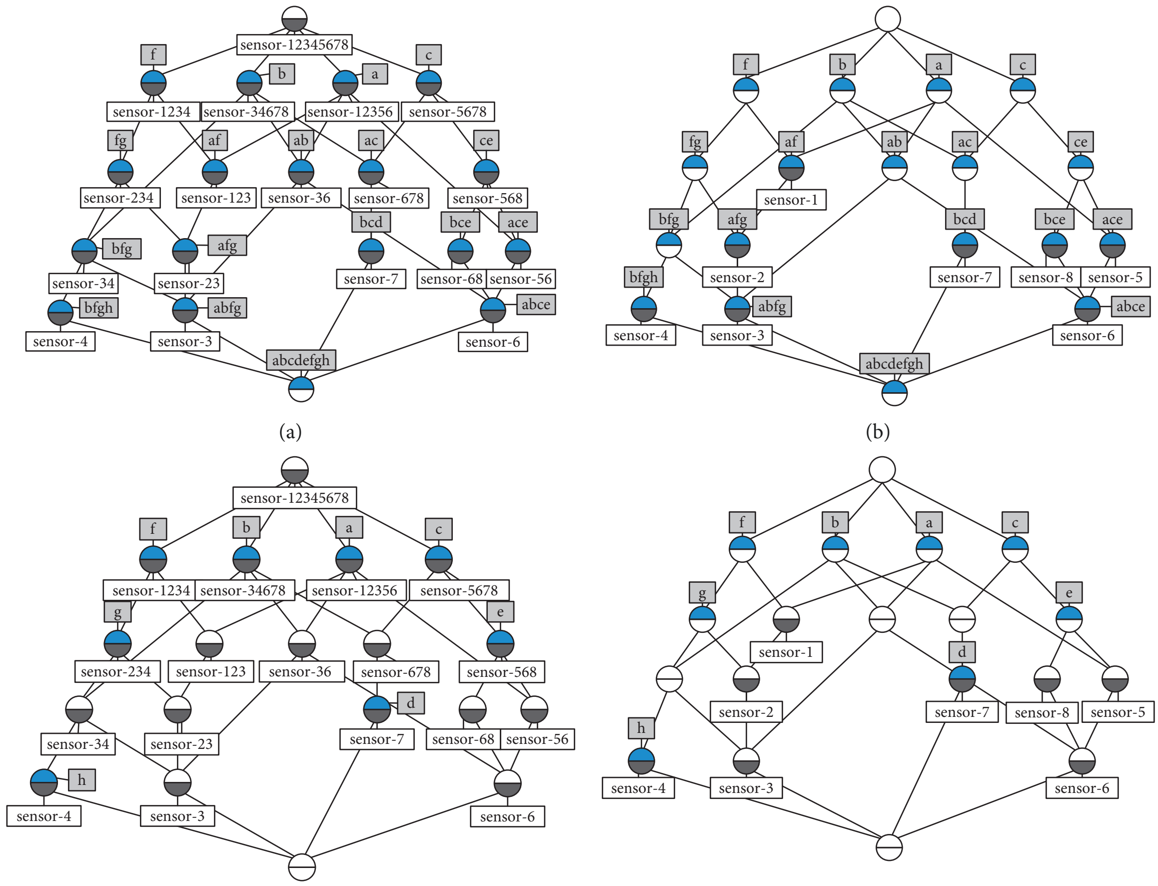


Figure 2: (a) Concept lattice for Table 1. (b) Line diagram with reduced objects labelling. (c) Line diagram with reduced attributes labelling. (d) Line diagram with reduced labelling.

Table 1: Monitoring area of each sensor.

	a	*b*	*c*	*d*	*e*	*f*	*g*	*h*
Sensor-1	+					+		
Sensor-2	+					+	+	
Sensor-3	+	+				+	+	
Sensor-4		+				+	+	+
Sensor-5	+		+		+			
Sensor-6	+	+	+		+			
Sensor-7		+	+	+				
Sensor-8		+	+		+			

Then we describe the algorithm flow of implications-based Algorithm 1 for extracting MSCs as below.

4.3. The Stem Base-Based Algorithm for Extracting MSCs. The basic idea of this algorithm is to extract the *stem base* of region implications from concept lattice, and then extract different minimum sensor sets according to the rules. To describe this algorithm, we provide several definitions such as the *stem base* of the attribute implications [15–18].

Definition 5. If each subset of M respects C also respects $A \longrightarrow B$, then the implication $A \longrightarrow B$ follows from a set C of implications between attributes.

Definition 6. If every implication follows from C, then a set of implications of a context is called complete.

Definition 7. If none of the implications follows from the others, then a set of implications of a context is called nonredundant.

Definition 8. $P \subseteq M$ is called the pseudointent of (G, M, I) if and only if $P \neq P'$ and $Q' \neq P$ holds for every pseudointent $Q \subset P$.

Theorem 1. *The set of implications* $C = \{P \longrightarrow P'' \mid P$ *is a pseudo − intent*$\}$ *is nonredundant and complete, which is called the stem base of the attribute implications.*

Then we define the stem base-based Algorithm 2 as follows:

4.4. Strategies for Reducing Sensor Energy Consumption. There are several ways to reduce the energy consumption of sensor nodes [54, 55], such as periodic awakening. One way is to make one node run out of its energy and then make the other overlapping nodes wake up. Another way is to calculate the size of the overlapping area of the nodes and decide whether they're awake or not.

In wireless sensor networks, there are the following ways to wake up the nodes:

(1) Full wake-up mode: In this mode, all nodes in the wireless sensor network wake up at the same time to detect and track targets in the network. Although this mode can obtain high tracking accuracy, it is at the cost of huge network energy consumption.

(2) Random wake-up mode: In this mode, the nodes in the wireless sensor network are randomly awakened by a given wake-up probability p.

(3) Select the wake-up mode by the prediction mechanism: In this mode, the nodes that have a greater return on tracking accuracy can be selectively awakened according to the needs of tracking tasks. Then predict the state of the target at the next moment through the information of this beat and wake up the nodes.

(4) Task cycle wake-up mode: In this mode, the nodes in the wireless sensor network are in the awake state periodically. Nodes in this mode can coexist with nodes in other working modes and assist these nodes to work.

In this study, we choose the period loop mechanism and prediction mechanism to wake up nodes. After the processing of the proposed algorithms, we can obtain different MSCs. Then sort the sensor combinations and use the period loop mechanism to put each group of nodes into quasi-wake state in order. The quasi-wake state is a state between wake-up and sleep, which means the node is ready to be awakened. The principle of the period loop mechanism is that only one group of sensor nodes is put into the quasi-wake state in each cycle until each group of nodes has been processed once, and then start again with the first set of nodes and repeat this process. Lastly, wake up the sensor nodes that may have a task arrangement by using a prediction mechanism. In this mechanism, the sending node calculates and predicts the wake-up time of receiving node by retaining the address of the target node in the neighbor node information table [51]. This step ensures the receiving nodes wake up timely and avoid the problems like collision caused by waking up all nodes simultaneously.

5. Simulation Results and Analysis

5.1. Extraction of Implications Rules from Table 1. In this section, we conduct some experiments with specific example to verify the effectiveness of the proposed algorithms. By using lattice-miner or ConExp, we can obtain the following concept lattices in Table 1, as shown in Figure 2. The concept lattice tools can be downloaded from the websites https://sourceforge.net/projects/lattice-miner/ and https://sourceforge.net/projects/conexp/.

From Figure 2(a), we can obtain the intents and extents of concepts in the concept lattice constructed according to the example. Furthermore, we conduct several experiments to simplify the labelling. Therefore, the concept of lattices can be described by the line diagrams with reduced labelling. By reducing objects labelling, the corresponding line diagram is shown as Figure 2(b). Likewise, the line diagram in Figure 2(c) is built by reducing attributes labelling. And the line diagram in Figure 2(d) is built by reducing labelling.

According to the concept lattices in Figure 2, we can obtain the concepts including the intents and extents of nodes in Table 2.

5.2. Extracting MSCs from Table 1 by Implications-Based Algorithm. We find out the implications and do some experiments to verify its effectiveness. The results including support and confidence are shown in Table 3.

From Figure 2(d), we can know that (*sensor-4*, h) and (*sensor-7*, d) are *1/2 concepts*, and (a', a''), (b', b''), (c', c''), (e', e''), (f', f''), (g', g'') are *2 concepts*. The specific steps of the implications-based algorithm are as follows:

Step 1. Initialization.

Let the concept set C = {(sensor-4, h), (sensor-7, d), (a′, a″), (b′, b″), (c′, c″), (e′, e″), (f′, f″), (g', g'')}, W = {rule set}, MWZ = empty.

Step 2. Cycle.

(1) First cycle: Select the first concept (sensor-4, h) from C. If h matches the 12th rule in W, then let $C = C/${sensor-4, h}, $M = M - \{b, f, g, h\} = \{a, c, d, e\}$, MWZ = {sensor-4};

(2) Second cycle: Select the first concept (sensor-7, d) from C. If d matches the 14th rule in W, then let $C = C/${sensor-7, d}, $M = M - \{b, c, d\} = \{a, e\}$, MWZ = {sensor-4, sensor-7};

(3) Third cycle: Select the first concept (e', e''). If e matches the 6th rule in W, then let $C = C/\{(e', e'')\}$, $M = M - \{e, c\} = \{a\}$. It is necessary to select one sensor from sensor-5, sensor-6 and sensor-8 covering e and put it into MWZ;

(4) Fourth cycle: Select the first concept (a', a''). If there is no matching rule, then let $C = C/\{(a', a'')\}$, $M = M - \{a\}$ = empty set. We just need to select one sensor from sensor-1, sensor-2, sensor-3, sensor-5 and sensor-6 covering a and put it into MWZ. If it has covered the area covered by the sensors in MWZ, then MWZ remains unchanged, otherwise the sensor will be added to MWZ;

Step 3. Let M = empty set. End loop and output MWZ. According to the analysis of the example, MWZ can be expressed by the following combination of sensor nodes:

MWZ = {sensor-4, sensor-7, sensor-5} or

{sensor-4, sensor-7, sensor-6} or

{sensor-4, sensor-7, sensor-8, sensor-1} or

{sensor-4, sensor-7, sensor-8, sensor-2} or

{sensor-4, sensor-7, sensor-8, sensor-3}.

```
Input: a context K = (G, M, I), where G and M is a set of sensors and domains respectively.
Output: the minimum sensor combinations that monitor all areas in M.
Generate the concept lattice with reduced labelling of K and implications in K
Let KC be the set of 1/2 − concept of K, KI be the set of implications in K, MSC = {.}, CE = {.}
If M ≠ {.}
Then For the first concept C in KC
If it matches the implication A ⟶ B
then M←M − A ⋃ B
KR←KR − {A ⟶ B}
KC←KC − {1 or 2 − concept generated from a|a ∈ A ⋃ B}
If C is a 1 − concept
then MSC←MSC ⋃ {x} // x is a sensor in the reduced extent of C
else for a sensor x in Extent(C)
if x ∈ CE
then MSC←MSC ∪ {x}
CE←CE ⋃ Extent(C)
Return 3
Output MSC
```

Algorithm 1: Implications-based algorithm for extracting MSCs.

```
Input: a context K = (G, M, I), where G and M is a set of sensors and domains, respectively.
Output: the minimum sensor combinations that monitor all areas in M.
Generate the concept lattice with reduced labelling of K and the stem base of implications
Let KC be the set of 1/2 − concept MSCs. Standalone K, KI be the stem base in K, MSC =, CE =
If M ≠
Then if there is an implication A ⟶ B satisfying M⊆ A⋃ B
then MSC←MSC ∪ {x} // x is a sensor in the support set of A ⟶ B
M =
else for the first concept C in KC
if it matches the implication A ⟶ B
then M←M − A ⋃ B
KR←KR − {A ⟶ B}
KC←KC − {1 or 2 − concept generated from a|a ∈ A ⋃ B}
If C is a 1 −concept
then MSC←MSC ⋃ {x} // x is a sensor in reduced extent of C
else for a sensor x in Extent(C)
if x ∉ CE
then MSC←MSC ⋃ {x}
CE←CE ⋃ Extent(C)
Return 3
Output MSC
```

Algorithm 2: The stem base-based algorithm for extracting MSCs.

The implications-based algorithm can find the minimum-maximum sensor combination, but a constant comparison is made between the selection and the rules. The number of rules affects the efficiency of the algorithm. Therefore, we will further reduce the number of rules in the rule set.

5.3. Extracting MSCs from Table 1 by Stem Base-Based Algorithm. In the same way, we conduct some experiments on the second algorithm. The implications and experimental results including support and confidence are shown in Table 4.

The specific steps of the stem base-based algorithm are as follows:

Step 4. Let the concept KC = {(*sensor-4*, h), (*sensor-7*, d), (a', a''), (b', b''), (c', c''), (e', e''), (f', f''), (g', g'')}, KI = {*The rules set*}, MWS = {}, M = {a, b, c, d, e, f, g, h}.

Table 2: Concepts in the obtained concept lattice.

Number	Extent	Intent
1	Sensor-12345678	Null
2	Sensor-12356	a
3	Sensor-34678	b
4	Sensor-1234	f
5	Sensor-5678	c
6	Sensor-123	af
7	Sensor-234	fg
8	Sensor-568	ce
9	Sensor-678	bc
10	Sensor-23	afg
11	Sensor-34	bfg
12	Sensor-36	ab
13	Sensor-56	ace
14	Sensor-68	bce
15	Sensor-4	bfgh
16	Sensor-3	abfg
17	Sensor-7	bcd
18	Sensor-6	abce
19	Null	abcdefgh

Table 3: Implications in Table 1.

Min. Support: 10%
Min. Confidence: 100%

ID	Implication	Support (%)	Confidence (%)
1	{g} ⟶ {f}	37.5	100
2	{a, c} ⟶ {e}	25	100
3	{a, e} ⟶ {c}	25	100
4	{b, f} ⟶ {g}	25	100
5	{b, g} ⟶ {f}	25	100
6	{e} ⟶ {c}	37.5	100
7	{a, g} ⟶ {f}	25	100
8	{a, b, f} ⟶ {g}	12.5	100
9	{a, b, g} ⟶ {f}	12.5	100
10	{a, b, c} ⟶ {e}	12.5	100
11	{a, b, e} ⟶ {c}	12.5	100
12	{h} ⟶ {b, f, g}	12.5	100
13	{b, e} ⟶ {c}	25	100
14	{d} ⟶ {b, c}	12.5	100

Table 4: The implications in Table 3.

Min. Support: 10%
Min. Confidence: 100%

ID	Implication	Support (%)	Confidence (%)
1	{g} ⟶ {f}	37.5	100
2	{a, c} ⟶ {e}	25	100
3	{b, f} ⟶ {g}	25	100
4	{e} ⟶ {c}	37.5	100
5	{h} ⟶ {b, f, g}	12.5	100
6	{d} ⟶ {b, c}	12.5	100

Step 5. Cycle.

(1) The first cycle: For any implication in Table 4, there is not $M \subseteq A \cup B$. Select the first concept ((*sensor-4*, h), h) from KC and h matches the fifth rule in W; Then let $KC = KC \backslash$ {(*sensor-4*, h), (b', b''), (f', f''), (g', g'')} = {(*sensor-7*, d), (a', a''), (c', c''), (e', e'')}, $M = M - \{b, f, g, h\} = \{a, c, d, e\}$, $KI = \{1, 2, 3, 4, 6\}$, MWZ = {*sensor-4*};

(2) The second cycle: For any implication in KI, there is not $M \subseteq A \cup B$. Select the first concept ((*sensor-7*, d), d) from KC, and d matches the sixth rule in Table 4; Then let $KC = KC \backslash$ {(*sensor-7*, d), (b', b''), (c', c'')} = {(a', a''), (e', e'')}, $M = M - \{b, c, d\} = \{a, e\}$, $KI = \{1, 2, 3, 4\}$, MWZ = {*sensor-4*, *sensor-7*};

(3) The third cycle: There is an implication $(a, c) \longrightarrow \{e\}$, which satisfies $\{a, e\} \subseteq \{a, c, e\}$. MWZ = {*sensor-4*, *sensor-7*, *sensor-5*} or {*sensor-4*, *sensor-7*, *sensor-6*}.

6. Conclusion

In wireless sensor networks, the energy storage and power supply capabilities of sensors are the main problems that limit the application of sensor network technology. Therefore, how to extend the service life of sensor networks and reduce the energy consumption of sensor nodes is a key issue in the research of wireless sensor networks. To solve this problem, this paper proposes two FCA-based methods to reduce the energy consumption of nodes, which are different from other commonly used methods. One is an implications-based algorithm for extracting MSCs. This method can find the minimum-maximum sensor combination, but it has to constantly compare the selection and the rules. The number of rules affects the efficiency of this algorithm. Therefore, we need to reduce the rule set. The second method is stem base-based algorithm, which can find the base of rules. These two methods are based on FCA, and both help solve the following problems. The first is how to choose a sensor. The second is how to optimize the sensor layout. And the third is how to wake up the corresponding sensors. According to experimental results, these two methods have achieved good results in practical applications.

The methods proposed in this paper are aimed at the research and discussion of the energy consumption of WSNs. The concept lattice of nodes in the wireless sensor network can express the relationship between nodes more effectively, and it is more helpful to choose nodes. With the deepening of research, we will propose more algorithms with concept lattice methods for other fields like radar sensor networks [56]. We believe there will be better effective solutions to such problems.

Data Availability

The data used to support this study are included within the article.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was funded by Shandong Province Teaching Reform Project (project no. S2018Z022).

References

[1] J. Huang, Z. Tong, and Z. Feng, "Geographical POI recommendation for Internet of Things: a federated learning approach using matrix factorization," *International Journal of Communication Systems*, 2022.

[2] A. Hussein, A. Elnakib, and S. Kishk, "Linear wireless sensor networks energy minimization using optimal placement strategies of nodes," *Wireless Personal Communications*, vol. 114, no. 4, pp. 2841–2854, 2020.

[3] X. Chen and T. Wu, "Region segmentation model for wireless sensor networks considering optimal energy conservation constraints," *Cluster Computing*, vol. 22, no. S3, pp. 7507–7514, 2019.

[4] Y. Chen, F. Zhao, Y. Lu, and X. Chen, "Dynamic task offloading for mobile edge computing with hybrid energy supply," *Tsinghua Science and Technology*, vol. 10, 2021.

[5] J. Huang, B. Lv, Y. Wu, Y. Chen, and X. Shen, "Dynamic admission control and resource allocation for mobile edge computing enabled small cell network," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 2, pp. 1964–1973, 2022.

[6] H. Yetgin, K. T. K. Cheung, M. El-Hajjar, and L. Hanzo, "A survey of network lifetime maximization techniques in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 828–854, 2017.

[7] L. Cao, Y. Yue, and Y. Zhang, "A data collection strategy for heterogeneous wireless sensor networks based on energy efficiency and collaborative optimization," *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–13, Article ID 9808449, 2021.

[8] Z. Zhang, J. Willson, Z. Lu, W. Wu, X. Zhu, and D.-Z. Du, "Approximating maximum lifetime k -coverage through minimizing weighted k -cover in homogeneous wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 24, no. 6, pp. 3620–3633, 2016.

[9] C. Catlett, P. Beckman, N. Ferrier et al., "Measuring cities with software-defined sensors," *Journal of Social Computing*, vol. 1, no. 1, pp. 14–27, 2020.

[10] J. Man, H. Dong, L. Jia, and Y. Qin, "GGC: gray-granger causality method for sensor correlation network structure mining on high-speed train," *Tsinghua Science and Technology*, vol. 27, no. 1, pp. 207–222, 2022.

[11] S. Yan, Z. Zhou, Y. Yang, Q. Leng, and W. Zhao, "Developments and applications of tunneling magnetoresistance sensors," *Tsinghua Science and Technology*, vol. 27, no. 3, pp. 443–454, 2022.

[12] X. Wan, J. Wu, and X. Shen, "Maximal lifetime scheduling for roadside sensor networks with survivability k," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 11, pp. 5300–5313, 2015.

[13] M. Kariman-Khorasani, M. A. Pourmina, and A. Salahi, "Energy balance based lifetime maximization in wireless Sensor networks employing joint routing and asynchronous duty cycle scheduling techniques," *Wireless Personal Communications*, vol. 83, no. 2, pp. 1057–1083, 2015.

[14] M. Yang, D. Kim, D. Li, W. Chen, and A. O. Tokuta, "Maximum lifetime suspect monitoring on the street with battery-powered camera sensors," *Wireless Networks*, vol. 21, no. 4, pp. 1093–1107, 2015.

[15] B. Ganter and R. Wille, *Formal Concept Analysis: Mathematical Foundations*, Springer Science & Business Media, Germany, 2012.

[16] J. Ma, W. Zhang, and Y. Qian, "Dependence space models to construct concept lattices," *International Journal of Approximate Reasoning*, vol. 123, pp. 1–16, 2020.

[17] T.-M. Liaw and S. C. Lin, "A general theory of concept lattice with tractable implication exploration," *Theoretical Computer Science*, vol. 837, pp. 84–114, 2020.

[18] W. Conradie, S. Frittella, K. Manoorkar et al., "Rough concepts," *Information Sciences*, vol. 561, pp. 371–413, 2021.

[19] A. K. Sandhu, "Big data with cloud computing: discussions and challenges," *Big Data Mining and Analytics*, vol. 5, no. 1, pp. 32–40, 2022.

[20] C. Chi, Y. Wang, X. Tong, M. Siddula, and Z. Cai, "Game Theory in Internet of Things: A Survey," *IEEE Internet of Things Journal*, 2021.

[21] M. Azrour, J. Mabrouki, A. Guezzaz, and Y. Farhaoui, "New enhanced authentication protocol for internet of things," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 1–9, 2021.

[22] Y. Chen, W. Gu, and K. Li, "Dynamic task offloading for Internet of Things in mobile edge computing via deep reinforcement learning," *International Journal of Communication Systems*, Article ID e5154, 2022.

[23] E. Shemis and A. Mohammed, "A comprehensive review on updating concept lattices and its application in updating association rules," *WIREs Data Mining and Knowledge Discovery*, vol. 11, no. 2, Article ID e1401, 2021.

[24] Y. N. Malek, M. Najib, M. Bakhouya, and M. Essaaidi, "Multivariate deep learning approach for electric vehicle speed forecasting," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 56–64, 2021.

[25] Y. Chen, Z. Liu, Y. Zhang, Y. Wu, X. Chen, and L. Zhao, "Deep reinforcement learning-based dynamic resource management for mobile edge computing in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 4925–4934, 2021.

[26] F. Li, X. Yu, R. Ge, Y. Wang, Y. Cui, and H. Zhou, "BCSE: blockchain-based trusted service evaluation model over big data," *Big Data Mining and Analytics*, vol. 5, no. 1, pp. 1–14, 2022.

[27] C. Jiang, A. D'Arienzo, W. Li, S. Wu, and Q. Bai, "An operator-based approach for modeling influence diffusion in complex social networks," *Journal of Social Computing*, vol. 2, no. 2, pp. 166–182, 2021.

[28] P. D. Waggoner, R. Y. Shapiro, S. Frederick, and M. Gong, "Uncovering the online social structure surrounding COVID-19," *Journal of Social Computing*, vol. 2, no. 2, pp. 157–165, 2021.

[29] H. Niu, Z. Chu, Z. Zhu, and F. Zhou, "Aerial intelligent reflecting surface for secure wireless networks: secrecy capacity and optimal trajectory strategy," *Intelligent and Converged Networks*, vol. 3, no. 1, pp. 119–133, 2022.

[30] L. Rajesh and C. B. Reddy, "Efficient wireless sensor network using nodes sleep/active strategy," *International Conference on Inventive Computation Technologies (ICICT)*, vol. 2, pp. 1–4, 2016.

[31] R. Wan, N. Xiong, and N. T. Loc, "An energy-efficient sleep scheduling mechanism with similarity measure for wireless sensor networks," *Human-centric Computing and Information Sciences*, vol. 8, no. 1, p. 18, 2018.

[32] P. Zhang, L. Li, K. Niu, Y. Li, G. Lu, and Z. Wang, "An intelligent wireless transmission toward 6G," *Intelligent and Converged Networks*, vol. 2, no. 3, pp. 244–257, 2021.

[33] C. J. Deepu, C.-H. Heng, and Y. Lian, "A hybrid data compression scheme for power reduction in wireless sensors for

IoT," *IEEE transactions on biomedical circuits and systems*, vol. 11, no. 2, pp. 245–254, 2017.

[34] D. Izadi, J. Abawajy, S. Ghanavati, and T. Herawan, "A data fusion method in wireless sensor networks," *Sensors*, vol. 15, no. 2, pp. 2964–2979, 2015.

[35] L. Qi, C. Hu, X. Zhang et al., "Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4159–4167, 2021.

[36] M. Getahun, M. Azath, D. P. Sharma, A. Tuni, and A. Adane, "Efficient energy utilization algorithm through energy harvesting for heterogeneous clustered wireless sensor network," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–17, 2022.

[37] T. V. Tran and W.-Y. Chung, "High-efficient energy harvester with flexible solar panel for a wearable sensor device," *IEEE Sensors Journal*, vol. 16, no. 24, pp. 9021–9028, 2016.

[38] C. Wang, J. Li, Y. Yang, and F. Ye, "Combining solar energy harvesting with wireless charging for hybrid wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 3, pp. 560–576, 2018.

[39] K. Li, W. Ni, L. Duan, M. Abolhasan, and J. Niu, "Wireless power transfer and data collection in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2686–2697, 2018.

[40] M. Zhang and W. Cai, "Data collecting and energy charging oriented mobile path design for rechargeable wireless sensor networks," *Journal of Sensors*, vol. 2022, pp. 1–14, 2022.

[41] S. He, J. Chen, F. Jiang, D. K. Y. Yau, G. Xing, and Y. Sun, "Energy provisioning in wireless rechargeable sensor networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 10, pp. 1931–1942, 2013.

[42] Z. Zhang, H. Pang, A. Georgiadis, and C. Cecati, "Wireless power transfer-an overview," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 2, pp. 1044–1058, 2019.

[43] T. A. Kohler, D. Bird, and D. H. Wolpert, "Social scale and collective computation: does information processing limit rate of growth in scale?" *Journal of Social Computing*, vol. 3, no. 1, pp. 1–17, 2022.

[44] M. Z. Siddiqi and T. Mir, "Reconfigurable intelligent surface-aided wireless communications: an overview," *Intelligent and Converged Networks*, vol. 3, no. 1, pp. 33–63, 2022.

[45] M. Jian, G. C. Alexandropoulos, E. Basar et al., "Reconfigurable intelligent surfaces for wireless communications: overview of hardware designs, channel models, and estimation techniques," *Intelligent and Converged Networks*, vol. 3, no. 1, pp. 1–32, 2022.

[46] S. Kim, C. Cho, K.-J. Park, and H. Lim, "Increasing network lifetime using data compression in wireless sensor networks with energy harvesting," *International Journal of Distributed Sensor Networks*, vol. 13, no. 1, Article ID 155014771668968, 2017.

[47] X. Fan, W. Wei, M. Wozniak, and Y. Li, "Low energy consumption and data redundancy approach of wireless sensor networks with bigdata," *Information Technology and Control*, vol. 47, no. 3, pp. 406–418, 2018.

[48] X. Liu and J. Wu, "A method for energy balance and data transmission optimal routing in wireless sensor networks," *Sensors*, vol. 19, no. 13, p. 3017, 2019.

[49] A. K. Sangaiah, M. Sadeghilalimi, A. A. R. Hosseinabadi, and W. Zhang, "Energy consumption in point-coverage wireless sensor networks via Bat algorithm," *IEEE Access*, vol. 7, pp. 180258–180269, 2019.

[50] S. You, J. K. Eshraghian, H. C. Iu, and K. Cho, "Low-power wireless sensor network using fine-grain control of sensor module power mode," *Sensors*, vol. 21, no. 9, p. 3198, 2021.

[51] D.-g. Zhang, S. Zhou, and Y.-m. Tang, "A low duty cycle efficient MAC protocol based on self-adaption and predictive strategy," *Mobile Networks and Applications*, vol. 23, no. 4, pp. 828–839, 2018.

[52] D. Ye and M. Zhang, "A self-adaptive sleep/wake-up scheduling approach for wireless sensor networks," *IEEE Transactions on Cybernetics*, vol. 48, no. 3, pp. 979–992, 2018.

[53] J. Qi, L. Pan, S. Ren, F. Chang, and R. Wang, "SMTS: a swarm intelligence-inspired sensor wake-up control method for multi-target sensing in wireless sensor networks," *Wireless Networks*, vol. 26, no. 5, pp. 3847–3859, 2020.

[54] J. Huang, C. Zhang, and J. Zhang, "A multi-queue approach of energy efficient task scheduling for sensor hubs," *Chinese Journal of Electronics*, vol. 29, no. 2, pp. 242–247, 2020.

[55] A. Varmaghani, A. Matin Nazar, M. Ahmadi, A. Sharifi, S. Jafarzadeh Ghoushchi, and Y. Pourasad, "DMTC: optimize energy consumption in dynamic wireless sensor network based on fog computing and fuzzy multiple attribute decision-making," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–14, Article ID 9953416, 2021.

[56] M. R. Khosravi and S. Samadi, "Mobile multimedia computing in cyber-physical surveillance services through UAV-borne video-SAR: a taxonomy of intelligent data processing for IoMT-enabled radar sensor networks," *Tsinghua Science and Technology*, vol. 27, no. 2, pp. 288–302, 2022.

Hindawi
Scientific Programming
Volume 2022, Article ID 6129150, 14 pages
https://doi.org/10.1155/2022/6129150

Research Article

Optimized Bandwidth Allocation for MEC Server in Blockchain-Enabled IoT Networks

Shengcheng Ma[1], **Wei-Tek Tsai**[1,2], **Shuai Wang**[1], **Yan Liang**[3], **and Dong Yang**[1]

[1]*Beihang University, No. 37 Xueyuan Road, Beijing 100191, China*
[2]*Digital Society & Blockchain Laboratory, No. 37 Xueyuan Road, Beijing 100191, China*
[3]*Beijing Information Science and Technology University, No. 35 the 4th Ring Middle Road, Beijing 100101, China*

Correspondence should be addressed to Shengcheng Ma; mashengcheng@163.com

Received 22 March 2022; Revised 20 April 2022; Accepted 29 April 2022; Published 16 May 2022

Academic Editor: Jiwei Huang

Powered by the development of the fifth-generation mobile communication technology (5G), the Internet of things (IoT) has been widely applied in people's life. Due to the limitation of storage and computing power, the data transmission of IoT devices faces challenges in terms of security and privacy. Therefore, many researchers provide the conjunction of blockchain and mobile edge computing (MEC) to make up for the lack of computing and security. MEC can meet the storage and computing requirements for IoT devices. Blockchain can provide a decentralized, antitamper solution that can help devices overcome security deficiencies, whereas the speed of blockchain communication is not fast enough because of the consensus mechanism. In this article, we focus on the permissioned blockchain and propose an optimized bandwidth allocation algorithm to promote the performance of consensus communication. The algorithm contains an In-Network control ideology and supports deployment on MEC servers. Deep reinforcement learning (DRL) is employed to perform the computation of available bandwidth in our scheme. We implement a prototype system in the testbed and perform a simulation, and the results show the advantages compared with the current widely used algorithm. By applying our method, the Internet of things devices can transmit data safely and efficiently.

1. Introduction

With the increasing popularity of the 5G, the IoT has been widely applied in people's life. 5G network provides infrastructure for the application of IoT, such as smart city [1], Internet to Vehicle (IoV) [2], and health monitoring [3]. However, the data generated by the Internet of things devices has security and privacy challenges. The devices only have limited storage and computing capability, which makes it difficult to ensure the safety of the data transmission.

To meet these challenges, a fusion paradigm of blockchain and MEC is proposed to strengthen the transmission of IoT [4, 5]. Blockchain can improve the security for IoT devices, and MEC offers the computing and storage resources for blockchain network. MEC is a distributed system deployed at the edge of network to process the offloading tasks from the mobile devices. Applying MEC can bring many benefits, including decreased transmission delay [6], provided more computing power [7], improved user experience [8], and reduced the energy cost [9]. Because of sinking the computing and storage resources close to the end devices, it is possible to improve the security of IoT data transmission by using blockchain.

Blockchain as a distributed network with decentralized ledger is widely used in many scenarios such as finance, e-business, and IoT. Every node on the blockchain reaches a consensus for every transaction, and each block is dependent on the previous one in a cryptographic hash way. All blocks are linked by hash timestamp like a chain. Attributed to these structures and organizations, blockchain could prevent data from being tampered with [10] and provide transactions with a traceable path [11].

Although blockchain brings many benefits to applications, it consumes huge resources to ensure data consistency.

All peers need to synchronize the information when a new block is generated. This will lead to communications between all nodes on the blockchain. In particular the classic Byzantine Fault Tolerance (BFT) consensus algorithm used by the permissioned blockchain causes too many communications among the peers [12]. When the number of peers on the blockchain is N, the communication boundary of this algorithm will be $O(N^2)$ [13]. Therefore, the performance of consensus in the blockchain is an urgent problem to solve.

To improve the performance of blockchain, many platforms attempt to optimize the process of the transaction. Some platforms add a database layer above the blockchain to speed up the operation of transactions [14]. And some researchers try to simplify the consensus mechanism to reduce the calculated workload [15]. However, these methods sacrifice parts of consistency to improve performance.

In this paper, we focus on the permissioned blockchain which employs BFT based consensus mechanism and propose an optimized bandwidth allocation strategy to improve the performance. The permissioned blockchain usually does not use Proof of Work (PoW) [16] or Proof of Stake (PoS) [17] as the consensus protocol. Therefore, the transaction process in this kind of blockchain depends more on the network. The performance of the network can directly affect the performance of the blockchain. Then it affects the data transmission of IoT network. In addition, the permissioned blockchain can be customized, so the deployment of the permissioned blockchain is easier than public blockchain on MEC servers. Hence, we consider the permissioned blockchain as the target scenarios and optimize the network to improve the performance of BFT consensus communication.

Our main contributions of this paper can be summarized as follows:

(i) First, we design blockchain platform in the MEC servers which can strengthen the IoT communication. And we analyze some current bandwidth congestion control algorithms and provide an available bandwidth notification (ABN) algorithm. The ABN algorithm combines the advantages of Bottleneck Bandwidth and Round-trip propagation time (BBR) and explicit congestion notification (ECN).

(ii) Second, according to the different deployment methods, we consider the application scenarios of private network and public network, respectively, and use the deep reinforcement learning method to solve the bandwidth allocation method in the public network with the competition.

(iii) Third, our prototype has run in the testbed and the results show the improvement compared with current widely used techniques.

The rest of this paper is organized as follows: Section 2 summarizes related work of blockchain optimizations. Next, we present a blockchain-enabled IoT network model to optimize the performance of consensus communication in Section 3. In particular, we illustrate the architecture and implementation of our new method in this section. Then, we provide the solution of bandwidth allocation for different scenarios in Section 4. In Section 5, we introduce the experimental environment and analyze the result to prove the improvement of performance. Finally, we conclude this paper in Section 6.

2. Related Work

2.1. Cost of Blockchain. The discussion concerning the cost of blockchain has always been a hot topic. Catalini and Gans consider the verification cost and network cost by blockchain technology from an economics perspective [18]. Sukhwani et al. use Stochastic Reward Nets (SRN) to simulate the PBFT consensus process [19] and investigate why the process could be a performance bottleneck in the network with a large number of peers. In their SRN model, conclusion is that the transmission delays can impact the average time to consensus. Marko from IBM discusses PoW and BFT blockchains focusing on overcoming and scalability [20]. In his conclusion, the performance of PoW-based blockchain depends on the computational power and scale of nodes.

In addition, scalability is a weakness in this kind of blockchain. A review on the cost of IoT application based on blockchain is presented in [4], which concludes that the reduction in overhead is carried out by removing the PoW consensus mechanism and choosing the right cryptographic scheme.

Sekaran et al. research the integration of blockchain and IoT technologies [21]. By addressing the shortcomings and limitations of IoT, they summarized the state of the art of high-level solutions. Li et al. propose a blockchain-based data security scheme for IoT networks in 6G [22]. The fusion technology of AI and blockchain is developed to evaluate and optimize the quality of service in IoT networks.

2.2. Attempts to Promote the Performance of Blockchain. Many researchers and companies have made efforts to improve performance and reduce costs in the blockchain. Some researchers analyze the relationship between performance and storage structure, and they change the size of block or database to optimize the blockchain performance [23, 24]. Sharding is an interesting technology that can promote the parallel processing capability of the blockchain. Choosing an optimal shard size or employing a suitable sharding protocol can maximize performance and improve scalability [25, 26].

2.3. Optimization from the Perspective of Network. With respect to performance optimization for blockchain, the consensus algorithm evolves many branches. As an infrastructure, the increased network throughput will benefit all types of blockchain. Many researchers applied Artificial Intelligence (AI) to optimize the performance for IoT network. Huang et al. [27] studied the federated learning with the Matrix Factorization method and optimized the recommendation system for IoT network. Chen et al. [5] dealt

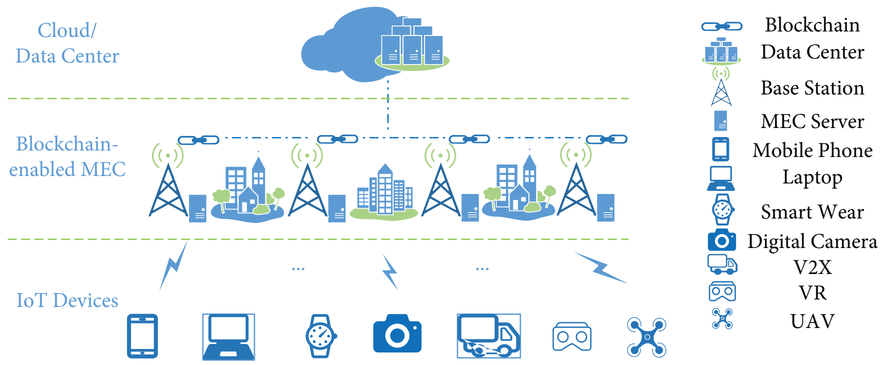


Figure 1: Blockchain-enabled IoT networks.

with task offload problem and used DRL method in MEC to improve the performance of IoT. The authors of [28] improved the performance of blockchain by enabling distributed NFV-MANO framework. Huang et al. proposed the joint admission control and computation resource allocation in the MEC server [29]. Their method reached the balance tradeoff between the utility and the queue length in small cell network. Huang's team proposed a BoR (Blockchain over RDMA) framework that can reduce the consuming time when a new node joins the blockchain network [30]. However, these methods need to add some hardware to ensure network achievement. This will lead to a sharp rise in network deployment costs. Jointly considering the cost and compatibility, we plan to optimize the congestion control of TCP in network to enhance the performance.

3. System Model

To illustrate our system, we first describe the network application scenarios. We classify the network flows according to the requirements in the scenarios. Then we build a model to represent the system operation mechanism and provide an optimal goal for blockchain consensus communication. Finally, we use the deep reinforcement learning method to optimize the bandwidth for blockchain business and introduce how feedback mechanism can avoid congestion.

3.1. Network Application Scenarios. We consider a blockchain-enabled IoT network, which consists of three layers, the IoT layer, the blockchain and MEC layer, and the cloud/data center layer. As shown in Figure 1, permissioned blockchain platform is deployed on the MEC servers, because MEC servers provide sufficient storage and computing capacities. A typical communication in this scenario is that the IoT devices transmit data to cloud server or data center via MEC servers. MEC servers deal with a large amount of offloading task from IoT devices. To enhance the security of data from IoT, blockchain collects the data as transactions from MEC servers. All blockchain nodes communicate with each other and participate in consensus. After the consensus is achieved, the transaction will be uploaded to blockchain as a new block. Then, the IoT data is protected against tampering.

To simplify the problem, we define three types of network elements according to network behavior in our system. *Sender*. The network node that sends data is defined as the sender. It is similar to a mobile phone in an IoT network. *Receiver*. The network node receiving data becomes the receiver, such as data center or cloud service. *Middlebox*. The network node that forwards data and has a buffer capacity is called middleboxes. It resembles the MEC servers in this scenario. This scenario can be shown as Figure 2.

The sender and the receiver can be converted to each other according to the data transmission direction. In Figure 2, senders sends data to receiver through middlebox and maybe many other middleboxes. This scenario can represent all network transmission forms.

In this scenario, the efficiency of data transmission is subject to these three types of nodes. The sender's congestion control strategy affects the sending speed. The middlebox's forwarding speed and buffer length determine whether it will become a bottleneck in the path. The receiver's feedback information ACK controls the sender's available windows, and it also decides whether retransmission is needed. Moreover, different types of data flows will also influence the utilization of transmission.

3.2. Network Flow Categories. According to the characteristics of connection, we classify network flows into three categories. The first type is fat flow; it usually is used for a large amount of data transmission with high bandwidth, for example, video streaming, large file download, and so on. The second type is slim flow; it is used for applications that send fewer data and close quickly. The most typical instance is web browsing and blockchain consensus communication. The third type is control flow; it is the ACK from receiver to sender. These three types of flows share the bandwidth of the network.

Each direction in middlebox has a specific bandwidth. The middlebox locates in a critical position when many flows go through it. It can be a bottleneck if the data inflow is greater than its processing capacity. Normally, the forward performance of the middlebox limits the receiving ability. If a great number of flows coming from senders fill the buffer of the middlebox, the effective bandwidth of the middlebox

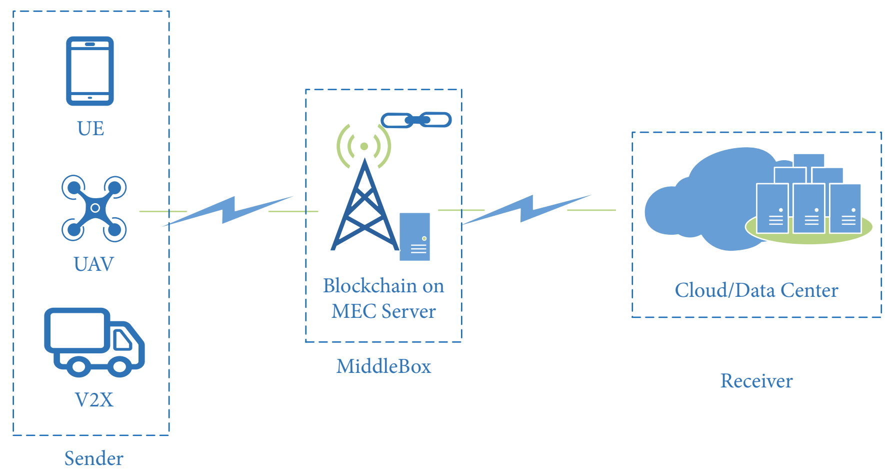


Figure 2: Three types of network elements.

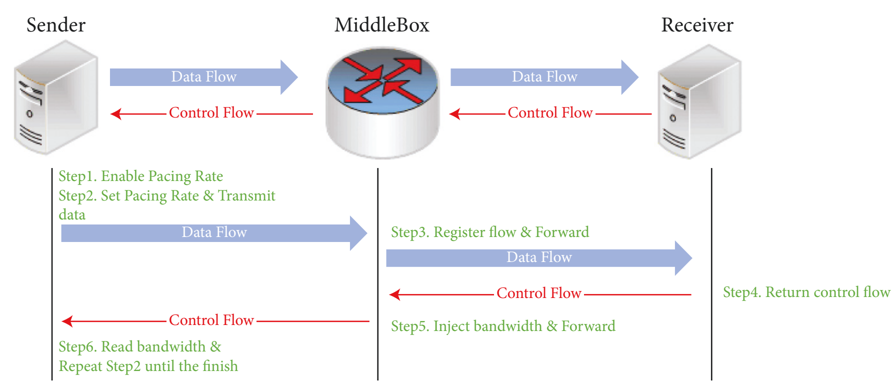


Figure 3: The process of available bandwidth notification.

will be exhausted. Then, congestion will form and delay will increase.

In order to avoid congestion, we propose a new method called available bandwidth notification.

3.3. Available Bandwidth Notification. Our available bandwidth notification is different from the existing methods. The traditional network is the end-to-end model. The senders do not understand the situation in the network. BBR's estimates of the fixed bandwidth and min RTT to operate congestion are like a guess [31]. Even the ECN method only tells the sender that congestion happens [32]. The sender does not know what speed is suitable, so it decreases the congestion window to the lowest value. The Software Defined Network (SDN) can organize the network topology and learn about the status of the network, but it needs a controller device and the standards of various manufacturers have not yet been unified. The SDN has not been widely used in many realistic production environments. Our method is designed to deploy in the traditional network for wider applicability; furthermore, it breaks through the end-to-end mode.

In our ABN method, we let middleboxes participate in congestion control. Middlebox can sense the real condition of the network in the self-system. All types of flows go through the middlebox, and the bandwidth utilization and queueing status can be read by the system of the middlebox. Then, the middlebox can calculate the most appropriate bandwidth value according to this information. With the help of the control flow returned from the receiver, the bandwidth value is put into control flow by the middlebox and delivered to the sender. When the sender receives the control flow, it will adjust the send window and sequence number by ACK information. Besides, it will set its pacing rate according to the value notified by the middlebox. In this way, the sender can adjust the sending speed in time conforming with the frequency of ACK return. The whole process of our available bandwidth notification method is shown in Figure 3. The details of operation steps are as follows.

(i) Step 1. The sender enables the pacing rate system setting. That means the sender will transmit data at a smooth setting speed without burst.

(ii) Step 2. The sender reads the pacing rate and transmits data. The pacing rate will be set to the initial value.

(iii) Step 3. The middlebox senses the data flow from the sender and forwards the data flow to the receiver. At the same time, the middlebox registers the flow information for available bandwidth calculation.

(iv) Step 4. The receiver accepts the data flow and returns the ACK information by the control flow.

(v) Step 5. The middlebox searches the registration information according to the five-tuple of the control flow and finds out which one is the corresponding sender. It injects the available bandwidth value into control flow after calculation, and it forwards the flow to the sender.

(vi) Step 6. The sender receives the control flow and parses the available bandwidth value from it and then repeats the operation in Step 2. The sender sets the pacing rate according to the bandwidth value and continues to send until the transmission task is accomplished.

After such a series of operations, the sender can reasonably arrange the sending speed according to the actual situation in the network. Because the situation of the network is as feedback from the devices inside the network, the devices inside the network participate in congestion control, which breaks the end-to-end mode. Next, we will analyze how the devices in the network, the middlebox, calculate the available bandwidth.

3.4. Bandwidth Allocation Model. The total bandwidth of a middlebox is a fixed value. This is determined by the service of ISP (Internet Service Provider) and the performance of equipment itself. In this fixed bandwidth, there is a part of the traffic that belongs to non-TCP protocol. It is not affected by the congestion avoidance algorithm, so our method ignores the bandwidth of this part of the data flow.

According to our previous classification of flow, control flow, fat flow, and slim flow all belong to TCP protocol. Although the control flow belongs to TCP protocol, the timely transmission of control flow is very important to restore the sender's window which improves the efficiency of the whole network. Therefore, this kind of flow should be given high priority to obtain the required bandwidth.

After removing the bandwidth occupied by non-TCP traffic and control flow, the remaining bandwidth of the middlebox is shared by fat flows and slim flows competitively. Our method should reasonably allocate the remaining bandwidth, calculate a suitable bandwidth for senders, and ensure that slim traffic representing blockchain communication gets better service.

In our bandwidth allocation model, the whole bandwidth of the middlebox is defined as B_{all}. The non-TCP traffic obtains the bandwidth presented as G, and the control flow's bandwidth is presented as C. Excluding these two parts of bandwidth, the actual bandwidth that can be allocated is B. And B can be derived as (1).

$$B = B_{\text{all}} - G - C. \tag{1}$$

In the middlebox, the number of flows changes with time. Combined with the strategy of middlebox adjustment,

Table 1: Flow information table.

Item	Description
src_ip	Source IP address
dst_ip	Destination IP address
src_port	Source port
dst_port	Destination port
Timer	Number of occupied timer cycles
data_inflow	The variable shows data inflow
Rate	The real transmit rate of flow
Count	The number of flows

we define a timer t_i to represent the period of bandwidth adjustment, and $i \in \{1, \ldots, N\}$. In order to obtain a steady-state of the system operation, we define the observation time as T, and it contains N timers which can be expressed as

$$T = \sum_{i=1}^{N} t_i. \tag{2}$$

When a new flow comes in, the middlebox will register the information of this flow on the flow table. After the timer t_i expires, the middlebox checks the flow table. If there is no data belonging to a flow, it is considered that this flow already has been processed. Consequently, the total number of flows stored in the flow table will be reduced by the number of processed flows. The detail of the flow table is shown in Table 1.

Based on the previous classification, the B bandwidth is allocated for fat flow and slim flow. We denote fat flow as a set $F_j = \{m_j, d_j, r_j\}$, $j \in \{1, \ldots, N^F(t)\}$, and $N^F(t)$ is the total number of fat flows through the middlebox in time phase t. It represents a single large amount of data transmission in the middlebox. In this set, m_j denotes the whole data size needed to transmit in flow F_j, d_j denotes the maximum tolerable time to complete the transmission, and r_j denotes the reward for completing transmission of this flow, which means the system should finish the work of the j th flow within d_j time.

Similar to the fat flow, the slim flow is defined as $S_k = \{m_k, d_k, r_k\}$, $k \in \{1, \ldots, N^S(t)\}$, where m_k is the data size of the flow S_k, d_k is the time requirement of this flow, and r_k is the reward of finishing this flow. These variables represent the same meaning as fat flow. The $N^S(t)$ is the total number of slim flows through the middlebox in time slot t. According to the actual situation, the value of data size m_k in the slim flow may be substantially smaller than in the fat flow.

For our method in the middlebox, the most critical work is to allocate bandwidth to each flow. The bandwidth allocated to the flow will change with the timer, so we consider the bandwidth of the flow as a function. We denote $B_j^F(t)$ as the bandwidth which allocated for fat flow F_j during phase t. The same to the slim flow, we use $B_k^S(t)$ to represent the bandwidth of slim flow S_k in time slot t.

The job of our approach is to find the available bandwidth for all flows through the middlebox. The available bandwidth for each flow should meet the processing time requirement and be limited by the whole actual bandwidth

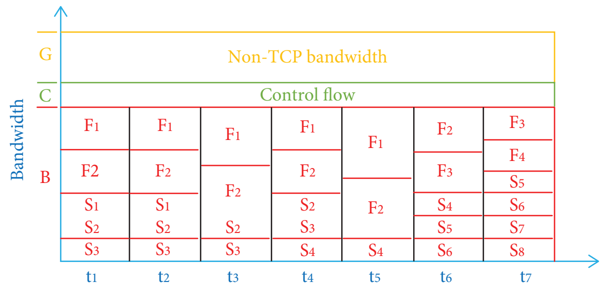


Figure 4: Bandwidth allocation for each flows.

of the middlebox B. At time slot t_i, we assume that the middlebox has $N^F(t_i)$ fat flows and $N^S(t_i)$ slim flows. The total fat flow's bandwidth can be denoted as $\sum_{j=1}^{N^F(t_i)} B_j^F(t_i)$, and all the slim flow's bandwidth can be denoted as $\sum_{k=1}^{N^F(t_i)} B_k^S(t_i)$. The holistic bandwidth of the flows can not be beyond the B of the middlebox, which can be represented by

$$\sum_{j=1}^{N^F(t_i)} B_j^F(t_i) + \sum_{k=1}^{N^S(t_i)} B_k^S(t_i) \leqslant B. \tag{3}$$

The timer of bandwidth regulation is a fixed value, so every time period is the same in our system. We assume that a slim flow S_k has completed transmission in X_k^S time periods, and the bandwidth allocated for S_k in each time slot t_i is denoted as $B_k^S(t_1), B_k^S(t_2), \ldots B_k^S(t_{X_k^S})$.

We make an instance to explain our model for bandwidth allocation. As shown in Figure 4, there are two fat flows (F_1, F_2) and three slim flows (S_1, S_2, S_3) in slot time t_1. We can derive that $N^F(t_1) = 2$ and $N^S(t_1) = 3$ in phase t_1. The height of each block represents their bandwidth at time t_1, so the height of S_1 equals $B_1^S(t_1)$ in time t_1. For example, in Figure 4, the slim flow S_1 takes two time phases to finish the data transmission; hence $X_1^S = 2$ in this flow.

For a data flow, the time to complete all the data transmission must be less than the required time. Bandwidth multiplied by time represents the amount of data that can be transmitted during this period of time. Then in the whole process, the total processing capacity of allocated bandwidth for slim slow S_k can be expressed as

$$P_{S_k} = \sum_{i=1}^{X_k^S} B_k^S(t_i) \cdot t_i, \tag{4}$$

where P_{S_k} is the total data size that the allocated resource can handle. P_{S_k} should be larger than the required data size of slim flow, which can be represented as

$$P_{S_k} \geqslant m_k. \tag{5}$$

The total processing time should be less than the required time of slim flow, which can be represented as

$$\sum_{i=1}^{X_k^S} t_i \leqslant d_k. \tag{6}$$

In order to optimize blockchain communication, we use slim flow to represent the network traffic in blockchain applications. Under these constraints, the object of our bandwidth allocation strategy is to serve as many slim flows as possible. In our model, we assume that the system will get a certain reward when a flow task completes. In the T period of observation, the situation of completed flow can be denoted as a Boolean array $I = \{1, 1, 1, 0, \ldots, 0\}$, and the rewards of all fat flows R^F can be represented as

$$R^F = \sum_{j=1}^{M} r_j \cdot I_j^F, \tag{7}$$

where M is the total number of fat flows that go through the middlebox during T time, I_j^F is the status of fat flow F_j, the situation $I_j^F = 1$ means the flow has been transmitted, and the situation $I_j^F = 0$ means the flow has not been served or has not finished. The rewards of all slim flows R^S can be represented as

$$R^S = \sum_{k=1}^{L} r_k \cdot I_k^S, \tag{8}$$

where L is the total number of fat flows, and the service status of slim flows can be represented by I^S.

The object of our model is to maximize rewards by allocating bandwidth to serve more data flows. By integrating all the limited conditions, our bandwidth allocation model can be defined as follows:

$$\begin{aligned}
\mathbf{P1}:&\ \max R^F + R^S,\\
s.t.\ C1:&\ P_{F_j} \geqslant m_j,\\
C2:&\ P_{S_k} \geqslant m_k,\\
C3:&\ \sum_{i=1}^{X_j^F} t_i \leqslant d_j,\\
C4:&\ \sum_{i=1}^{X_k^S} t_i \leqslant d_k,\\
C5:&\ \sum_{j=1}^{N^F(t_i)} B_j^F(t_i) + \sum_{k=1}^{N^S(t_i)} B_k^S(t_i) \leqslant B,
\end{aligned} \tag{9}$$

where $P1$ represents the goal of our model which means the rewards from fat flows and slim flows. Constraints $C1$ and $C2$ represent that the amount of transmission capability provided by allocated bandwidth should be greater than the amount of data required by the flow itself. $C1$ is for fat flows, and $C2$ is for slim flows. Constraints $C3$ and $C4$, respectively, represent the cost time of flow transmission within the required time. The constraint $C5$ shows the bandwidth division. In every certain time slot, the sum of the bandwidth of fat flows and slim flows should be less than the actual available bandwidth of the middlebox.

The design of this model considers blockchain communication in a general network environment. In the shared network environment, blockchain data flows compete with other network flows for bandwidth. We use slim flow to represent blockchain communication in our model, because slim flow conforms to the properties of consensus communication in blockchain applications. It is a type of frequent data communication with small bandwidth. The communication of blockchain needs to obtain enough bandwidth resources to ensure the timeliness of consensus. Consequently, we define a fat flow to simulate the competition between other network flows and blockchain data flows in our model.

4. Solution and Optimization for Blockchain Communication

In order to optimize the performance of blockchain communication, we propose a model of available bandwidth allocation. By solving this model, we can accelerate the speed of blockchain network traffic. For our bandwidth allocation model, we have two types of network environments to consider. The first case is the private network, where the consensus nodes of blockchain are connected by private networks. The second case is the public network, where the data of the blockchain is transmitted in a competitive environment. We will analyze these two cases respectively.

4.1. Private Network. For the private network, all the network resources are used for blockchain services. It usually is deployed by consortium blockchain or private blockchain.

In this type of network, all the data flows are blockchain communications. It means that these flows do not need to share bandwidth with other services. In our model, if there are only slim flows, this situation can be represented. All the slim flows should guarantee quality of service. The rewards of these flows are the same, and the bandwidth is exclusively used by slim flows without the competition of fat flow.

The solution of resource allocation in a private network can be simplified as average allocation. The average allocation method is to equally allocate the current network bandwidth to the flows through the middlebox.

Although it is an average allocation, there are still three factors that can change the allocated bandwidth. The first factor is the incoming of the new flows in the middlebox. The second factor is that the leaving flows after the transmission are finished. The third factor is the urgency of flow that needs to complete the service in time.

For the first factor, the flow information table updates when a new flow comes into the middlebox. The current total number of flows will increase. This will result in less bandwidth allocated to each flow. For the second factor, once a flow is finished, there is no more data belonging to this flow coming into the middlebox. Then the flow information table will delete the information of this flow, and at the same time, it will reclaim the bandwidth occupied by this flow. The reclaimed bandwidth is allocated to other data flows; this will increase the bandwidth to each flow. The third factor is a special case, which occurs only when the flow is about to time out. The system will allocate enough bandwidth to this flow so that it can accomplish the data transmission in the required time.

Combined with these three factors, we propose a bandwidth allocation algorithm based on the private network. Our priority of consideration is the third factor. To find out the flows that are about to time out, we look up the flow information table. The value of the timer in table X_k^S is the number of time slots that has been spent by this flow. The variable d_k is the required time slots of the flow k. If $d_k - X_k^S$ equal to 1, that means this flow has only one time slot to finish the transmission. This is the condition to determine whether a flow is about to time out. Next, we need to calculate the residual amount of data to be transmitted. We assume variable r_k is the nontransmitted data, where $r_k = m_k - \sum_i^{X_k} B_k(t_i) \cdot t_i$. Then r_k/t_{i+1} is enough bandwidth that can satisfy the time requirement.

For the first factor and the second factor, we can combine them to deal with. By calculating the difference between the new and reduced flows, the total number of the current flows through the middlebox is updated. The system sets the total number as the count value of the flow information table.

When calculating the bandwidth for each flow, we first give the bandwidth to the flow that is about to time out. Then, the total number of flows is subtracted from the number of flows about to timeout, and the difference is obtained for the average allocation of bandwidth. The bandwidth allocation algorithm for the private network is shown in Algorithm 1.

4.2. Public Network. In the more widely used public network environment, it is not only blockchain communication, but also any other type of transmission. Network resources are no longer exclusive to blockchain services but compete with various data flows. In order to reflect the real environment of blockchain communication, we use the fat flow and slim flow competition model and define the state space and action space.

We choose the Deep Q Network (DQN) as DRL method to optimize the bandwidth allocation. DQN has some advantages: first, it only needs an incentive mechanism to evaluate the behavior so that it does not need data sets for training. Better results can be obtained by training according to the environment observation and action of the agent. Second, when the number of flows increases, the DQN can well deal with the large-dimensional state action space. Third, DQN has an experience replay mechanism, which reuses the samples to improve the learning efficiency. In this case, we use DQN algorithm to solve the problem.

4.2.1. State Space. For the public network, the model contains slim flows representing blockchain communications and fat flows representing other application flows. We define the network resource and flow status as the environment in reinforcement learning and the state of the environment is changing along with time. The flow information table contains all the flows state and updates in each time slot. Except for srcip, dstip, sport, and dport variables, other variables change with time. And the total number of data flows in the table also changes along with time.

```
Input: srcip, dstip, sport, dport
Output: bandwidth
  Caculate an available bandwidth for the flow identified by srcip, dstip, sport, and dport
  load data from flow information table into flowList
  create a timeoutList
  for each S_k in flowList do
    X_k^S←S_k.timer
    if d_k − X_k^S = 1 then
      timeoutList.add (S_k)
    end if
  end for
  use r_k to store the residual data for flow k
  use B_timeout to accumulate the bandwidth of the timeoutList
  B_timeout←0
  for each S_k in timeoutList do
    calculate the residual data for each flow
    r_k←m_k − ∑_i^{X_k} B_k(t_i) · t_i
    B_k^S←r_k/t_{i+1}
    B_timeout←B_timeout + B_k^S
  end for
  B_k^S←B − B_timeout/flowList.count − timeoutList.count
  update B_k^S for each flow to flowList
  bandwidth ←flowList.lookup(srcip, dstip,sport, dport)
  return bandwidth
```

Algorithm 1: Bandwidth allocation algorithm for private network.

According to the bandwidth allocation model and flow information table, the environment has three parts, the fat flow state, the slim flow state, and the whole number of flows. The fat flow state can be represented as $F_j(t) = [d_j(t), m_j(t)]$. And $d_j(t)$ is a function that changes with time t; it represents the remaining available time slots of the flow j. Function $m_j(t)$ denotes the remaining data to be transmitted. For fat flows j, $j \in \{1, 2, 3, \ldots N^F(t)\}$ and the number of fat flows is $N^F(t)$. Same to the fat flow, $S_k(t) = [d_k(t), m_k(t)]$ denotes the state of slim flow k. The number of slim flows is denotes as $N^S(t)$, and $k \in \{1, 2, 3, \ldots N^F(t)\}$. The total number of flows through the middlebox is the sum of fat and slim flows. It can be denoted as $N^F(t) + N^S(t)$, so the state space can be represented as

$$\text{State}(t) = \left[F_j(t), S_k(t), N^F(t) + N^S(t)\right]. \tag{10}$$

4.2.2. Action Space. In the public network, the decision made by the middlebox in each time slot is the calculation of the bandwidth value for each data flow. According to our bandwidth allocation model, the actions in each time slots can be denoted as

$$\text{Action}(t) = \left[B_j^F(t), B_k^S(t)\right], \tag{11}$$

where $B_j^F(t), j \in \{1, 2, 3, \ldots N^F(t)\}$ refers to the bandwidth of fat flow j, and $B_k^S(t), k \in \{1, 2, 3, \ldots N^S(t)\}$ refers to the bandwidth of slim flow k. The total bandwidth allocated in each action should meet the constraint C5 in equation (9).

The action includes all flows' bandwidth value in time slot t. The value computed by the middlebox is the available bandwidth for the sender. It will ensure that congestion is avoided and bandwidth utilization is improved. The action will affect the transition of state and the accumulation of reward.

4.3. Reward Function. The goal of our model is to maximize the total reward through bandwidth allocation in a certain period of time. The object is to maximize the rewards, which also means that our allocation strategy serves more data flows in the observation time.

We choose the RL method to optimize the bandwidth allocation, because it only needs an incentive mechanism to evaluate the behavior. We build such a mechanism to train the strategy of bandwidth allocation. If a flow is not completed within the required time due to improper bandwidth allocation, the reward is negative. If the bandwidth allocation is appropriate and the flow is completed within the specified time, a positive reward is obtained. We add a punish rule in the reward function, so it is a little different from the model definition in equations (7) and (8).

The details of the reward rules can be described as follows:

(i) In a time slot t, suppose the state space has a flow state $F_j(t) = [d_j(t) = 0, m_j(t) > 0]$, where $d_j(t) = 0$ indicates that it has timed out and $m_j(t) > 0$ indicates that there is still data that has not been served. Such a state is a bad situation and should be punished. We use $\theta_j(t) = -1$ to represent this state, and the reward of this flow $F_j(t)$ should be $r_j \cdot \theta_j(t)$.

(ii) In a certain time slot t, suppose the state space has a flow state $S_k(t) = [d_k(t) > = 0, m_k(t) = 0]$, where $m_k(t) = 0$ indicates that the data has been served, and $d_k(t) > 0$ indicates that the data has been completed before the timeout. Else $d_k(t) = 0$ indicates that the flow finished in time. This state is a good situation and it should be rewarded with the value of $r_k \cdot \theta_k(t)$ and $\theta_k(t) = 1$.

(iii) In other states, there is no change in the number of data flows, so no reward can be obtained in this situation. The corresponding parameter $\theta(t)$ equals zero in this state.

(iv) To optimize blockchain communication, the absolute value of the reward of slim flow should be greater than that of fat flow. It can be shown as $|r_k| > |r_j|, \forall k \in \{1,2,3,\ldots N^S(t)\}, j \in \{1,2,3,\ldots\ N^F(t)\}$.

The reward function $R(t)$ is the sum of rewards of all the flows at time slot t.

$$R(t) = R^F(t) + R^S(t) = \sum r_j \cdot \theta_j(t) + \sum r_k \cdot \theta_k(t),$$
$$\theta_{j,k}(t) = \begin{cases} -1 & , d_{j,k}(t) = 0, m_{j,k}(t) > 0, \\ 1 & , d_{j,k}(t) \geq 0, m_{j,k}(t) = 0, \\ 0 & , \text{others}. \end{cases} \quad (12)$$

After each action of bandwidth allocation, the system calculates the state of each flow. When the system has finished updating the state and the total number of data flows, it will enter the next state. According to the state of served data and the residue time requirement, the system decides whether to get a reward.

4.4. DQN. Our system needs to compute proper bandwidth for data flows in each time slot. The principle of proper bandwidth calculation is to maximize the rewards in the whole observation time period. When the number of flows increases or the network bandwidth to be allocated is large, the space of states and actions becomes difficult to calculate. Due to the large scale of state-changing of data flows in the middlebox, we employ the DQN method to deal with the high-dimensional state spaces and action spaces.

DQN is a reinforcement learning algorithm with the advantages of neural networks [33]. The agent of DQN can replace the role of the middlebox system and complete the bandwidth allocation according to learning from the environment. For our model, the long-term value function represents the accumulated reward from the data flow service. It can be denoted as

$$\begin{aligned} Q(\text{State}(t), \text{Action}(t)) \leftarrow & (1-\alpha) Q(\text{State}(t), \text{Action}(t)) + \\ & \alpha[R(t) + \gamma \max Q(\text{State}(t+1), \text{Action}(t+1))]. \end{aligned} \quad (13)$$

In this function, α is the learning rate that is used to balance the current and future reward value, and γ is the discount rate. The DQN method for bandwidth allocation is shown as Figure 5.

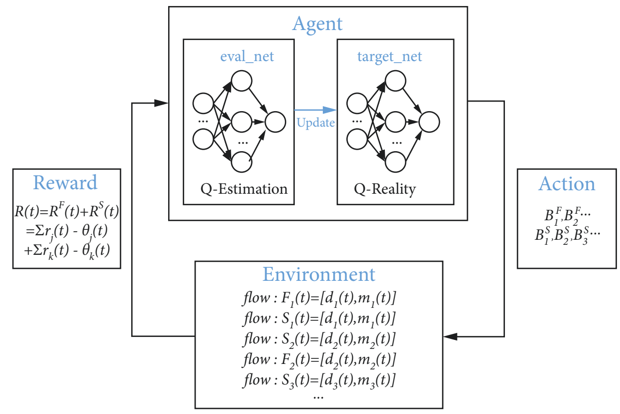


Figure 5: DQN method for bandwidth allocation.

In the DQN method, two neural networks are used to train the agent to do bandwidth allocation. One neural network is called eval_net that trains the parameter of agent to estimate Q value. The other neural network is called the target_net. It is used to save the target Q value. The eval_net and the target_net are with the same structure but different parameters. When training the agent to allocate bandwidth, the experience replay is well utilized to learn from the past operations. By random sampling of the experiences, the neural network will update more efficiently. The eval_net represents the Q-estimation which contains the new parameter. The target_net predicts Q-reality, but its parameter is not new. After some steps, the target_net will be updated according to the parameters of the eval_net. This mechanism, which is called fixed Q-targets, can overcome the data correlation and ensure the convergence of parameters. The algorithm of DQN for bandwidth allocation is present in Algorithm 2.

5. Experimental Environment and Results

In this section, we first introduce our experimental environment. Then we discuss performance evaluation.

5.1. Experimental Environment. In our test environment, we use VMware ESXi virtual machine system to build a local network. This network has 4 middleboxes and 4 endpoints to act as senders and receivers. These middleboxes can play the role of consensus nodes in blockchain communication. Each node is a virtual machine with 2 gigabit netcards and 4 GB memory. The operating system is Ubuntu 18.04 LTS with 4.15 version Linux kernel.

For throughput test, we use the network testing tool IPerf3. To simulate a complex network environment, TC (Traffic Control) tool is employed. Netem (Network emulation) is a kernel module controlled by TC. We use Netem to set packet loss and delay in the middlebox. In order to make a bottleneck in the network path, HTB (Hierarchy Token Bucket) is utilized to do shaping and policing for network traffic. It is also a type of queue managed by TC and can set a rated speed for a network interface.

In order to make the system support our available bandwidth notification method, we create a new kernel

Initialize memory M for experience replay
Initialize eval_net and $Q_{\text{estimation}}$ function with weight parameter ω
Initialize target_net and Q_{reality} function with weight parameter $\omega^{-} = \omega$
for each episode **do**
  Initialize State(t)
  for each time slot t **do**
    Generate a random number λ using ϵ – greedy policy for balancing exploration and exploitation
    if $\lambda < \epsilon$ **then**
      Randomly choose a series of bandwidth $\{B_1^F, B_2^F, \ldots, B_1^S, B_2^S, \ldots\}$ as the Action(t) for current flows.
    else
      Action(t) = avg max $Q(\text{State}(t), \text{Action}(t); \omega)$
    end if
    Execute Action(t)
    Obtain $R(t)$ and State$(t+1)$
    Save $\{\text{State}(t), \text{Action}(t), R(t), \text{State}(t+1)\}$ into memory D
    Randomly select batch of experience data $\{\text{State}_i, \text{Action}_i, R_i, \text{State}_{i+1}\}$ from D
    if $State_{i+1}$ is the terminal state **then**
      set $y_i = R_i$.
    else
      set $y_i = R_i + \gamma \max_a Q_{\text{reality}}(\text{State}_{i+1}, a)$
    end if
    Perform gradiant descent on $(y_i - Q_{\text{estimation}}(\text{State}_i, \text{Action}_i; \omega))^2$ with ω
    Update Q_{reality} parameter with $Q_{\text{estimation}}$
  end for
end for

Algorithm 2: DQN algorithm for bandwidth allocation in public network.

module called "tcp_abn" following the example of "tcp_bbr" that is developed by Google. Some kernel files of the TCP stack have been modified, while the compatibility has been kept. As our method needed, the modified files contain two types. One type is deployed on the middlebox. This type of file contains sch_htb.c. In sch_htb.c, the calculation of available bandwidth is added, and the bandwidth value is injected into the corresponding ACK packet. The other type is deployed on the sender. This type of file contains tcp.h, tcp_input.c, tcp_ipv4.c, tcp_output.c, and tcp_abn.c. The tcp_abn.c file is the major part of the "tcp_abn" module. Other files make the system suitable for the "tcp_abn" module and support applying the bandwidth values from the ACK packet.

5.2. Results. For the private network, we only initiate blockchain communication in our local network. In the middlebox, we set the upper limit of network bandwidth to 2 Gbit/s by HTB policing function. We use the sysctl command to change the kernel file "/proc/sys/net/ipv4/tcp_wmem" as "4096 16384 100000000" in each sender. This operation can make the system provide better performance which avoids limiting network speed due to insufficient send windows.

In this network configuration, we compare our ABN method with two recently popular algorithms, Cubic and BBR. Cubic is the default TCP congestion control algorithm in Linux system, and BBR is a Google provided algorithm that is also supported in Linux kernel.

In order to get a more significant effect, we set the delay changing in a certain range. It is hard for the RTT probing mechanism to measure an accurate value in variable delay. We use Netem tool to add latency in the network, and the delay range is 50 ms, 100 ms, and 200 ms, respectively. We use IPerf3 to transmit data in TCP protocol where the packets go through the middlebox. The middlebox will be deployed on Cubic, BBR, and our ABN. We use tcpdump to catch all the traffics generated by IPerf3 during 60 seconds and save then into a pcap file. Then, we convert these pcap files to csv format files and draw Figures 6–8.

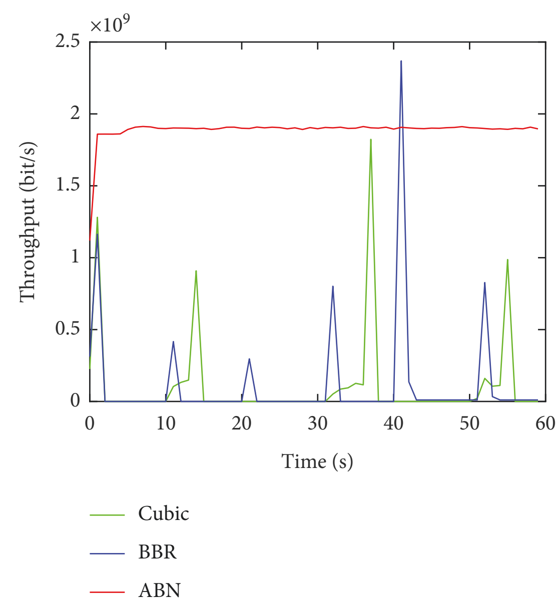


Figure 6: Throughput with 50 ms delay.

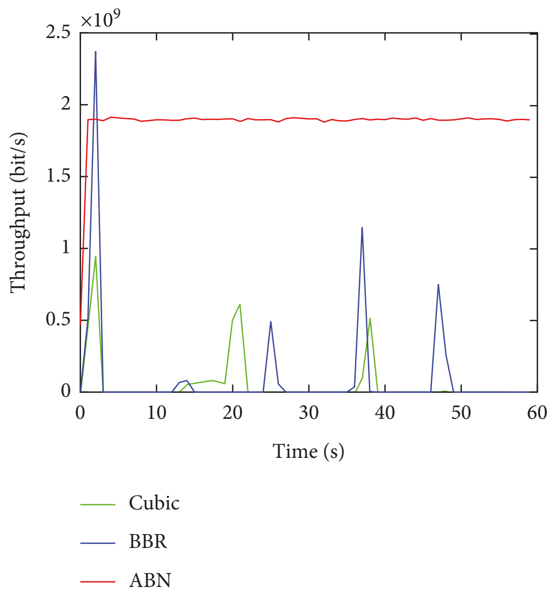


Figure 7: Throughput with 100 ms delay.

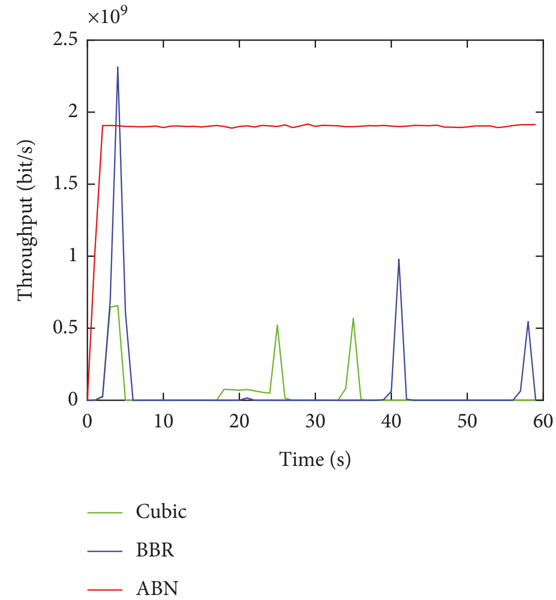


Figure 8: Throughput with 200 ms delay.

The throughput with 50 ms range delay is shown in Figure 6. We can find that our ABN method has a relatively stable speed close to the ceiling bandwidth. BBR has 6 spikes and Cubic has 4; in addition, the average height of spikes of BBR is higher than Cubic. It means the throughput of BBR is better than Cubic. The throughput results during 60 s with 50 ms range of delay are Cubic : 235 Mbit/s, BBR : 1370 Mbit/s, and ABN : 1990 Mbit/s. At the position of 40 s in Figure 6, the throughput is higher than the limit capacity. That is because BBR will probe the available bandwidth and sometimes overestimate it. Then it will occupy the network buffer of the middlebox.

The throughput with 100 ms and 200 ms range of delay is shown in Figures 7 and 8. With the delay increasing, we can see that the number of spikes of BBR and Cubic gets smaller. The throughput of our ABN decreases minorly and still stays at a stable speed. That is attributed to the bandwidth notification from the middlebox. The speed of ABN accords with the feedback of the network environment, not probing or guessing. BBR will drop the speed when it perceives the delay increasing in probeRTT phase. Cubic sends packets continually, so the spike of Cubic usually lasts longer than BBR. When delay increases and leads to packet drop occurrence, Cubic sharply reduces the transmission speed. If the packet loss state continues, Cubic will always maintain the low speed state. The throughput of Cubic with 100 ms range of delay is 205 Mbit/s. BBR is 1120 Mbit/s, and ABN is 1990 Mbit/s. With 200 ms range of latency, the throughput of Cubic drops to 93.2 Mbit/s. BBR's throughput shrinks to 465 Mbit/s. The average speed of ABN during 60 seconds is 1970 Mbit/s. The performance analyses demonstrate that the ABN method can stand delay influence and improve the network bandwidth utilization.

For the public network, we use a DQN model to do the performance simulation. There are two neural networks in DQN, eval_net and target_net, each network has two layers, and each layer has 20 neurons. The input of neural network is set according to the amount of flows and the size of bandwidth allocation action space. Tensorflow 1.12.0 with python 3.7 is employed to do the training on Windows 10 system. In the simulation, 4 blockchain nodes with PBFT consensus initiate slim flows through a middlebox. In order to highlight the support of the ABN method for blockchain communications, we add two fat flows with large bandwidth as the background traffic to compete with blockchain communications, and the agent tends to allocate bandwidth to blockchain flow for more rewards. The detail of simulation parameters can be shown in Table 2.

The loss of DQN model is as shown in Figure 9. We can find that the loss is getting lower with the training steps increasing. The loss value tends to be stable which means the model is available. In Figure 10, the reward variation with different learning rates is presented. The learning rate affects the rewards during the training. When the learning rate is 0.1 and 0.01, the reward curve oscillates violently and gets the optimal value occasionally. When the learning rate is 0.001, the reward curve is relatively convergent, but it does not converge to the optimal value. In our simulation, the learning rate of 0.0001 obtains better performance than others. As shown in the figure, the learning speed is acceptable, and the reward curve stably converges to the optimal value. The rule of reward function is to accumulate the reward of each served flow.

The comparison of reward between ABN, BBR, and Cubic is presented in Figure 11. In this training, we send one slim flow and fat flow at the beginning and add slim flows during 200 s. To simulate PBFT communications, we add slim flows as a multiple of 4 which is the number of consensus nodes. In the case of background traffic, ABN can obtain more rewards with the increase of flows. Although the reward of Cubic is the same as others at the beginning, it will

Table 2: The parameter settings in the simulation.

Parameter	Value	Description
Bandwidth	10 Mbit/s	The whole bandwidth in the network
Fat flow	1000 Mbits, 200 s, 80	A type of fat flow as background traffic
Slim flows	1 Mbits, 1 s, 10	Four slim flows in a group to represent blockchain communications
T	200 s	Total time of observation
Node	4	The number of blockchain nodes that initiate PBFT consensus communications
γ	0.9	The discount factor

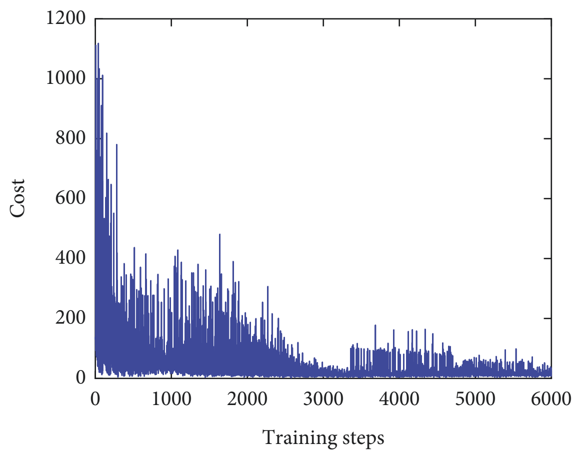


Figure 9: The learning loss of the DQN model for ABN.

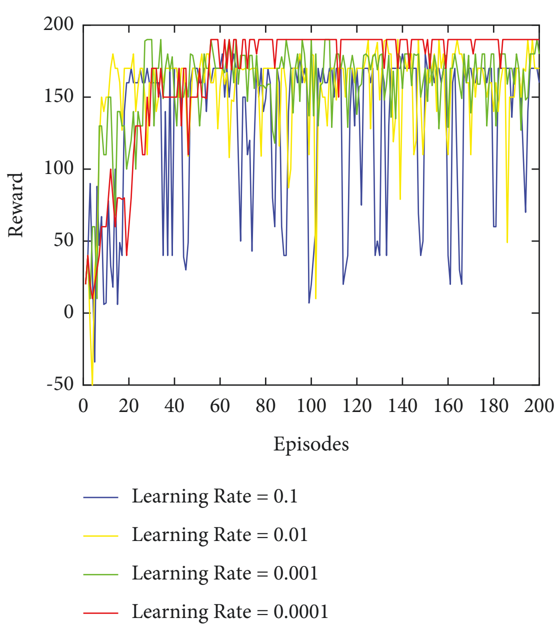


Figure 10: The rewards of DQN model for ABN with different learning rate.

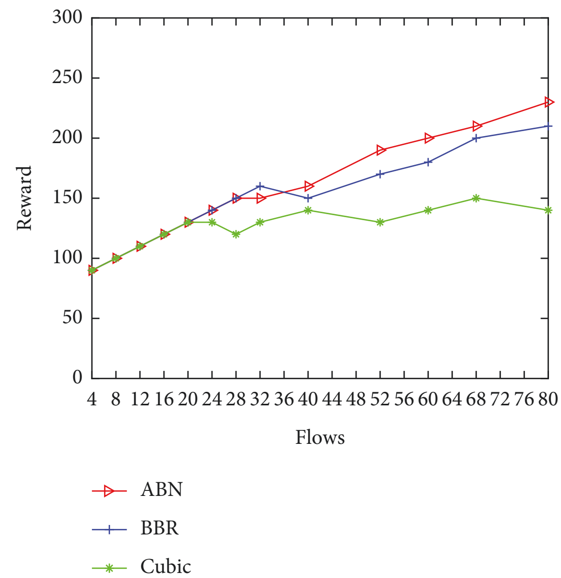


Figure 11: The reward comparison of ABN with BBR and Cubic.

be punished when congestion occurs with the increase of flows. The result is the slow growth of reward for Cubic. BBR sometimes can get a better reward, because it attempts to fill the pipe to probe bandwidth, and it possibly occupies the buffer. We can find that when the amount of flow is 32, the reward of BBR is a little higher than ABN. But this situation cannot be persistent; it will drop the speed when BBR is in the drain phase. As we can see, ABN can get a more stable reward augmentation than BBR in a long-term running.

6. Conclusions

In this paper, we investigate a blockchain-enabled IoT network deployed on MEC servers and analyze the characteristics of the network flow of blockchain in consensus communication. After researching the relationship between consensus communication and performance in the permissioned blockchain, we determine to optimize the performance by focusing on network bandwidth utilization. We proposed an optimal allocation algorithm called available bandwidth notification. The ABN method can provide efficient bandwidth utilization and avoid congestion; it will improve the speed of blockchain consensus, so as to ensure IoT data is safe and fast. The results derived from the evaluation demonstrate the high performance of our method. Compared with the current TCP network, our method is more friendly to the communication of blockchain. With the further development of the research, we would consider the data privacy protection function of blockchain-enabled IoT network.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] X. Xu, Z. Fang, J. Zhang et al., "Edge content caching with deep spatiotemporal residual network for iov in smart city," *ACM Transactions on Sensor Networks*, vol. 17, no. 3, pp. 1–33, 2021.

[2] J. Huang, C. Zhang, and J. Zhang, "A mqueue approach of energy efficient task scheduling for sensor hubs," *Chinese Journal of Electronics*, vol. 29, no. 2, pp. 242–247, 2020.

[3] X. Xu, H. Tian, X. Zhang, L. Qi, and Q. He, "Discov: distributed covid-19 detection on x-ray images with edge-cloud collaboration," *IEEE Transactions on Services Computing*, 2022.

[4] T. M. Fernandez-Carames and P. Fraga-Lamas, "A review on the use of blockchain for the internet of things," *IEEE Access*, vol. 6, pp. 32979–33001, 2018.

[5] Y. Chen, W. Gu, and K. Li, "Dynamic task offloading for internet of things in mobile edge computing via deep reinforcement learning," *International Journal of Communication Systems*, 2022.

[6] J. Feng, F. R. Yu, Q. Pei, J. Du, and L. Zhu, "Joint optimization of radio and computational resources allocation in blockchain-enabled mobile edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 4321–4334, 2020.

[7] Y. Chen, F. Zhao, Y. Lu, and X. Chen, "Dynamic task offloading for mobile edge computing with hybrid energy supply," *Tsinghua Science and Technology*, 2021.

[8] Y. Chen, Y. Zhang, Y. Wu, L. Qi, and X. X. Chen, "Joint task scheduling and energy management for heterogeneous mobile edge computing with hybrid energy supply," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8419–8429, 2020.

[9] Y. Chen, H. Xing, and Z. Ma, *Cost-Efficient Edge Caching for NOMA-Enabled IoT Services"*, China Communications, 2022.

[10] Q. F. Shao, C. Q. Jin, Z. Zhang, W. N. Qian, and A. Y. Zhou, "Blockchain: architecture and research progress," *Chinese Journal of Computers*, vol. 41, no. 8, pp. 969–988, 2018.

[11] T. Mitani and A. Otsuka, "Traceability in permissioned blockchain," in *Proceedings of the IEEE International Conference on Blockchain*, Atlanta, GA, USA, July 2019.

[12] M. Castro, "Barbara and Liskov, "Practical byzantine fault tolerance," in *Proceedings of the third symposium on Operating systems design and implementation*, pp. 173–186l, New Orleans Louisiana, USA, 1999.

[13] T. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K. L. Tan, "Blockbench: a framework for analyzing private blockchains," in *Proceedings of the 2017 ACM International Conference on Management of Data*, pp. 1085–1100, Chicago Illinois, USA, May 2017.

[14] M. El-Hindi, C. Binnig, A. Arasu, D. Kossmann, and R. Ramamurthy, "BlockchainDB," *Proceedings of the VLDB Endowment*, vol. 12, no. 11, pp. 1597–1609, 2019.

[15] J. Zou, B. Ye, L. Qu, Y. Wang, M. A. Orgun, and L. Li, "A proof-of-trust consensus protocol for enhancing accountability in crowdsourcing services," *IEEE Transactions on Services Computing*, vol. 12, no. 3, pp. 429–445, 2019.

[16] S. Nakamoto, *Bitcoin: A Peer-To-Peer Electronic Cash System"*, 2008.

[17] V. Buterin, *A Next-Generation Smart Contract and Decentralized Application Platform"*, 2014.

[18] C. Catalini and J. S. Gans, "Some simple economics of the blockchain," *Communications of the ACM*, vol. 63, no. 7, pp. 80–90, 2020.

[19] H. Sukhwani, JM. Martinez, X. Chang, KS. Trivedi, and A. Rindos, "Performance modeling of pbft consensus process for permissioned blockchain network (hyperledger fabric)," in *Proceedings of the IEEE 36th Symposium on Reliable Distributed Systems*, Hong kong, China, September 2017.

[20] M. Vukoli, "The quest for scalable blockchain fabric: proof-of-work vs. bft replication," in *Proceeding of the Open Problems in Network Security*, Zurich, Switzerland, October 2015.

[21] R. Sekaran, R. Patan, A. Raveendran, F. Al-Turjman, M. Ramachandran, and L. Mostarda, "Survival study on blockchain based 6G-enabled mobile edge computation for IoT automation," *IEEE Access*, vol. 8, pp. 143453–143463, 2020.

[22] W. Li, Z. Su, R. Li, K. Zhang, and Y. Wang, "Blockchain-based data security for artificial intelligence applications in 6G networks," *IEEE Network*, vol. 34, no. 6, pp. 31–37, 2020.

[23] I. G. AK. Gemeliarana and RF. Sari, "Evaluation of proof of work (pow) blockchains security network on selfish mining," in *Proceedings of the IEEE Internasional Seminar on Research of Information Technology & Intelligent Systems*, Yogyakarta, Indonesia, November 2018.

[24] T. Nakaike, Q. Zhang, Y. Ueda, T. Inagaki, and M. Ohara, "Hyperledger fabric performance characterization and optimization using goleveldb benchmark," in *Proceedings of the IEEE International Conference on Blockchain and Cryptocurrency*, Toronto, Canada, May 2020.

[25] NN. Lan, T. DT. Nguyen, TN. Dinh, and MT. Thai, "Optchain: optimal transactions placement for scalable blockchain sharding," in *Proceedings of the IEEE 39th International Conference on Distributed Computing Systems*, Dallas, TX, USA, July 2019.

[26] S. Kantesariya and D. Goswami, "Determining optimal shard size in a hierarchical blockchain architecture," ", in *Proceedings of the IEEE International Conference on Blockchain and Cryptocurrency*, Toronto, Canada, May 2020.

[27] J. Huang, Z. Tong, and Z. Feng, "Geographical POI recommendation for internet of things: a federated learning approach using Matrix factorization," *International Journal of Communication Systems*, 2022.

[28] X. Fu, F. R. Yu, J. Wang, Q. Qi, and J. Liao, "Performance optimization for blockchain-enabled distributed network function virtualization management and orchestration," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6670–6679, 2020.

[29] J. Huang, B. Lv, Y. Wu, and Y. X. Chen, "Dynamic admission control and resource allocation for mobile edge computing

enabled small cell network," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 2, pp. 1964–1973, 2022.
[30] B. Huang, L. Jin, Z. Lu, X. Zhou, P. CK. Hung, and Bor, "Toward high-performance permissioned blockchain in rdma-enabled network," *IEEE Transactions on Services Computing*, vol. 13, no. 2, 2020.
[31] N. Cardwell, "Bbr: congestion-based congestion control," *ACM Queue*, vol. 60, no. 2, 2017.
[32] S. Floyd, "Tcp and explicit congestion notification," *ACM Computer Communications Review*, vol. 24, no. 5, 1994.
[33] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

Hindawi
Scientific Programming
Volume 2022, Article ID 3943419, 17 pages
https://doi.org/10.1155/2022/3943419

Research Article

Keywords-Driven Paper Recommendation Based on Mobile Edge Computing Environment Framework

Hanwen Liu [1], Shuo Wang [1], Huali Ren [2], Shunmei Meng [1], Jun Hou [3,4] and Qianmu Li [1,5]

[1] *School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210096, China*
[2] *Institution of Artificial Intelligence and Blockchain, Guangzhou University, Guangzhou 510006, China*
[3] *School of Social Science, Nanjing Vocational University of Industry Technology, Nanjing 210046, China*
[4] *Intelligent Manufacturing Department, Wuyi University, Jiangmen 529020, China*
[5] *School of Cyber Science and Engineering, Nanjing University of Science and Technology, Nanjing 210096, China*

Correspondence should be addressed to Qianmu Li; qianmu@njust.edu.cn

Received 23 January 2022; Revised 1 March 2022; Accepted 24 March 2022; Published 8 April 2022

Academic Editor: Jiwei Huang

In the cloud computing era, a paper recommender system is usually deployed on the cloud server and return recommendation results to readers directly. However, considering the paper recommender system, processing tremendous paper citation data on the cloud cannot provide fine-grained personalized and real-time recommendations for each reader because these recommended papers from the cloud are far from readers and probably not correlated strongly with each other for helping each reader research further and deeper in the interested field. Recently, the edge-cloud collaboration-based recommender system has been used for releasing parts of the cloud computing task to the edge and provides the recommendation near the client. Based on the edge computing recommender system, a keywords-driven and weight-aware paper recommendation approach is presented, namely, $LP\text{-}PR_{k+}\ w$ (link prediction-paper recommendation), to enable intelligent, personalized, and efficient paper recommendation services in the mobile edge computing environment. Specifically, the whole paper recommendation process mainly covers two parts: optimizing the existing paper citation graph via introducing a weighted similarity (i.e., building a weighted paper correlation graph) and then recommending a set of correlated papers according to the weighted paper correlation graph and the users' query keywords. Experiments on a real-world paper correlation dataset, Hep-Th, show the capability of our proposal for improving the paper recommendation performance and its superiority against other related solutions.

1. Introduction

Generally, recommender system is based on the cloud-client structure that raises the risk of delays due to network bandwidth and latency, data leakage during remote transmission, and recommending uncorrelated results for users by processing tremendous multivariate heterogeneous data. In recent years, edge-cloud collaboration-based recommender systems are proposed to solve these problems [1, 2]. In the mobile edge computing environment, the recommender system can utilize the real-time information of users on the edge end to provide better recommendations.

Currently, the paper recommender system is one of the major tools and ways for readers to find their required papers. For example, popular academic paper search tools on the mobile edge end (e.g., Baidu Academic and Google Scholar) allow readers to look for their interesting papers from massive papers registered on the web, based on a set of desired keywords. In practice, one paper only covers readers' partial query keywords. Therefore, the paper search tools probably recommended a set of papers that contain all the query keywords to meet the readers' tastes for papers.

As shown in the right part in Figure 1, the fundamental paper recommendation process mainly consists of three phases in the cloud computing environment. First, a reader types a set of query keywords into the paper recommender system (e.g., $\{k_1, k_2, k_3, k_4, k_5\}$). The second phase is papers' discovery, in which vast candidate papers can be identified

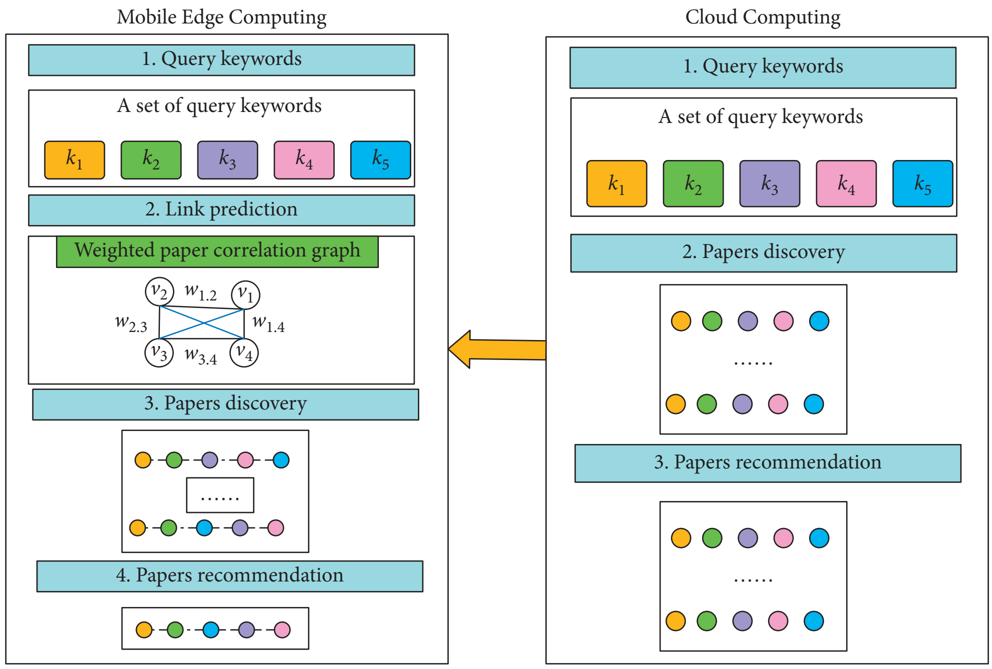


FIGURE 1: The recommendation process on the cloud computing and the mobile edge computing.

by using traditional keyword search methods [3]. The third phase is papers' recommendation, in which the recommender system will output all candidate papers to the reader.

However, finding a set of desirable papers from massive candidates on the cloud server for the reader is often a sophisticated job even for an experienced reader. The reasons are mainly threefold, which are clarified as follows:

(1) As the paper recommender system usually returns recommendation results to readers directly in the cloud computing, the paper recommender system cannot provide fine-grained personalized and real-time recommendation for readers.

(2) The returned paper list should meet the following conditions. First, these papers collectively cover all the query keywords from the reader. Second, the recommended papers should own certain correlations with each other so as to aid the reader in launching in-depth and continuous research on an identical idea.

(3) Traditional keywords-based paper recommendation approaches hardly analyze the possible correlations among different papers. Although paper citation graphs provide a good indicator of paper correlation relationships, they still face the serious problem of sparse data.

Currently, link prediction approaches are the best alternatives to address the sparse data problem of graphs (networks) [4], which aims to find the missing links and forecast future links based on known graphs (networks) information [5]. Specifically, link prediction approaches estimate the likelihood of building a new link among two unconnected nodes according to their graph structure and attribute information. Hence, applying link prediction to paper recommendations is of theoretical and practical significance.

To satisfy readers' requirements on paper search, we propose a novel keywords-driven and weight-aware paper recommendation approach deployed on the mobile edge end, that is, $LP-PR_{k+w}$ (link prediction-paper recommendation). As shown in the left part in Figure 1, $LP-PR_{k+w}$ integrates multiple operations: query keywords, link prediction, papers' discovery, and papers' recommendation. On the mobile edge end [6], through analyzing a reader's query keywords, $LP-PR_{k+w}$ can return a set of correlated papers (depicted by a connected subgraph) by mining the potential correlation patterns hidden in the weighted paper correlation graph. As output, the returned subgraph is interconnected by the target papers (containing query keywords) and bridging papers (containing no query keywords). Particularly, we consider interchangeability of one paper and its corresponding node in this paper, denoted as p or v.

Generally, we achieve the following main contributions:

(1) We propose a novel keywords-driven and weight-aware paper recommendation approach based on the mobile edge computing framework, i.e., $LP-PR_{k+w}$, which enables intelligent, personalized, and efficient paper recommendation services in the mobile edge computing environment.

(2) We optimize the existing paper citation graph model by introducing weighted similarity-based link prediction. Thus, we construct a weighted paper correlation graph.

(3) To evaluate the usefulness and feasibility of $LP - PR_{k+w}$, we conduct a group of experiments on a real-world paper citation dataset, Hep-Th.

The rest of the paper is organized as follows. Related researches are summarized in Section 2. Paper motivation is demonstrated in Section 3. In Section 4, we introduce the weighted similarity-based link prediction approach. Section 5 answers readers' keywords query via the proposed $LP - PR_{k+w}$ solution. Section 6 analyzes and evaluates $LP - PR_{k+w}$ through experimental comparisons. Finally, we summarize the paper and point out the future research directions in Section 7.

2. Related Work

2.1. Paper Recommendation. In general, Collaborative Filtering (CF) approach can calculate the similarity scores among different items; thus, paper recommender systems apply the CF approach and focus on ratings matrices. Furthermore, the CF approach can also calculate the ratings' matrixes that are created on a paper citation graph [7]. Therefore, the early works of paper recommendation mainly used the CF approach to recommend papers. With the application of the CF approach in the paper recommendation, researchers found that the CF approach is generally restricted to cold-start and data sparsity problems [8]. In addition, Content-Based Filtering (CBF) approach is similar to the CF approach. The CBF [9] approach mainly focuses on the content relevance [10, 11] among papers to recommend papers. However, these recommended papers hardly match readers' deep and continuous research around an identical focus. Furthermore, the CBF encounters the semantic ambiguity problem.

As the citation relationships between papers can reflect the correlations among papers' research content, a graph-based approach becomes research focus. For example, Meng et al. [12] regarded authors, papers, topics, and keywords as nodes and multiple relationships as edges; the approach executed a random walk on the constructed four-layer heterogeneous graph to recommend papers. Furthermore, [13] proposed a graph-based PageRank-like paper recommendation approach, which mainly executed a biased random walk on paper citation graphs to recommend papers. Although [12] and [13] take papers' citation relationships into account, these approaches do not tackle the data sparsity in the existing paper citation graphs.

2.2. Link Prediction. Link prediction [14] is an important approach of resolving links' sparsity problem as it can compute the likelihood of building new links between two unconnected nodes. At present, three types of link prediction have been identified, that is, similarity-based method, maximum likelihood approach, and probabilistic method. The similarity-based method was used in large-scale networks as it could compute the similarity score among any two nodes. The maximum likelihood approach predicted links by utilizing specific parameters, and the probabilistic method could employ the trained model to forecast links [15]. However, these two approaches did not apply to the large-scale networks. In the paper, our proposal mainly employs the similarity-based method considering that an existing paper citation graph is a large-scale paper relationships network [16].

Currently, more researchers considered utilizing the weight of links in link prediction approaches. For example, the work of [17] tried to identify strong ties (e.g., spouses or romantic partners) in social network, and these strong ties were considered as the different link weight among users. Furthermore, [18] investigated the use of link strength on link prediction approaches; their proposed weighting criterion was based absolutely on topological data (i.e., the frequency of interactions among nodes). In addition, link prediction model could be categorized into two types: an unsupervised link prediction model and a supervised link prediction model. For example, [18] and [19] made use of user's attributes information (i.e., gender, age, and sex) and interactive activities to estimate the social relationships strength on unsupervised link prediction model. Furthermore, [20] proposed a supervised link prediction model that used similarity metrics to calculate the similarity eigenvectors between a pair of nodes, and these eigenvectors were regarded as new databases for constructing the supervised model in predictive tasks [21].

In view of the above research content, a novel keywords-driven and weight-aware paper recommendation approach, that is, $LP - PR_{k+w}$, is proposed in the paper, to cope with the sparse data problem and recommend a set of correlated papers. Next, actual examples are presented in Section 3 to further demonstrate the research motivation of the paper.

3. Research Motivation

In this section, the examples of Figures 2 and 3 are recruited to directly demonstrate our research motivation. Figure 2 describes a scenario that a reader needs to perform five keywords research before he creates a new paper: (1) *keyword search* (i.e., k_1) is used by readers to search for interesting papers by typing query keywords; (2) *paper citation graph* (i.e., k_4) is for finding the correlation relationships among different papers (i.e., mining the potential correlation patterns); (3) *link prediction* (i.e., k_2) *is* to solve the data sparsity of paper citation graph; (4) *Steiner tree* (i.e., k_3) [22] is utilized to find a set of correlated papers; and (5) *dynamic programming* (i.e., k_5) [23] is applied to solve the Steiner tree problem. Thus, the reader obtains five corresponding query keywords, $Q = \{k_1, k_2, k_3, k_4, k_5\}$.

Figure 3 is a part of an undirected paper citation graph. It contains 14 nodes (papers), that is, $v_1, \ldots, v_{14}$, covering diverse keywords. The notation v_{13} $\{k_{11}, k_{13}\}$ $\{2021\}$ $\{a_1\}$ indicates that node v_{13} offers keywords k_{11} and k_{13}, an author a_1, and the published time 2021. $e(v_1, v_{10})$ indicates that nodes v_1 and v_{10} generate an undirected citation relationship.

According to the query keywords, the reader easily obtains a set of papers in Figure 3 (i.e., $R_p = \{v_1, v_2, v_3, v_4, v_5, v_6, v_{11}, v_{12}\}$). However, these returned papers fail to satisfy the reader deep and continuous

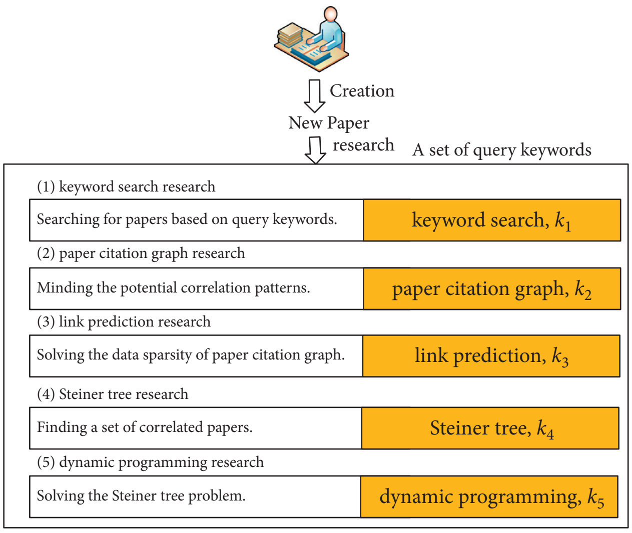


Figure 2: The paper research and creation tasks: an example.

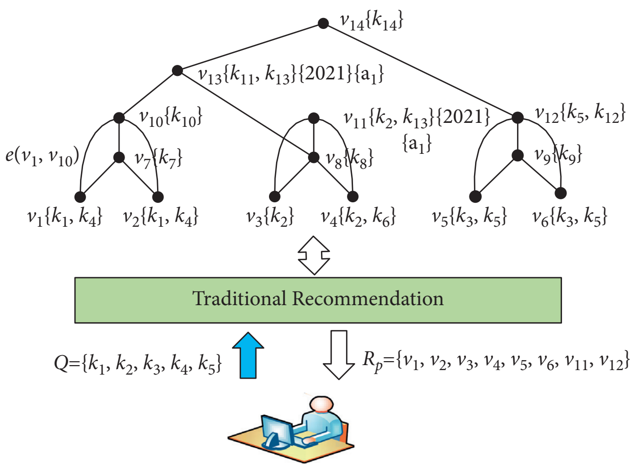


Figure 3: Tradition recommendation: an example.

academic research around an identical focus. In fact, a paper recommender system generally recommends a larger number of candidate papers to readers; furthermore, the correlation relationships among candidate papers are transparent to readers. Therefore, readers hardly select a set of correlated papers from these candidate papers. Luckily, a paper citation graph depicts citation relationships among diverse papers, so it provides a promising way to mine the potential correlation patterns. However, the paper citation graph faces to the sparse data problem; that is, it does not consider the potential correlations among different papers not connected in the graph. For example, nodes v_{11} and v_{13} have common research content, while they fail to build a correlation relationship in Figure 3.

According to the above examples, in the mobile edge computing environment, we firstly address the data sparsity of paper citation graph and then recommend a set of correlated papers, these contents are presented in detail in Section 4 and Section 5, respectively.

4. Link Prediction of $LP\text{-}PR_{k+w}$

4.1. Weighted Similarity-Based Link Prediction Approach. According to the analysis of the research motivation, we propose a weighted similarity-based link prediction schema that follows tasks' sequence [24]. In Figure 4, the link prediction process mainly comprises the following activities:

Activity 1. Preprocessing of Graph. For simplicity, a directed paper citation graph is treated as an undirected graph.

Activity 2. Nodes' Weighting. This activity mainly calculates the actual weight (i.e., w_{ij}) between all nodes of the undirected paper citation graph. Here, the weights of two connected nodes and two unconnected nodes are both computed by employing the following KTA weighting criterion.

Criterion. Keywords, Time, and Authors (KTA). Here, the number of common keywords of two papers increases and their published time are relatively close; thus, their weight (i.e., w_{ij}) will be larger. Furthermore, the theory of [25] states that two different papers containing common authors tend to refer to each other. Thus, the KTA weighting criterion is as follows:

$$\text{cosine}\left(K_{v_i}, K_{v_j}\right) = \frac{\left|K_{v_i} \cap K_{v_j}\right|}{\sqrt{\left|K_{v_i}\right|} * \sqrt{\left|K_{v_j}\right|}},$$

$$k(t) = \frac{1}{1 + e^{-\left|t_{v_i} - t_{v_j}\right|}},$$

$$\text{cosine}\left(A_{v_i}, A_{v_j}\right) = \frac{\left|A_{v_i} \cap A_{v_j}\right|}{\sqrt{\left|A_{v_i}\right|} * \sqrt{\left|A_{v_j}\right|}},$$

$$w_{ij} = w^{\text{KTA}}\left(v_i, v_j\right) = \beta^{\left(1-cosine\left(K_{v_i},K_{vj}\right)\right)} \times \lambda^{k(t)} \times \alpha^{\left(1-cosine\left(A_{v_i},A_{vj}\right)\right)}, \tag{1}$$

where β, λ, and α are arbitrary parameter values, i.e., β, λ, and $\alpha \in (0, 1)$. Higher (or lower) values of β, λ, and α intensify (or attenuate) the influence of keywords, published time, and authors in the weighting criterion, respectively. K_{v_i} (K_{v_j}) and A_{v_i} (A_{v_j}) denote a set of keywords and authors, respectively. $A_{v_i} \cap A_{v_j}$ and $K_{v_i} \cap K_{v_j}$ represent co-authors and common keywords, respectively. $\text{cosine}(A_{v_i}, A_{v_j})$ and $\text{cosine}(K_{v_i}, K_{v_j})$ denote that the method computes the similarity authors of and keywords between two nodes, respectively. t_{v_i} and t_{v_j} indicate the published time of two nodes, respectively.

Activity 3. Score Calculation and Ranking.

(1) Firstly, we get the actual weight of two connected nodes (e.g., v_i and v_z, v_j and v_z). Then, we calculate

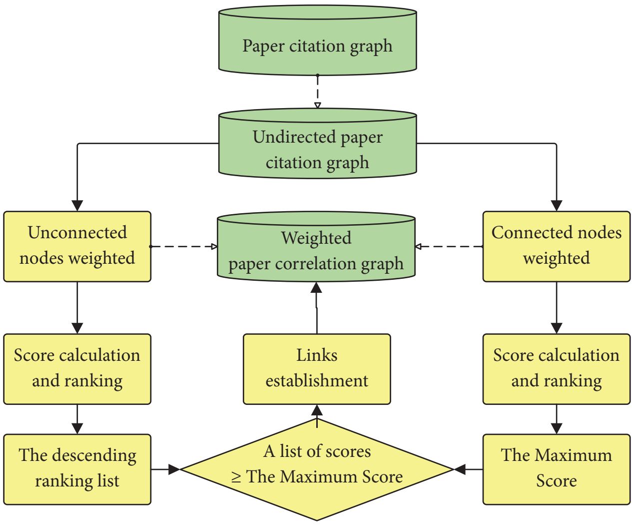


FIGURE 4: Process of the weighted similarity-based link prediction: overview.

the weight of two nonconnected nodes (e.g., v_i and v_j) by using the weighted similarity function (i.e., weighted common neighbor). Finally, we sort a descending list in order, and the maximum score [26] is saved in $w_{\max}$ (v_i, v_j).

Weighted Common Neighbor—WCN (v_i, v_j) [20]. It computes the average weight between two nonconnected nodes v_i and v_j; that is,

$$WCN(v_i, v_j) = \sum_{v_z \in \Gamma(v_i) \cap \Gamma(v_j)} \frac{w^{\text{KTA}}(v_i, v_z) + w^{\text{KTA}}(v_z, v_j)}{2},$$

$$WCN(v_i, v_j) = \frac{WCN(v_i, v_j)}{|\Gamma(v_i) \cap \Gamma(v_j)|},$$

$$w_{\max}(v_i, v_j) = \arg \max_{i,j=1,\ldots,N} w^{\text{WCN}}(v_i, v_j), \tag{2}$$

where (6) calculates the weight of two nonconnected nodes v_i and v_j by using the WCN. $|\Gamma(v_i) \cap \Gamma(v_j)|$ represents the number of common neighbor nodes. Equation (2) obtains a maximum weight (i.e., the maximum score) over the undirected paper citation graph.

(2) Here, we directly use the actual weight of all unconnected nodes to produce a descending-ranking list.

Activity 4. Links Establishment. This activity connects a pair of unconnected nodes. *LP* (link prediction) is defined as

$$LP = \{w_{ij} \geq w_{\max}(v_i, v_i)\}. \tag{3}$$

4.2. Weighted Paper Correlation Graph. In the mobile edge computing environment, $LP - PR_{k+w}$ builds a weighted paper correlation graph by using the weighted similarity-based link prediction. The weighted paper correlation graph is defined as follows.

Definition 1 (Nodes). For one paper p, the weighted paper correlation graph has a mapped node v. And the node covers diverse keywords (i.e., $k_1 \ldots k_z$), where these keywords denote its main research content.

Definition 2 (Edges). For any pair of nodes (v_i, v_j), the weighted paper correlation graph has a corresponding edge $e(v_i, v_j)$. And $e(v_i, v_j)$ denotes the fact that two papers have a correlation relationship.

Definition 3 (Weights). For any two connected nodes v_i and v_j, the weight of w_{ij}/w_{ij} (larger than 0) directly indicates the strength of correlation between nodes v_i and v_j. For example, considering two edges $e(v_1, v_2)$ and $e(v_2, v_3)$, if $w_{1,2} > w_{2,3}$, then node v_2 has better "similarity" (correlation) with node v_1 instead of node v_3.

Definition 4 (Weighted Paper Correlation Graph (W-PCG)). The W-PCG is expressed in $G_w(V_p, E_p, W_p)$, where V_p, E_p, and W_p represent a set of nodes, edges, and weights.

According to Definition 1, each node of G_w contains diverse keywords. To expediently answer the keywords query of Section 5, $LP - PR_{k+w}$ needs to preestablish an inverted index S_k. Concretely, given a query keyword k, we can faster search for all papers containing keyword k. For example, nodes v_1 and v_2 cover keywords k_1 (i.e., $S_k\{k_1\} = \{v_1, v_2\}$).

5. Paper Recommendation of $LP\text{-}PR_{k+}w$

5.1. Problem Formalization of Keyword-Driven Pattern Mining. On the mobile edge end [27], according to reader's query keywords, we will introduce our paper recommendation approach based on a W-PCG (i.e., G_w). Specifically, given a query Q containing l $(l \geq 2)$ query keywords (i.e., $Q = \{k_1, \cdots, k_l\}$), our proposal can find optimal answer trees on G_w, denoted as $T_w(Q)$, where $T_w(Q)$ is not only a connected tree containing all query keywords (i.e., Q) but also having highest correlation. To better clarify the paper, we summarize the symbols in Table 1.

Figure 5 is a part of G_w and contains the same nodes of Figure 3 (i.e., $v_1, \cdots, v_{14}$). And a new link $e(v_{11}, v_{13})$ (i.e., blue line) is added in Figure 5. According to the query keywords of Figure 2, i.e., $Q = \{k_1, k_2, k_3, k_4, k_5\}$, nodes v_1 and v_2 contain keywords k_1 and k_1, nodes v_3, v_4, and v_{11} contain keyword k_2, nodes v_5 and v_6 contain keywords k_3 and k_5, and node v_{12} contains keyword k_5. Thus, an answer tree $T_w(Q)$ connects one node from $\{v_1, v_2\}$, one node from $\{v_3, v_4, v_{11}\}$, one node from $\{v_5, v_6\}$, and one node from $\{v_5, v_6, v_{12}\}$; furthermore, $T_w(Q)$ may connect nodes that do not contain any query keywords, i.e., v_{10}, v_{13}, and v_{14}.

Table 1: Symbol definition.

Symbol	Definition
p/v	A paper
$k_1, \ldots, k_z$	The paper contains keywords
$e(v_i, v_j)$	A correlation relationship
Q/K	A set of query keywords
A_{v_i}/K_{v_i}	A set of authors/keywords
V_p	A set of nodes
E_p	A set of edges
W_p	A set of weights
$G_w(V_p, E_p, W_p)$	W-PCG
S_k	An inverted index
$T_w(Q)$	An optimal answer tree
Q^1/Q^2	Two queues
$T_{w\min}(v, K')$	A minimum group weighted Steiner tree rooted at v
R_P	Recommendation result

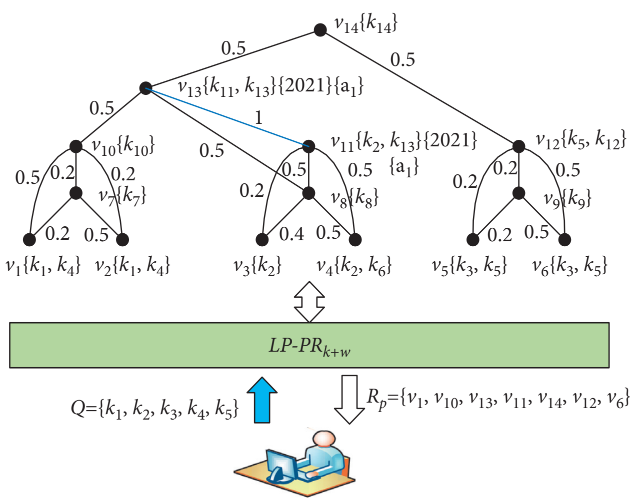


Figure 5: $LP-PR_{k+w}$: the same paper recommendation example.

Therefore, the reader will obtain a set of papers that is different from Figure 3 (i.e., $R_p = \{v_1, v_{10}, v_{13}, v_{11}, v_{14}, v_{12}, v_{16}\}$).

According to the above example, in the mobile edge computing environment, the reader initially pursues a weighted Steiner tree $T_w(Q)$ [28], and the definition is as follows.

Definition 5 (Weighted Steiner Tree). Given a W-PCG (i.e., G_w) and a set of nodes $V_p' \subseteq V_p$, when $T_w(Q)$ is a connected subgraph covering all nodes of V_p', $T_w(Q)$ is a weighted Steiner tree.

Considering the inverted index of Section 4, we recognize multiple groups of nodes according to diverse query keywords from $Q = \{k_1, \ldots, k_l\}$, denoted as $V_{p1}, \ldots, V_{pl}$, where V_{pn} $(1 \le n \le l)$ is a set of nodes covering a query keyword k_n $(1 \le n \le l)$. Next, we need to find a group weighted Steiner tree that covers all query keywords. The definition is as follows.

Definition 6 (Group Weighted Steiner Tree). Given G_w and multiple groups of nodes $V_{p1}, \ldots, V_{pl} \subseteq V_p$, when a weighted Steiner tree $w_{ij}(Q)$ selects nicely one node from each group V_{pn} $(1 \le n \le l)$, $T_w(Q)$ is a group weighted Steiner tree.

There may be obtained multiple diverse group weighted Steiner trees in answer to reader's keywords query. In fact, our recommendation goal, in the mobile edge computing environment, is simply to recommend a set of "most correlative" papers covering all query keywords. Here, the weight T_w of G_w denotes the "similarity" (correlations) of nodes v_i and v_i. Hence, the recommendation goal is finding a group weighted Steiner tree with maximal weight. In practice, $LP-PR_{k+w}$ considers the weight-aware (i.e., correlation-aware) paper recommendation problem as an optimization problem, and the object function is of "the smaller weight the better" case. Here, we transform Figure 5 into Figure 6. In Figure 6, the weight w_{ij} of edge $e(v_i, v_j)$ is represented by $1/w_{ij}$. For example, when $w_{1,10} = 0.5$ holds in Figure 5, $w_{1,10} = 2$ holds in Figure 6. In the remainder of this paper, we only use a converted W-PCG (i.e., G_w) as illustration. Next, we must search for a group weighted Steiner tree with minimum weight. The definition is as follows.

Definition 7 (Minimum Group Weighted Steiner Tree). Given a set of alternative group weighted Steiner trees, that is, $T_{w1}(Q), \ldots, T_{wm}(Q)$, when $w(T_{wi}(Q)) = min(w(T_{w1}(Q)), \ldots, w(T_{wm}(Q)))$ $(1 \le i \le m)$, $T_{wi}(Q)$ is a minimum group weighted Steiner tree.

5.2. Searching for Optimal Answer via Pattern Mining. According to readers' keywords query Q, $LP-PR_{k+w}$ must search a minimum group weighted Steiner tree $T_w(Q)$ representing final solution for readers via mining on a W-PCG (i.e., G_w). However, the computation of the $T_w(Q)$ is the NP-complete problem. Therefore, we will utilize a dynamic programming (DP) technique [28] to solve this problem.

Generally, the DP technique firstly divides the MGWST (minimum group weighted Steiner tree) problem into a set of easier subproblems. Next, each same subproblem is addressed only once and the DP technique stores the corresponding result. Finally, the DP technique can exactly provide optimal solutions to readers by combining previously saved results.

In DP model, $T_w(v, K')$ $(K' \subseteq K = Q)$ is a state, which covers a set of query keywords K'. Furthermore, $w(T_w(v, K'))$ denotes the weight of $T_w(v, K')$. The model state transitions equations are defined as follows.$w(\underset{\forall v \in V_p}{T_{w\min}}(v, K')) = \infty$ if $\underset{\forall v \in V_p}{T_{w\min}}(v, K')$ does not contain any query keywords:

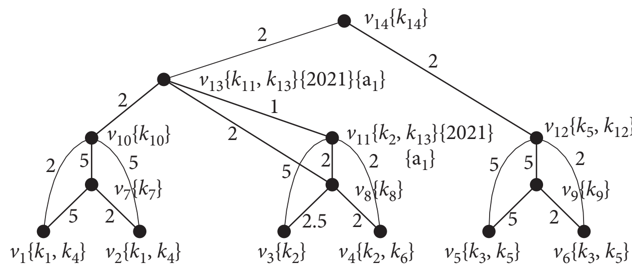


Figure 6: The W-PCG converted from Figure 5.

$$w\left(\underset{\forall v\in V_k\subseteq V_p}{T_{w\min}}(v,K')\right) = 0 \text{ if} \left|\underset{\forall v\in V_k\subseteq V_p}{T_{w\min}}(v,K')\right| = 1, \tag{4}$$

$$w(T_{w\min}(v,K')) = \min\left(w\left(T_{wg}(v,K')\right), w(T_{wm}(v,K'))\right), \tag{5}$$

$$w\left(T_{wg}(v,K')\right) = \min_{u\in N(v)}\left\{w(T_{w\min}(u,K'_u)) + w_{u,v}\right\}, \tag{6}$$

$$w\left(T_{wg}(v,K')\right) = \min_{\left(\substack{K'_u\cap K'\neq K'_u \,\&\&K'_u\cap K' = K' \\ \| \; K'_u\cap K'\neq K' \,\&\&K'_u\cap K' = K'_u}\right)} \{w(T_{w\min}(v,K')) + w(T_{w\min}(u,K'_u))\}, \tag{7}$$

$$w(T_{wm}(v,K')) = \min_{\left(\substack{K'_u\subseteq K\&\&K'\subseteq K\&\& \\ K'_u\cap K'\neq K'\&\&K'_u\cap K'\neq K'_u}\right)} \{w(T_{w\min}(v,K')) + w(T_{w\min}(u,K'_u))\}, \tag{8}$$

where $N(v)$ is a set of neighbor nodes of node v; u, v $\in\in G_w(V_p, E_p, W_p)$, $u \in\in N(v)$, $e(u,v) \in E_p$, and $w_{u,v} \in\in W_p$. Since tree $T_{w\min}$ (v, K') does not contain any query keywords, the weight of the tree is infinite in formula (3). In formula (10), a tree covers one query nodes and its weight is 0. Formula (5) denotes that tree $T_{w\min}$ (v, K') is acquired by doing the following operations: tree growth operation (i.e., formulas (7) and (8)) and tree merging operation (i.e., formulas (9) and (10)). In Figure 7(a), the tree growth operation creates new tree T_w (v, K') by adding new node u, and the pseudocode of the tree growth operation is specified in Algorithm 1. In Figure 7(b), the tree merging operation creates new tree T_w (v, K') by merging two trees; these two trees are both rooted at v. And the pseudocode of the tree merging operation is specified in Algorithm 2. When $K' = K$, $LP - PR_{k+w}$ searches for a minimum group weighted Steiner tree $T_{w\min}$ (v, K') in formula (11), and $T_{w\min}$ (v, K') contains at least one node.

$$w(T_{wm}(v,K')) = \min_{\substack{K'_1\cap K'_2=\Phi \\ K'_1\cup K'_2=K'}} \{w(T_{w\min}(v,K'_1)) + w(T_{w\min}(v,K'_2))\},$$
$$T_w(Q) = T_{w\min}(v,K') \text{ if } K' = K. \tag{9}$$

According to formulas (4)–(11), T_w (Q) is returned by repeating the tree growth and the tree merging operations, and the pseudocode of the MGWST algorithm is specified in Algorithm 3. Furthermore, the algorithm maintains two queues: Q^1 and Q^2: the queue Q^1 records intermediate trees only containing partial query keywords, and the queue Q^2 records qualified trees covering all query keywords K.

Note that, in the worst-case scenario, the MGWST algorithm may return the entire W-PCG as its output. In addition, the algorithm does not take into account the role of

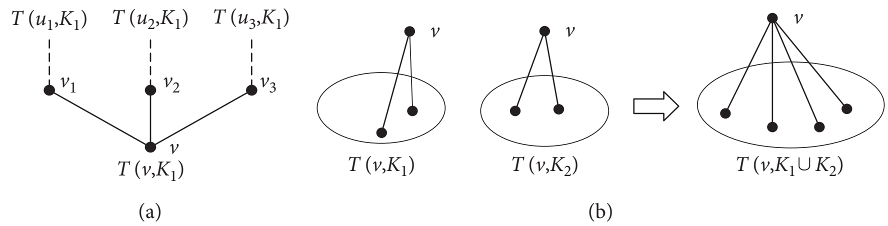


Figure 7: Tree operations. (a) Tree growth. (b) Tree merging.

Input: $K = \{k_1, k_2, \ldots, k_l\}$, $Q^1 = \Phi$
Output: Q^1
For each u $\in$ **N**(v) **do**
    If $w_{u,v} + w(T_{w\min}(u, K_u')) < w(T_{w\min}(v, K'))$
        $T_{w\min}(v, K') = e(u, v) + T_{w\min}(u, K_u')$
        enqueue $T_{w\min}(v, K')$ **into Q1**
        update Q1
    End If
    If $(K_u' \cap K' \neq K' \&\& K_u' \cap K' = K_u' \,\|\, K_u' \cap K' \neq K_u' \&\& K_u' \cap K' = K')$
        $w(T_{w\min}(u, K_u')) + w(T_{w\min}(v, K')) < w(T_{w\min}(v, K'))$
    $w(T_{w\min}(v, K')) = w(T_{w\min}(u, K_u')) + w(T_{w\min}(v, K'))$
        enqueue $T_{w\min}(v, K')$ **into Q1**
        update Q1
    End If
    Return Q1
End For

Algorithm 1: Tree growth.

Input: $K = \{k_1, k_2, \ldots, k_l\}$, **Q1** $= \Phi$
Output: Q1
For each u $\in$ **N**(v) **do**
    If $(K_u' \subseteq K \&\& K' \subseteq K \&\& K_u' \cap K' \neq K_u' \&\& K_u' \cap K' \neq K')$
        $T_{w\min}(v, K') = T_{w\min}(u, K_u') + T_{w\min}(v, K')$
        enqueue $T_{w\min}(v, K')$ **into Q1**
        update Q1
    End If
End For
$K_1' = K'$
    For each $T_{wmwminin}(v, K_2')$ **in Q1 s.t**$(K_1' \cap K_2' = \Phi \,\&\&\, K_1' \bigcup K_2' = K')$
        If $w(T_{w\min}(v, K_1')) + w(T_{w\min}(v, K_2')) < w(T_{w\min}(v, K_1' \bigcup K_2'))$
            $T_{w\min}(v, K_1' \bigcup K_2') = T_{w\min}(v, K_1') \oplus T_{w\min}(v, K_2')$
            enqueue $T_{w\min}(v, K_1' \bigcup K_2')$ **into Q1**
            update Q1
    End If
    Return Q1
End for

Algorithm 2: Tree merging.

the synonymy and word inflections in the paper recommendation [29–35] process.

6. Experiments

To evaluate the usefulness and feasibility of $LP - PR_{k+w}$, the large-scale experiments are tested on Hep-Th dataset [36].

6.1. Experimental Environment

Dataset. We select part of the Hep-Th data set for the experiment (i.e., these papers are published between 1997 and 2003); the partial data contains 8721 papers and forms a paper citation graph. Here, each paper node stores paper published time and authors' information. Furthermore, the keywords' information of each paper is constructed by employing the RAKE (rapid automatic keyword extraction) technique.

Experiment Settings. To obtain an optimal W-PCG (i.e., G_w), we execute the link prediction process depicted in Figure 4. Furthermore, we also set the parameters values of the KTA weighting criterion: α, $\beta \in \{0.3, 0.5, 0.7, 0.9\}$, $\lambda \in \{0.3, 0.9\}$.

In the keywords query experiments, three diverse sets of the keywords are set, i.e., set A, set B, and set C. In set A, the keywords of one paper are regarded as the query keywords, where readers can provide all query keywords for their research content. In set B, the query keywords are randomly selected from different papers (in excess of one paper) as the readers' research content covers diverse research topics. In set C, the query keywords are randomly selected from any two papers to further verify the feasibility of the MGWST algorithm. Furthermore, an author of each paper basically creates the number of keywords with up to 6, so there are up to six query keywords in each set. Each experiment is repeated 100 times and the average experiment results are adopted.

Evaluation Criteria. We compare the following evaluation criteria:

(1) Number of new edges: the larger, the better; that is, more new edges denote that our link prediction approach can better solve the sparse problems of the existing paper citation graph.

(2) Number of nodes: number of recommended papers (the smaller, the better, i.e., fewer papers denote that high correlation). Note that the return solution contains at least one paper.

(3) Success rate [37]: when the number of papers of a recommendation result is less than twice the number of the readers' query keywords, the recommendation result is successful (the larger, the better).

(4) Weight: the weight of a recommended result (the smaller, the better).

(5) Computation time: the time for generating an answer tree (the smaller, the better). Here, the computation time can be well described by logarithmic function (log_2).

(6) Precision [20] is defined as

$$\text{precision} = \frac{TP}{R_p}, \tag{10}$$

where TP denotes a set of papers containing query keywords and R_p denotes a recommendation result.

(7) Recall [38] is defined as

$$\text{recall} = \frac{1}{|p|} \sum_{p_a \in p} \frac{R_p \cap T_{p_a}}{T_{p_a}}, \tag{11}$$

where $|p|$ is the number of papers (e.g., $|p| = 2$) in set C. T_{p_a} is a set of papers cited by p_a.

(8) $F1$ score is defined as

$$F1\ \text{score} = \frac{2 * \text{recall} * \text{precision}}{\text{recall} + \text{precision}}. \tag{12}$$

Here, we compare $LP - PR_{k+w}$ with several similar paper recommendation approaches:

Baseline 1. Link Prediction-Random (*LP-Random*) [28]. The approach is randomly finding a set of nodes from the G_w, and these nodes collectively cover readers' query keywords. Finally, this approach grows a minimum weighted spanning tree.

Baseline 2. Link Prediction-Greedy (LP-Greedy) [28]. Likewise, the approach selects a set of nodes contained all query keywords from the G_w. Next, this approach regards these nodes as the initial root nodes and continuously generates a tree until these nodes are interconnected.

Baseline 3. Link Prediction-Random Walk (LP-RW) [13]. First, using papers' keywords and the correlation relationships of G_w to build 2-layer graph, and then the approach runs on the 2-layer graph to recommend papers. Furthermore, each query only uses readers' entered keywords: $q = [0, qW]$.

Baseline 4. Link Prediction-Random Walk Restart (LP-RWR) [12]. Likewise, the approach also uses the 2-layer graph for paper recommendation. If the state vectors of LP-RWR steadily grows linearly in the experiment, then we consider the approach achieves linear convergence.

Experimental Tools. All experiments are implemented by python and executed with Intel® Core® CPU @ 3.0 GHz, 16 GB RAM, and Windows 10 @ 1809, 64-bit operating system.

6.2. Experimental Results

(1) *Profile 1: The Number of New Edges*.

In this profile, we compare the number of new edges to select a set of appropriate parameters values that are better at solving the sparsity of an existing paper citation graph in some extent. Tables 2–3 present that we mostly gain different experimental results as paper keywords, paper published time, and paper authors' information have a significant role in the link prediction process. According to the above experiments results, when $\alpha = 0.3$, $\beta = 0.5$, and $\lambda = 0.9$, we get the best experimental results; that is, the number of new edges is 348. Therefore, we select the set of appropriate parameters values to build a W-PCG.

(2) *Profile 2: The Number of Nodes of Different Approaches*.

Input: $\mathbf{K} = \{k_1, k_2, \ldots, k_l\}$, **Q1** $= \Phi$, **Q2** $= \Phi$
Output: **Q1, Q2,** $T_{w\min}(v, K)$
Let Q1 $= \Phi$, **Q2** $= \Phi$
For each $v \in V_p$ **do**
  If v contains any nonempty keyword set $K' \subset K$
    $w(T_{w\min}(v, K')) \neq \infty$
    enqueue $T_{w\min}(v, K')$ **into Q1**
  End If
End For
While Q1 $\neq \Phi$ **do**
  dequeue Q1 to $T_{w\min}(v, K')$
  If $K' = \mathbf{K}$
    enqueue $T_{w\min}(v, K')$ **into Q2**
    Continue
  End If
  Else tree growth
  Else tree merging
  Return $T_{w\min}(v, K) =$ **Q2.top ()**
End While

ALGORITHM 3: **MGWST** (G_w, K).

TABLE 2: α, β, and $\lambda = 0.3$ are employed in the link prediction to obtain the number of new edges.

$\alpha\beta$	0.3	0.5	0.7	0.9
0.3	190	192	136	160
0.5	264	270	276	238
0.7	258	258	264	310
0.9	258	258	258	264

TABLE 3: α, β, and $\lambda = 0.9$ are employed in the link prediction to obtain the number of new edges.

$\alpha\beta$	0.3	0.5	0.7	0.9
0.3	208	152	160	168
0.5	348	234	216	176
0.7	342	342	314	226
0.9	288	288	288	294

In the experiment, we test the number of recommended nodes (papers) of different approaches. Here, the number of readers'query keywords ranges from 2 to 6. As shown in Figure 8, these experiment results show that the number of recommended including more papers can satisfy requirements of readers on more keywords query. When readers can accurately provide all query keywords, Figure 8(a) presents that our approach can accurately answer readers' keywords query; that is, the number of papers is 1. Furthermore, the experiment results of Figure 8(b) demonstrate that $LP - PR_{k+w}$ can find necessary bridging nodes between target nodes. In the case of same keywords query, Figure 8 shows that the number of recommended papers of $LP - PR_{k+w}$ is fewer than that of the other two approaches (i.e., *LP-Random* and *LP-Greedy*). In fact, a recommended result contains a handful of papers, which means these recommended papers have higher correlation. Thus, the experiment results of Figure 8 can show directly that our paper recommendation approach is superior to *LP-Random* and *LP-Greedy*.

(3) *Profile 3: The Success Rate of Different Approaches.* As shown in Figure 9, we compare the success rate of different approaches in the different sets. According to [39], if the number of query keywords equals 6, then the number of papers of a successful recommended result must range from 1 to 12. When readers can accurately provide all query keywords, Figure 9(a) presents that the success rate of our approach is 100%. For the other keywords query cases, the experiment results of Figures 9(b) and 9(c) again present that the success rate of $LP - PR_{k+w}$ is 100%. However, *LP-Random* and *LP-Greedy* are difficult to obtain successful paper recommendation results, especially with the number of query keywords equal to 6. In conclusion, the experiment

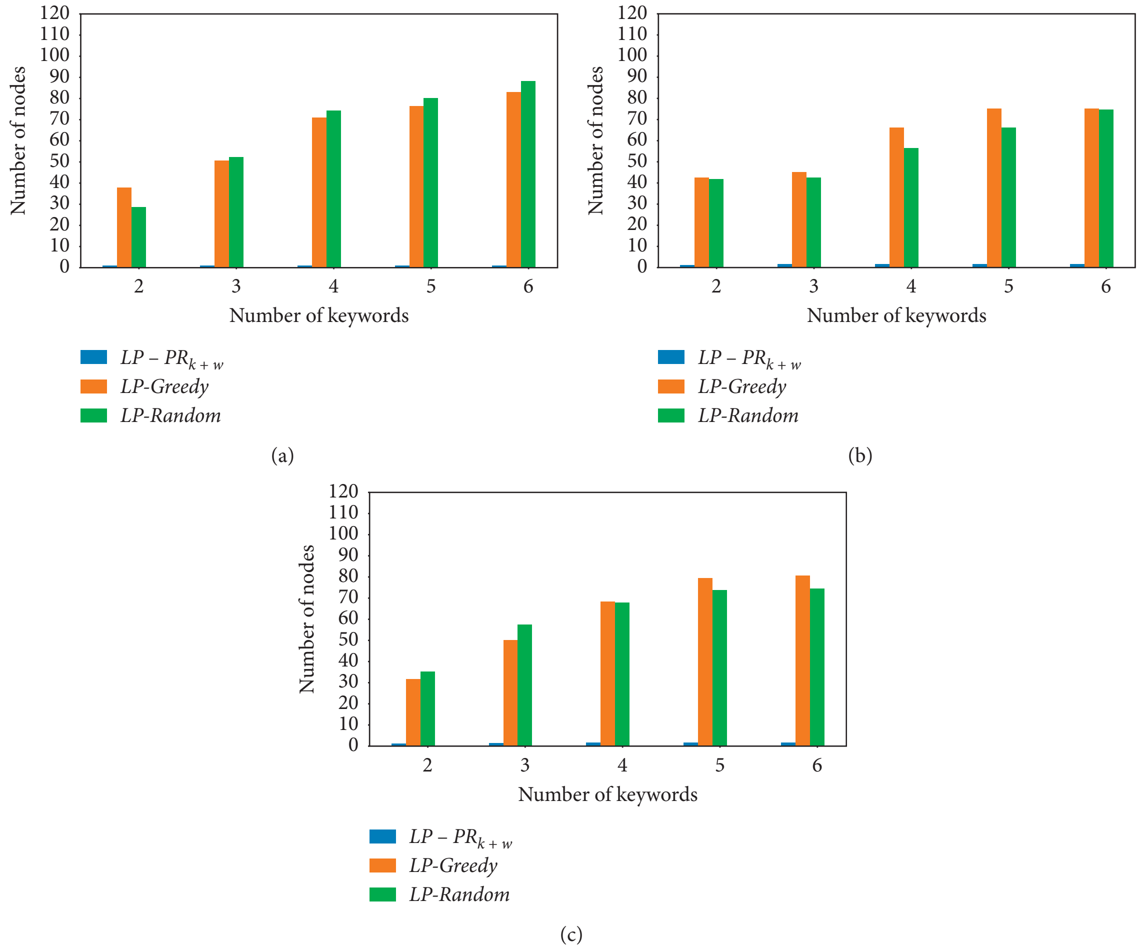


Figure 8: The number of recommended nodes of different approaches. (a) The number of recommended nodes in set A. (b) The number of recommended nodes in set B. (c) The number of recommended nodes in set C.

results of the set C can verify the feasibility of our paper recommendation algorithm (i.e., the MGWST algorithm). Furthermore, the experiment results of Figure 9 can show that $LP-PR_{k+w}$ can recommend more sets of papers with the higher correlations than other two approaches.

(4) *Profile 4: The Weight of Different Approaches.*

In the paper, $LP-PR_{k+w}$, *LP-Random*, and *LP-Greedy* are both returning a tree containing all query keywords to readers. Furthermore, the weight of a tree can reflect the correlation of a set of papers; the smaller the weight, the higher the correlation. Hence, we compare the weight of different approaches in this experiment. As shown in Figure 10, these experiment results show that the weights of different approaches increase with the number of query keywords increasing, as these approaches need to find more papers to meet readers' keywords query requirements. In the case of the same keywords query, the weight of our approach is less than the other two approaches (i.e., *LP-Random* and *LP-Greedy*) as $LP-PR_{k+w}$ adopts the weight optimization strategy. Furthermore, Figure 10(a) shows that the weight of $LP-PR_{k+w}$ equals 0, which further shows that our proposal can find one corresponding paper containing all query keywords. According to the above experiment results, $LP-PR_{k+w}$ can guarantee to return a set of papers with the minimum weight; that is, these papers have the higher correlation.

(5) *Profile 5: The Computation Time of Different Approaches.*

In the paper, a computation time is defined as the time of finding paper recommendation results. Thus, we compare the computation time of different approaches. As *LP-RW* and *LP-RWR* only do iterative operations and matrix operations in the process of paper recommendation, the computation time of these two approaches are both fixed values in the sets B and C.

Moreover, we only count the time when *LP-RW* achieves linear convergence, so the computation time of the approach is smaller than that of our approach. As shown in Figure 11, the computation time of $LP-PR_{k+w}$, *LP-Random*, and *LP-Greedy* generally increases as the number of query keywords increases; that is, these three approaches all take

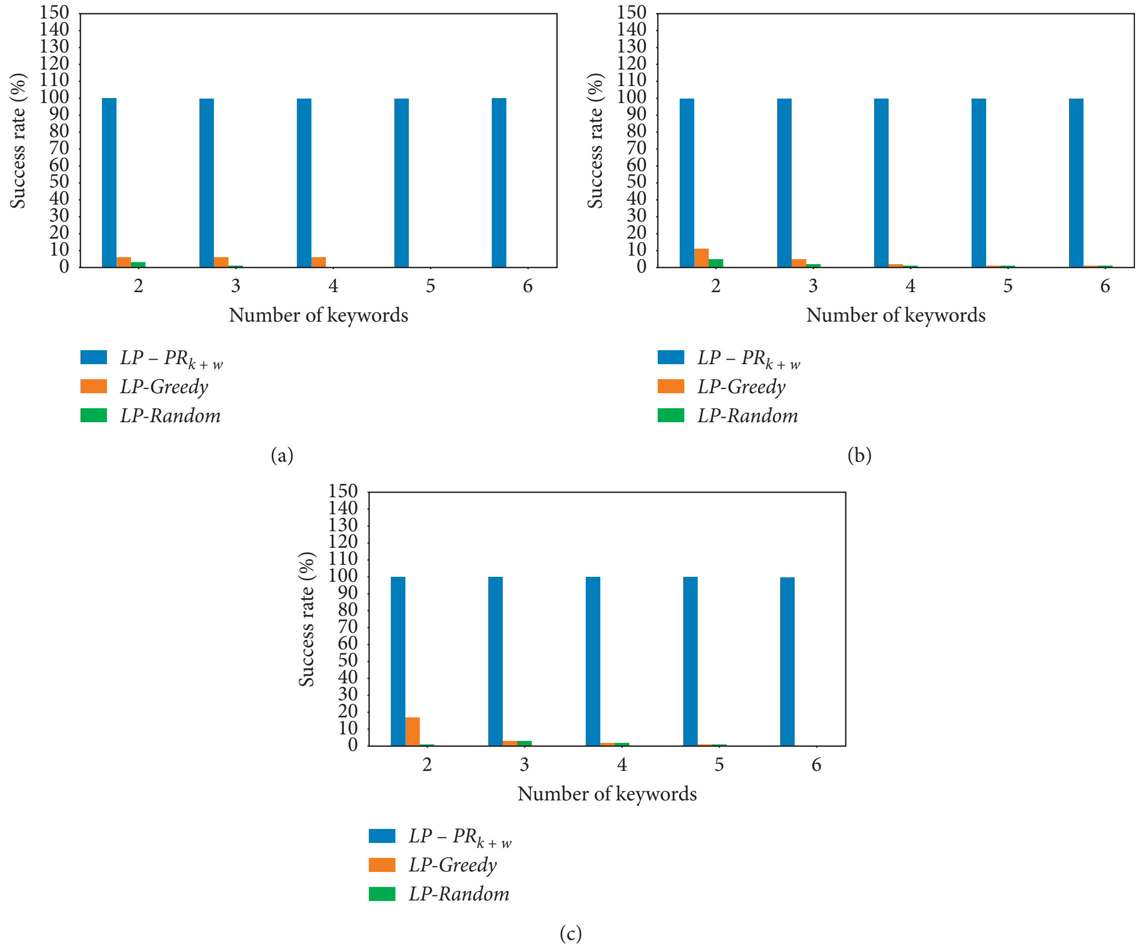


FIGURE 9: The success rate of different approaches. (a) The success rate in set A. (b) The success rate in set B. (c) The success rate in set C.

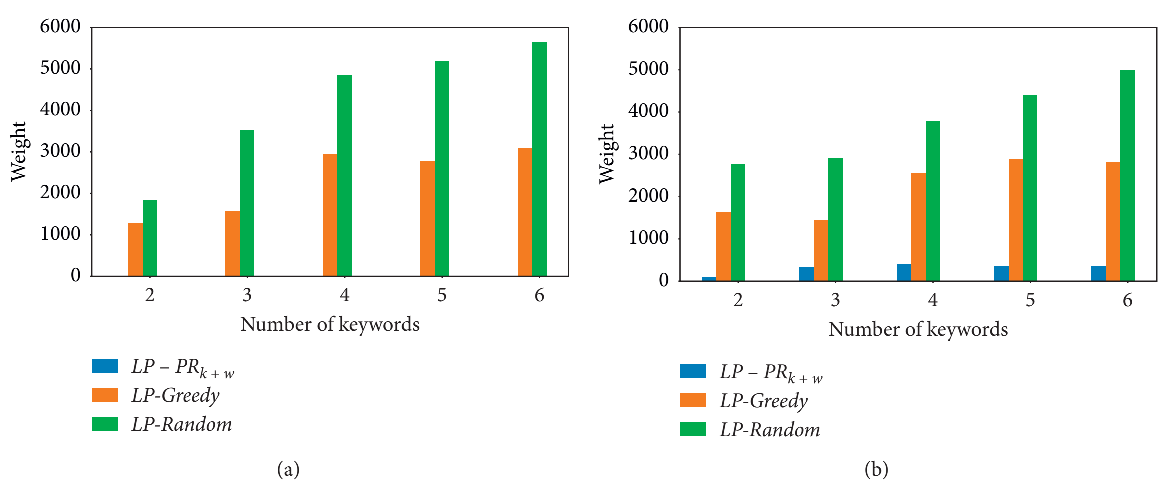
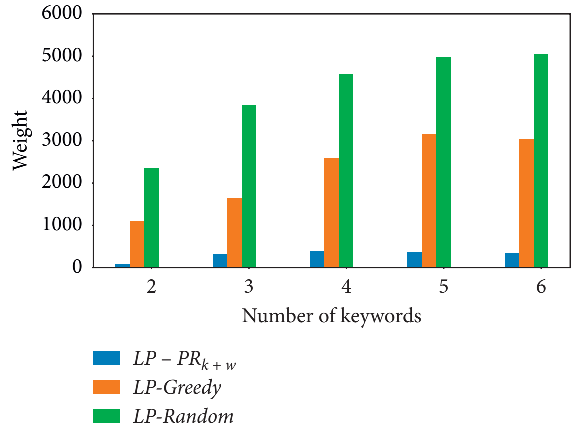
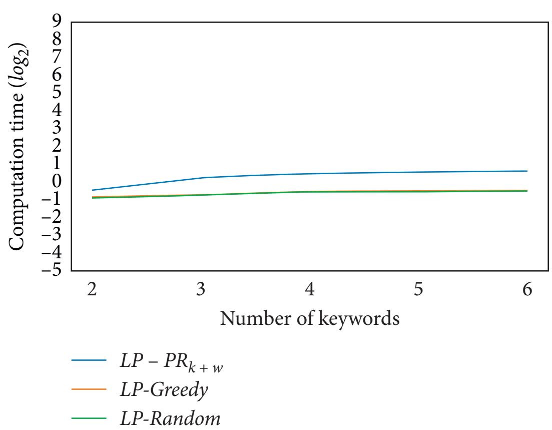


FIGURE 10: Continued.

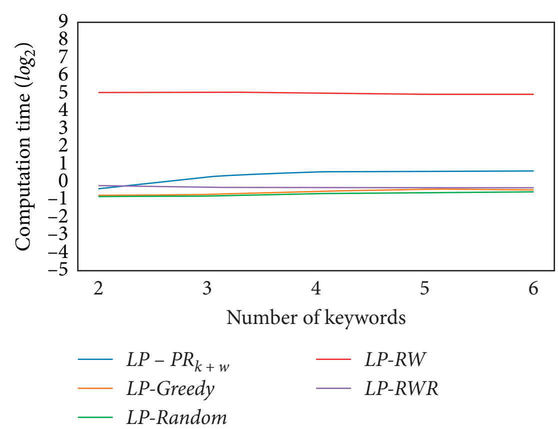


(c)

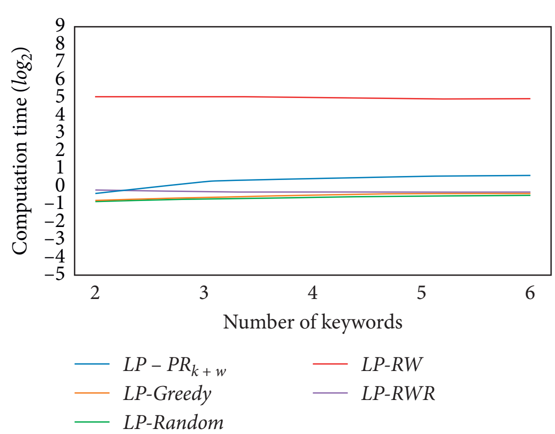
FIGURE 10: The weight of different approaches. (a) The weight in set A. (b) The weight in set B. (c) The weight in set C.



(a)



(b)



(c)

FIGURE 11: The computation time of different approaches. (a) The computation time in set A. (b) The computation time in set B. (c) The computation time in set C.

more time to find an answer tree with the increasing number of query keywords. Furthermore, the computation time of these three approaches increases exponentially. As *LP-Random* and *LP-Greedy* both use extremely simple heuristic for selecting the bridging nodes from G_w, these two approaches find the answer tree faster than our approach. In fact, the recommended results of *LP-Random* and *LP-Greedy*

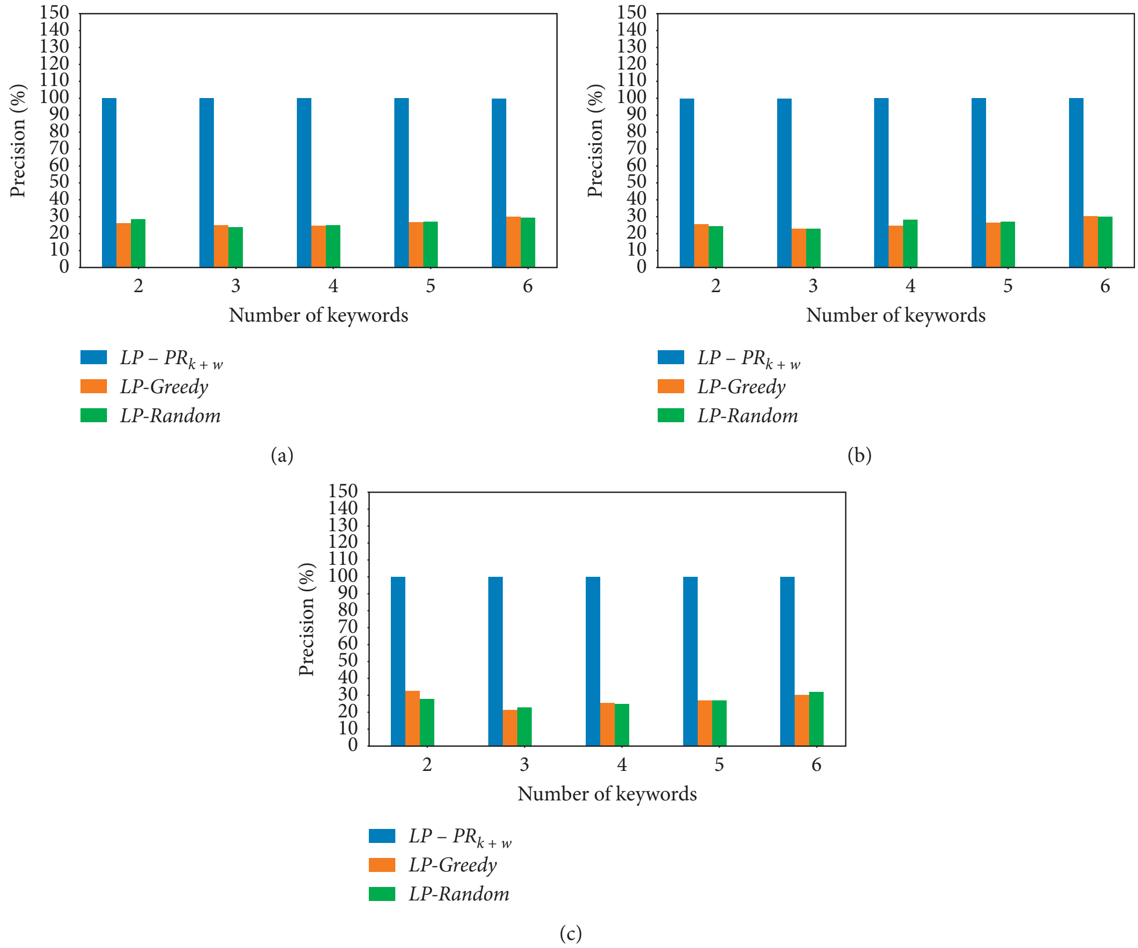


FIGURE 12: The precision of different approaches. (a) The precision in set A. (b) The precision in set B. (c) The precision in set C.

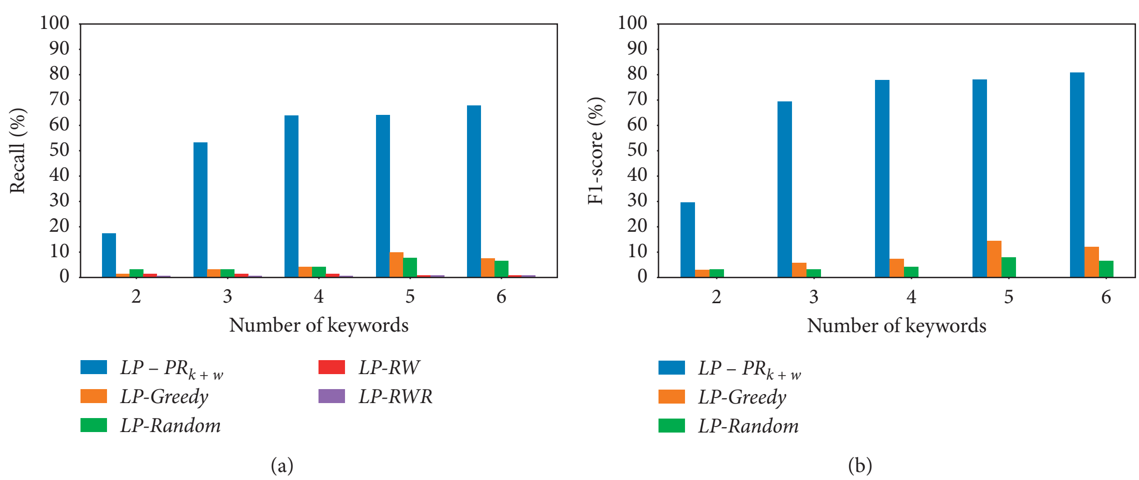
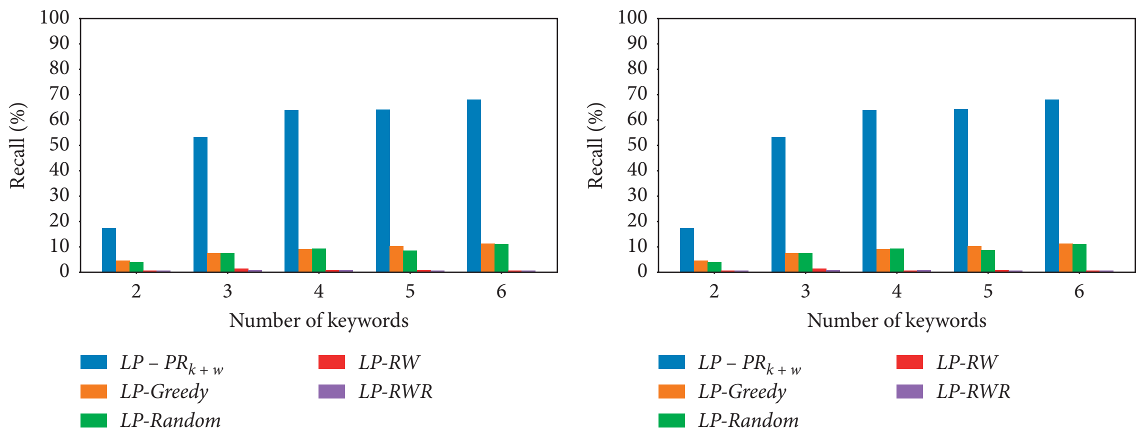


FIGURE 13: Continued.



(c) (d)

FIGURE 13: The recall and $F1$ score of different approaches. (a) The recall in set B. (b) The $F1$ score in set B. (c) The recall in set C. (d) The $F1$ score in set C.

are not ideal for readers as their recommended results contain some unnecessary bridging nodes (see Figure 8). In most really cases, the computation time of $LP-PR_{k+w}$ is allowable and receivable, which is the price to pay if readers save more time and energy on achieving their research goal.

(6) *Profile 6: The Precision of Different Approaches.*

For diverse paper recommendation approaches, we compare the precision of their recommended results. As shown in Figure 12, these experiment results show that the precision of diverse approaches is very different in the same keywords query cases. Whether readers accurately or randomly provide their query keywords, $LP-PR_{k+w}$ can accurately answer the readers' keywords query, and the precision equals 100%. However, the precision of *LP-Random* and *LP-Greedy* has both a range from 22% to 33% in the experiment. Therefore, the experiment results of Figure 12 further show that readers will easily realize their research aim by using our recommended results.

(7) *Profile 7: The Recall and F1 Score of Different Approaches.*

In this profile, we further compare the recall and $F1$ score of different approaches in sets B and C. Here, the numbers of recommended papers of *LP-RW* and *LP-RWR* are both equal to 20. As can be seen Figures 13(a) and 13(b), the recall and $F1$ score of $LP-PR_{k+w}$ range from 17% to 69% and from 29% to 81%, respectively; meanwhile, the other four approaches have smaller recall and $F1$ score than our approach. Likewise, Figures 13(c) and 13(d) also show that the recall and $F1$ score of other four approaches are smaller than $LP-PR_{k+w}$. Therefore, the experiment results of Figure 13 are not only demonstrating that our proposal can satisfy readers' keywords query requirements but also verifying further the feasibility of the MGWST algorithm (i.e., Algorithm 3).

7. Conclusions

In this paper, a novel keywords-driven and weight-aware paper recommendation approach based on the mobile edge computing framework, that is, $LP-PR_{k+w}$, is put forward to address the sparsity and compatibility issues existing in paper recommendations. The recommended papers are not only collectively containing readers' query keywords but also having the highest correlation degree. Therefore, the recommended papers can significantly promote readers' in-depth and continuous research around an identical focus. Furthermore, our proposal indirectly solves the problem of latency of recommendation results by adjusting the order of paper recommendation in the mobile edge computing environment. Finally, the experimental results show the feasibility of $LP-PR_{k+w}$, in terms of multiple evaluation metrics. In summary, our proposal can provide intelligent and personalized paper recommendation services in the mobile edge computing environment and further promote the quality-of-retrieval experience of readers.

Although our work shows desirable results, there are still several issues remaining unsolved. Firstly, our adopted link prediction approach in recommender system [40–46] is a bit naive and straightforward. Therefore, more refinements are needed in the future. Second, readers' paper search is often a multiobjective decision-making problem that involves a number of influencing factors [47–54] in the mobile edge computing environment. We will further improve our proposal by considering these factors.

Data Availability

The experiment dataset Hep-Th used to support the findings of this study has been deposited in "http://snap.stanford.edu/data/cit-HepTh.html."

Additional Points

This work is intended to improve and perfect the model in [40]. Thus, the work enables intelligent, personalized, and efficient paper recommendation services in the mobile edge computing environment.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Key Research Base of Philosophy and Social Sciences in Jiangsu Universities "Academic Center of Huang Yanpei Vocational Education Thought Research Association," Jiangsu University Philosophy and Social Science Research Project (2020SJA0675), Scientific Research Project of Nanjing Vocational University of Industry Technology (2020SKYJ03), Jiangsu Province Modern Education Technology Research Project (2020-R-84365), and National Vocational Education Teacher Enterprise Practice Base "Integration of Industry and Education" Special Project (Study on Evaluation Standard of Artificial Intelligence Vocational Skilled Level).

References

[1] J. Huang, B. Lv, Y. Wu, Y. Chen, and X. Shen, "Dynamic admission control and resource allocation for mobile edge computing enabled small cell network," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 2, pp. 1964–1973, 2022.

[2] Y. Gong, Z. Jiang, Y. Feng et al., "EdgeRec," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 2477–2484, Association for Computing Machinery, New York, NY, USA, October 2020.

[3] A. Xiong, D. Liu, H. Tian, Z. Liu, P. Yu, and M. Kadoch, "News keyword extraction algorithm based on semantic clustering and word graph model," *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 886–893, 2021.

[4] T. Murata and S. Moriyasu, "Link Prediction of Social Networks Based on Weighted Proximity Measures," in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI'07)*, pp. 85–88, Fremont, CA, USA, November 2007.

[5] Z. Wang, J. Liang, R. Li, and Y. Qian, "An approach to cold-start link prediction: establishing connections between non-topological and topological information," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 11, pp. 2857–2870, 2016.

[6] J. Huang, Z. Tong, and Z. Feng, "Geographical POI recommendation for internet of things: a federated learning approach using matrix factorization," *International Journal of Communication Systems*, 2022.

[7] S. M. McNee, I. Albert, D. Cosley et al., "On the recommending of citations for research papers," in *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work (CSCW '02)*, pp. 116–125, Association for Computing Machinery, NY, USA, November 2002.

[8] S. Meng, L. Qi, Q. Li, W. Lin, X. Xu, and S. Wan, "Privacy-preserving and sparsity-aware location-based prediction method for collaborative recommender systems," *Future Generation Computer Systems*, vol. 96, pp. 324–335, 2019.

[9] L. Wang, X. Zhang, R. Wang, C. Yan, H. Kou, and L. Qi, "Diversified service recommendation with high accuracy and efficiency," *Knowledge-Based Systems*, vol. 204, Article ID 106196, 2020.

[10] P. Nitu, J. Coelho, and P. Madiraju, "Improvising personalized travel recommendation system with recency effects," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 139–154, 2021.

[11] Y. Xu, L. Qi, W. Dou, and J. Yu, "Privacy-preserving and scalable service recommendation based on SimHash in A distributed cloud environment," *Complexity*, vol. 2017, p. 9, Article ID 3437854, 2017.

[12] F. Meng, D. Gao, W. Li, X. Sun, and Y. Hou, "A unified graph model for personalized query-oriented reference paper recommendation," in *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management (CIKM '13)*, pp. 1509–1512, Association for Computing Machinery, NY, USA, October 2013.

[13] M. Gori and A. Pucci, "Research paper recommender systems: a random-walk based approach," in *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)*, pp. 778–781, Hong Kong, China, December 2006.

[14] H. Liu, H. Kou, C. Yan, L. Qi, and P. Link, "Paper citation network to construct paper correlation graph," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, pp. 1–12, 2019.

[15] R.-H. Li, J. X. Yu, and J. Liu, "Link prediction," *Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11*, Association for Computing Machinery, in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM'11)*, pp. 1147–1156, October 2011.

[16] Y. Chen, W. Gu, and K. Li, "Dynamic task offloading for internet of things in mobile edge computing via deep reinforcement learning," *International Journal of Communication Systems*, 2022.

[17] L. Backstrom and J. Kleinberg, "Romantic partnerships and the dispersion of social ties," *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, Association for Computing Machinery, in *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '14)*, pp. 831–841, February 2014.

[18] C. P. Muniz, R. Goldschmidt, and R. Choren, "Combining contextual, temporal and topological information for unsupervised link prediction in social networks," *Knowledge-Based Systems*, vol. 156, pp. 129–137, 2018.

[19] R. Xiang, J. Neville, and M. Rogati, "Modeling relationship strength in online social networks," in *Proceedings of the 19th International Conference on World Wide Web*, pp. 981–990, Association for Computing Machinery, NY, USA, April 2010.

[20] P. M. Chuan, L. H. Son, M. Ali, T. D. Khang, L. T. Huong, and N. Dey, "Link prediction in co-authorship networks based on hybrid content similarity metric," *Applied Intelligence*, vol. 48, no. 8, pp. 2470–2486, 2018.

[21] P. R. S. Soares and R. B. C. Prudêncio, "Proximity measures for link prediction based on temporal events," *Expert Systems with Applications*, vol. 40, no. 16, pp. 6652–6660, 2013.

[22] B. Filipecki and M. Van Vyve, "Stronger path-based extended formulation for the Steiner tree problem," *Networks*, vol. 75, no. 1, pp. 3–17, 2020.

[23] N. Löhndorf and A. Shapiro, "Modeling time-dependent randomness in stochastic dual dynamic programming," *European Journal of Operational Research*, vol. 273, no. 2, pp. 650–661, 2019.

[24] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[25] M. Lu, Z. Qu, M. Wang, and Z. Qin, "Recommending authors and papers based on ACTTM community and bilayer citation network," *China Communications*, vol. 15, no. 7, pp. 111–130, 2018.

[26] H. Liu, H. Kou, X. Chi, and L. Qi, "Combining time, keywords and authors information to construct papers correlation graph," in *Proceedings of the The 27th International Conference on Software Engineering and Knowledge Engineering (SEKE'2019)*, pp. 11–19, USA, July 2019.

[27] Y. Chen, F. Zhao, Y. Lu, and X. Chen, "Dynamic task offloading for mobile edge computing with hybrid energy supply," *Tsinghua Science and Technology*, 2021.

[28] L. Qi, Q. He, F. Chen et al., "Finding all you need: web APIs recommendation in web of things through keywords search," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 5, pp. 1063–1072, 2019.

[29] S. Wang, L. Hu, Y. Wang, L. Cao, Q. Z. Sheng, and M. Orgun, "Sequential recommender systems: challenges, progress and prospects," *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019.

[30] S. Wang, L. Hu, Y. Wang et al., "Graph learning based recommender systems: a review," *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, Survey Track, , January 2021.

[31] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, "Generative adversarial networks," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–38, 2021.

[32] X. Zhou, X. Yang, J. Ma, and K. I.-K. Wang, "Energy efficient smart routing based on link correlation mining for wireless edge computing in IoT," *IEEE Internet of Things Journal*1 page, 2021.

[33] J. Hu, Y. Pan, T. Li, and Y. Yang, "TW-Co-MFC: two-level weighted collaborative fuzzy clustering based on maximum entropy for multi-view data," *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 185–198, 2021.

[34] X. Zhou, Y. Li, and W. Liang, "CNN-RNN based intelligent recommendation for online medical pre-diagnosis support," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 3, pp. 912–921, 2021.

[35] Z. Xue and H. Wang, "Effective density-based clustering algorithms for incomplete data," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 183–194, 2021.

[36] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time," in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining (KDD '05)*, pp. 177–187, Association for Computing Machinery, NY, USA, August 2005.

[37] H. Liu, H. Kou, C. Yan, and L. Qi, "Keywords-driven and popularity-aware paper recommendation based on undirected paper citation graph," *Complexity*, vol. 2020, Article ID 2085638, 2020.

[38] L. Guo, X. Cai, F. Hao, D. Mu, C. Fang, and L. Yang, "Exploiting fine-grained co-authorship for personalized citation recommendation," *IEEE Access*, vol. 5, pp. 12714–12725, 2017.

[39] L. Qi, Q. He, F. Chen, X. Zhang, W. Dou, and Q. Ni, "Data-driven web APIs recommendation for building web applications," *IEEE Transactions on Big Data*, IEEE Transactions on Big Data1 page, IEEE Transactions on Big Data, 2020.

[40] H. Liu, J. Hou, Q. Li, and J. Jiang, "Keywords-driven and weight-aware paper recommendation via paper correlation pattern mining," *Research Square*, 2021.

[41] S. Wang, L. Cao, Y. Wang, Q. Z. Sheng, M. A. Orgun, and D. Lian, "A survey on session-based recommender systems," *ACM Computing Surveys*, vol. 54, no. 7, pp. 1–38, 2022.

[42] J. Li, H. Peng, Y. Cao et al., "Higher-order attribute-enhancing heterogeneous graph neural networks," *IEEE Transactions on Knowledge and Data Engineering*1 page, 2021.

[43] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2020.

[44] X. Zhou, W. Liang, W. Li, K. Yan, S. Shimizu, and K. I.-K. Wang, "Hierarchical adversarial attacks against graph neural network based IoT network intrusion detection system," *IEEE Internet of Things Journal*1 page, 2021.

[45] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.

[46] Z. Cai and Z. He, "Trading private range counting over big IoT data," *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, in *Proceedings of the The 39th IEEE International Conference on Distributed Computing Systems (ICDCS)*, Dallas, TX, USA, July 2019.

[47] J. Huang, C. Zhang, and J. Zhang, "A multi-queue approach of energy efficient task scheduling for sensor hubs," *Chinese Journal of Electronics*, vol. 29, no. 2, pp. 242–247, 2020.

[48] W. Fan, P. Chen, D. Shi, X. Guo, and L. Kou, "Multi-agent modeling and simulation in the AI age," *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 608–624, 2021.

[49] Z. Tong, F. Ye, M. Yan, H. Liu, and S. Basodi, "A survey on algorithms for intelligent computing and smart city applications," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 155–172, 2021.

[50] X. Zhou, W. Liang, K. I.-K. Wang, and L. T. Yang, "Deep correlation mining based on hierarchical hybrid networks for heterogeneous big data recommendations," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 1, pp. 171–178, 2021.

[51] M. Wang, W. Han, and W. Chen, "MetaOJ: a massive distributed online judge system," *Tsinghua Science and Technology*, vol. 26, no. 4, pp. 548–557, 2021.

[52] H. Liu, S. Meng, J. Hou, S. Wang, Q. Li, and C. Huang, "Locality-sensitive hashing-based link prediction process on smart campus education or online social platform," *Journal of Circuits, Systems, and Computers*, Article ID 2250160, 2022.

[53] X. Zhou, X. Xu, W. Liang, Z. Zeng, and Z. Yan, "Deep-learning-enhanced multitarget detection for end-edge-cloud surveillance in smart IoT," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12588–12596, 2021.

[54] Y. Bie and Y. Yang, "A multitask multiview neural network for end-to-end aspect-based sentiment analysis," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 195–207, 2021.