Hindawi
Security and Communication Networks
Volume 2022, Article ID 1336094, 15 pages
https://doi.org/10.1155/2022/1336094

Research Article

Privacy-Aware Task Assignment for IoT Audit Applications on Collaborative Edge Devices

Linyuan Liu,[1] **Haibin Zhu**,[2] **Shenglei Chen**,[1] **and Zhiqiu Huang**[3]

[1] *Department of E-Commerce, Nanjing Audit University, Nanjing 211815, China*
[2] *Collaborative Systems Laboratory, Nipissing University, North Bay ON P1B8L7, Canada*
[3] *College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China*

Correspondence should be addressed to Linyuan Liu; liulinyuang@nau.edu.cn

Received 7 January 2022; Revised 18 April 2022; Accepted 1 June 2022; Published 21 June 2022

Academic Editor: Ke Gu

To meet the rapidly increasing demand for Internet of Things (IoT) applications, edge computing, as a novel computing paradigm, can combine devices at the edge of the network to collaboratively provide computing resources for IoT applications. However, the dynamic, heterogeneous, distributed, and resource-constrained nature of the edge computing paradigm also brings some problems, such as more serious privacy leakages and performance bottlenecks. Therefore, how to ensure that the resource requirements of the application are satisfied, while enhancing the protection of user privacy as much as possible, is a challenge for the task assignment of IoT applications. Aiming to address this challenge, we propose a privacy-aware IoT task assignment approach at the edge of the network. Firstly, we model the resource and privacy requirements for IoT applications and evaluate the resource satisfaction and privacy compatibility between edge devices and tasks. Secondly, we formulate the problem of privacy-aware IoT task assignment on edge devices (PITAE) and develop two solutions to the PITAE problem based on the greedy search algorithm and the Kuhn–Munkres (KM) algorithm. Finally, we conduct a series of simulation experiments to evaluate the proposed approach. The experimental results show that the PITAE problem can be solved effectively and efficiently.

1. Introduction

With the development of the Internet of Things (IoT), various IoT applications emerge as the times require, such as disaster relief, public safety, and face recognition [1–3]. According to Garner [4], the global IoT-enabled applications and infrastructure market will represent a 33 billion US dollar opportunity in 2025. IoT applications usually have a large amount of data that need to be processed in time. Hence, they have strict requirements on computing resources, response time, and privacy [5–7]. The traditional cloud-centric task processing model fails to meet these requirements, because it often needs to transmit a large amount of data to the cloud, which increases network transmission delay and network traffic [8].

To address the shortcomings of the cloud model, researchers have proposed edge computing [9, 10]. As a novel computing paradigm, edge computing can combine the resources of multiple devices at the edge of the network to provide task processing for IoT applications [11, 12]. With the wide adoption of wireless sensing and communication technology, a large number of IoT devices are emerging at the network edge, such as closed-circuit television (CCTV) cameras, smartphones, tablets, smart watches, smart home devices, and smart vehicles. Due to limited resources, these devices are generally only responsible for data collection and preprocessing, while complex data analysis work is offloaded to edge servers or cloud servers.

Supported by the advances in hardware and networking technologies, IoT devices are constantly increasing in resources and processing capabilities. They communicate with each other to collect and share data, and immediately process tasks near the data source [13, 14]. Edge computing has recently moved beyond the initial principle of utilizing IoT devices to collect and preprocess sensory data and is now able to combine and coordinate multiple IoT devices to provide processing for IoT applications [1, 14, 15]. Therefore, the advantages of edge computing such as low latency and local data processing are further highlighted [2, 16]. In this paper, we refer to these resource-constrained IoT devices with data collection and task processing capabilities as edge devices.

Due to the dynamic, heterogeneous, and distributed nature of the edge computing paradigm, edge devices are generally owned by individuals with different interests and affiliations [17]. As a result, the owner of edge devices may illegally use and disclose the user privacy information hidden in IoT data, e.g., faces, motions, locations, etc., resulting in serious privacy leakages [18, 19].

Consider a data-intensive IoT application consisting of multiple interrelated tasks, where each task has different resource requirements, e.g., CPU, memory, storage, bandwidth, etc. To protect user privacy, each private data in the task specify a set of privacy requirements. Correspondingly, each edge device has a set of available resources and provides a set of privacy policies. An important prerequisite for an edge device to be qualified to execute an IoT task is that it must satisfy the resource and privacy requirements of the task. Moreover, a single edge device is difficult to process relatively complex computations due to limited resources. Consequently, multiple tasks of an IoT application need to be assigned to multiple edge devices for execution. In summary, how to assign tasks to multiple edge devices that satisfy resource and privacy requirements is an important challenge in task assignment for IoT applications.

In the research of task assignment for IoT applications, some useful approaches were proposed to offload tasks to cloud, fog, and edge [1, 20–22]. However, most of them regard the task assignment from the perspective of resources and quality of service (QoS), while ignoring the privacy requirements of the users. Moreover, some researches focus on the privacy-aware IoT task assignment. They mainly adopt various privacy technologies like differential privacy, data generalization, task fragmentation, and privacy conflict avoidance to control data access [23–26], but they are inadequate to address the issue of how private data will be used after being accessed, such as the purpose of using the data, the retention time of the data, and the operations executed on the data.

Inspired by these works, in this paper, we propose a privacy-aware IoT task assignment approach at the edge of the network, which assigns IoT tasks to multiple edge devices close to the data source. These devices do not rely on a central coordinator and collaborate to process IoT tasks in a distributed manner. Specifically, we first model the resource and privacy requirements of the IoT tasks and evaluate whether the edge devices can satisfy these requirements. Then, we formulate the problem of privacy-aware IoT task assignment on edge devices (PITAE) as an optimization problem to maximize the privacy compatibility degree between IoT tasks and edge devices. Furthermore, we develop two solutions based on the greedy search algorithm and the KM algorithm [27, 28] to solve the problem. The main contributions of this paper are as follows:

(1) An integer programming optimization model is used to formulate the PITAE problem considering both the resource and privacy constraints.

(2) A privacy model is presented to specify the privacy requirements and privacy policies, and the weighted Euclidean distance is employed to measure the privacy compatibility degree between edge devices and tasks.

(3) Two solutions based on greedy search and KM algorithm are developed to solve the PITAE problem. The experimental results demonstrate that the proposed approaches can significantly improve the privacy compatibility degree of the solution compared with the benchmark approach.

The rest of this paper is structured as follows. Section 2 describes the motivation and framework of the PITAE problem. Section 3 formally specifies the PITAE problem. Section 4 presents two solutions to solve the PITAE problem. The experiments and results are illustrated in Section 5. The related work is reviewed in Section 6. Finally, the conclusion and further works are given in Section 7.

2. Motivation and Framework

In this section, we show an audit example of emergency supply distribution in a disaster relief scenario. In such a scenario, emergency supplies are usually ample in quantity and variety, and the distribution time is urgent. Therefore, it is a very complicated task for traditional manual audit methods to handle. An IoT-based audit application can quickly and automatically execute this process. Such a process captures emergency supply distribution videos stored in CCTV cameras and uses nearby edge devices to analyze the videos to automatically identify some violations, e.g., fake or erroneous emergency supply distribution.

As shown in Figure 1, the workflow of the IoT audit application includes six tasks (t_0-t_5): data collection, object detection, face recognition, supply recognition, violation analysis, and alarm and report. Firstly, task t_0 collects data required for subsequent tasks, e.g., supply distribution video, supply distribution location, and supply application form. Secondly, t_1 uses video data as input to execute object detection and sends the detected face and supply images to t_2 and t_3, respectively. Thirdly, t_2 recognizes the face image to obtain personal identity information (PII), t_3 recognizes the type and quantity of supplies, and t_4 conducts violation analysis based on the recognition results, location, and supply application form. Finally, t_5 issues an alert based on the violation result and generates an audit report. In Figure 1, the rectangular boxes represent tasks, the arrows represent the invocation of the tasks within the application workflow, the vertical solid lines mean that all the previous tasks should be accomplished before the next task is initiated, and the workflow starts from the left and ends at the right.

This example is a typical data-intensive IoT application. The input data of each task may involve the user's private

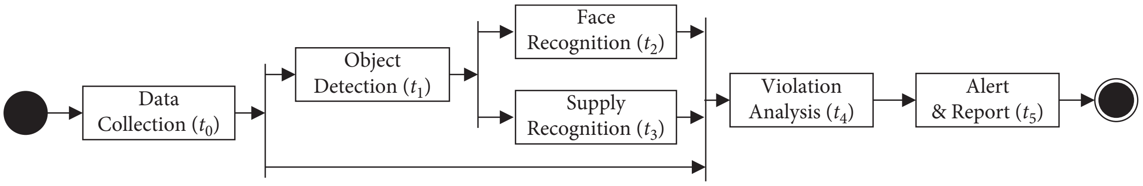


Figure 1: An example IoT audit application.

data, e.g., face, location, application form, etc. In addition, each task needs to be assigned to edge devices with different available resources, e.g., CPU, memory, storage, bandwidth, etc. The resources and private data required for each task are shown in Table 1. There are 10 available edge devices (d_0-d_9) in the demonstration scenario, and the available resources of each device are shown in Table 2.

Before assigning IoT tasks to edge devices, it is necessary to evaluate whether these devices can satisfy the resource requirements of the tasks [29, 30]. As shown in Tables 1 and 2, d_0 only satisfies the resource requirements of t_5, while d_9 can satisfy the resource requirements of all the tasks.

According to the General Data Protection Regulation (GDPR) [31], data consumers can only collect private data for legal purposes. At the same time, the GDPR also requires data consumers not to use the collected data for other purposes, and the retention time of the data and the operations executed on the data must be consistent with those necessary for the stated purpose. To comply with GDPR, private data in Table 1 have a set of privacy requirements, e.g., the sensitivity of the data, the purpose of using the data, the retention time of the data, and the operations executed on the data. Correspondingly, each edge device also provides a set of privacy policies. Therefore, another prerequisite for assigning tasks to edge devices is that the privacy policies of the edge devices should be compatible with the privacy requirements of the tasks. The higher the privacy compatibility degree between the edge device and the task, the more suitable the edge device is to undertake the task.

For example, a privacy requirement of the video data in task t_0 is <video, 0.8, data collection, {read, transfer}, 1>. It means that the sensitivity degree of video is 0.8, and an edge device can only execute read and transfer operations on the video for the purpose of data collection. At the same time, it also requires that the trust degree of the device must be greater than or equal to 0.8 (sensitivity degree), and video cannot be retained more than 1 month. Correspondingly, a privacy policy of edge device d_2 for the video data is <video, 0.6, data collection, {read, write, transfer, profiling}, 12>. It indicates that the trust degree of d_2 is 0.6, d_2 will execute read, write, transfer, and profiling operations on the video for the purpose of data collection, and d_2 will retain the video for at least 12 months. As can be seen from this example, the privacy policy of d_2 is incompatible with the privacy requirement of t_0 in terms of sensitivity degree, operations executed, and retention time.

In summary, the task assignment problem of IoT audit applications is to assign multiple tasks to suitable edge devices, so as to satisfy the resource requirements of the tasks while maximizing the overall privacy compatibility of the assigned edge devices.

Based on the above example, the privacy-aware IoT task assignment framework at the edge of the network is shown in Figure 2. In Figure 2, the developer designs an IoT application based on resource requirements, privacy requirements, tasks, and their dependencies. The tasks of the IoT application need to be deployed to qualified edge devices for execution. Each edge device contains an available resource description file, a privacy policy description file, and is equipped with a task assignment manager responsible for device discovery, qualification evaluation, task assignment, and coordination.

The framework in Figure 2 does not depend on a central coordinator and supports distributed task assignment. Therefore, each participating edge device of IoT applications can generally play the role of coordinator or collaborator. To protect privacy and reduce network transmission, all edge devices participating in the application should be as close as possible to the data source. The IoT application shown in Figure 1 is a typical stream data processing application, and the data collection device (e.g., CCTV camera) is the data production source of the application. Therefore, the application developer selects it as the coordinator of the application, which delivers offloading requests to nearby edge devices. If there are multiple data collection devices (i.e., multiple data sources) in an application, the application developer will select an edge device with large data volume and high privacy protection requirements from these devices as the coordinator.

The coordinator is responsible for discovering a set of qualified edge devices from nearby and forming a collaborative group with these devices as its collaborators, and offloading tasks to these collaborators at the same time. Specifically, once the coordinator receives the deployment request of an IoT application, it will advertise the task processing request to nearby edge devices. The edge devices that are willing to participate in the collaboration accept the request and reply their available resources status and privacy policies to the coordinator. Then, the coordinator evaluates the resource satisfaction and privacy compatibility of each collaborative device. More specifically, the application developer sets a privacy compatibility threshold for tasks. During the privacy evaluation process, if the privacy compatibility between the task and all its candidate devices fails to satisfy the threshold constraints, the coordinator will request the application developer to relax the privacy threshold to ensure that the task has enough devices to perform its function. Finally, the coordinator assigns tasks to the most suitable set of devices according to the evaluation

Table 1: Resource requirements and private data request for tasks.

Tasks	Resource requirements				Private data
	CPU (GHz)	Memory (GB)	Storage (TB)	Bandwidth (Mbps)	
t_0	1.4	4	0.6	18	Video, location, application form
t_1	1.8	6	0.5	18	Video
t_2	1.8	8	0.4	15	Face image
t_3	1.8	8	0.4	15	
t_4	1.6	6	0.6	12	PII, location, application form
t_5	1.2	2	0.2	10	Violation result

Table 2: Available resources provided by edge devices.

Edge devices	Available resources			
	CPU (GHz)	Memory (GB)	Storage (TB)	Bandwidth (Mbps)
d_0	1.2	2	0.2	12
d_1	1.4	4	0.4	18
d_2	1.6	6	0.6	18
d_3	1.8	8	0.8	20
d_4	2.0	10	1.0	22
d_5	1.6	8	0.6	18
d_6	1.8	10	0.8	20
d_7	2.0	12	1.0	22
d_8	2.2	14	1.2	25
d_9	2.5	16	1.5	28

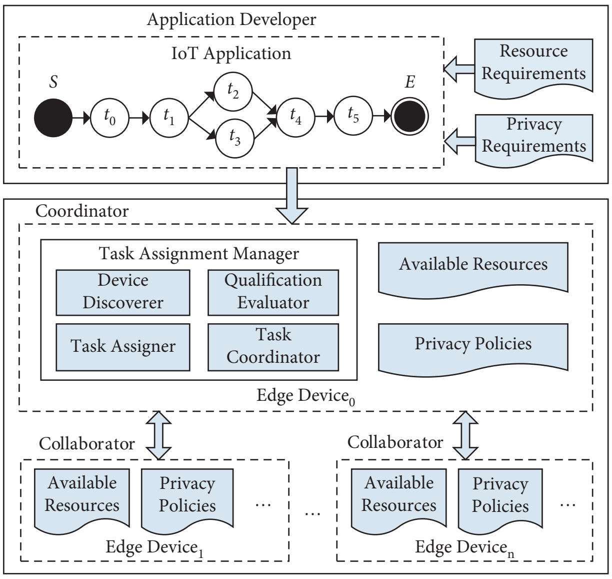


Figure 2: Privacy-aware IoT task assignment framework.

results to maximize the overall privacy compatibility degree of the collaboration group.

After the collaboration group is established, each device starts to execute the assigned tasks. The coordinator is responsible for managing and coordinating the execution of all tasks, and periodically scanning the network to discover new edge devices. Once an edge device leaves the collaboration group, the coordinator will invite a new device to join the collaboration group and assign a task to it. Considering that a task may have multiple candidate new devices, the coordinator first evaluates the resource satisfaction and privacy compatibility between these devices and the task, and then selects the one with the highest privacy compatibility degree for the task from the qualified devices.

3. Problem Description

3.1. Application Model. A typical IoT application is defined by the developer at design time. It specifies the functional and nonfunctional requirements. Formally, it is described by a directed acyclic graph $G = (T, E)$, nodes $T = \{t_0, t_1, .., t_{n\text{-}1}\}$ represent a set of tasks where t_j $(0 \le j < n)$ is the jth task, and edges $E = \{(t_g, t_h) | t_g, t_h \in T\}$ are a set of links between tasks, which represent data and task dependencies. Each task t_j is characterized by a set of inputs $\mathrm{IN}_j = \left\{\mathrm{in}_j^0, \mathrm{in}_j^1, \ldots\right\}$, a set of outputs $\mathrm{OUT}_j = \left\{\mathrm{out}_j^0, \mathrm{out}_j^1, \ldots\right\}$, and a set of resource and privacy requirements.

(1) Resource requirements RR_j: RR_j represents a set of resources required to execute task t_j, such as CPU, memory, storage, and bandwidth. $RR_j = \{rr_j^0, rr_j^1, \ldots, rr_j^o\}$, where rr_j^c $(0 \le j < n, 0 \le c < o)$ is the requirement of task t_j for the cth resource.

(2) Privacy requirements PR_j: Let PD be a set of private data of the user in an IoT application. $PR_j = \{pr_j^0, pr_j^1, \ldots, pr_j^p\}$ specifies a set of privacy requirements for task t_j, where pr_j^k $(0 \le j < n, 0 \le k < p)$ is the kth privacy requirement of t_j, it is defined as a tuple $<pd_j^k, sd_j^k, pu_j^k, OP_j^k re_j^k>$, where $pd_j^k \in PD$ is a private data item of the user, $sd_j^k \in [0, 1]$ is the sensitivity degree of pd_j^k, it specifies the trust degree that an edge device must have when it uses pd_j^k, $sd_j^k = 0$ indicates the lowest sensitivity and 1 the highest, pu_j^k specifies the purpose for which the pd_j^k can be used, OP_j^k specifies a set of operations that can be executed on the pd_j^k, and re_j^k specifies the longest time that the edge device can retain pd_j^k.

3.2. System Model. A PITAE scenario usually consists of multiple heterogeneous edge devices that communicate with each other and collaborate to execute multiple tasks of an IoT application. Let $D = \{d_0, d_1, .., d_{m\text{-}1}\}$represent a set of edge devices, where d_i $(0 \leq i < m)$ is the ith edge device. Each edge device d_i is characterized by a set of available resources and a set of privacy policies.

(1) Available resources AR_i: $AR_i = \{ar_i^0, ar_i^1, \ldots ar_i^{o-1}\}$ represents a set of available resources of d_i, where ar_i^c $(0 \leq i < m,\ 0 \leq c < o)$ is the cth resource of d_i.

(2) Privacy policies $PP_i: PP_i = \{pp_i^0,\ pp_i^1,\ \ldots,\ pp_i^{q-1}\}$ represents a set of privacy policies of d_i, where pp_i^l $(0 \leq i < m,\ 0 \leq l < q)$ is the lth privacy policy of d_i. Each privacy policy pp_i^l is defined as a tuple $< pd_i^l$, td_i^l, pu_i^l, OP_i^l, $re_i^l >$, where $pd_i^l \in PD$ is a private data item for which the policy is defined, $td_i^l \in [0, 1]$ is the trust degree of d_i, where 0 indicates complete no-trust and 1 complete trust, the larger the value of td_i^l, the stronger is the privacy protection provided by the d_i, pu_i^l is the purpose for d_i using pd_i^l, OP_i^l is a set of operations executed by d_i on the pd_i^l, and re_i^l is the time for d_i to retain pd_i^l.

Example 1. Figure 3 demonstrates a privacy-aware IoT task assignment model including 3 tasks and 6 edge devices. That is, $T = \{t_0, t_1, t_2\}$ and $D = \{d_0, d_1, d_2, d_3, d_4, d_5\}$. In Figure 3, circles represent IoT tasks, rectangles represent edge devices, and dashed lines represent potential assignments between tasks and edge devices. The dashed rectangles show the resources requirements and privacy requirements of each task, and the available resources and privacy policies of each device. The prerequisite for whether a task can be assigned to an edge device is that the device can satisfy the resource and privacy requirements of the task.

3.3. Qualification Evaluation Model. To determine whether the edge device d_i is qualified to execute the task t_j, it is necessary to evaluate the resource satisfaction and privacy compatibility between d_i and t_j. The specific evaluation process is as follows:

(1) Resource satisfaction evaluation. Considering that RR_j is a set of minimum resources required to fulfill task t_j, if the edge device d_i is a qualified edge device for t_j, then the available resources AR_i of d_i must satisfy the requirements RR_j. The resource satisfaction evaluation $f_{i,j}^R$ is obtained by

$$f_{i,j}^R = \begin{cases} 1, & \text{if}\ \forall c \in o, ar_i^c \geq rr_j^c \\ 0, & \text{otherwise}, \end{cases} \quad \text{where}\ ar_i^c \in AR_i, rr_j^c \in RR_j \tag{1}$$

(2) Privacy compatibility evaluation. The privacy compatibility degree between the edge device d_i and the task t_j is measured by the average compatibility degree of the privacy requirements of t_j with the corresponding privacy policies in d_i, and it is evaluated by

$$f_{i,j}^P = \frac{\sum_{k=0}^{p-1} f_{i,j}^k}{p}, \tag{2}$$

where $f_{i,j}^k \in [0, 1]$; it represents the privacy compatibility degree between the kth privacy requirement pr_j^k of t_j and the corresponding privacy policy pp_i^l in d_i, and p expresses the number of privacy requirements of t_j, which is an integer greater than or equal to 0.

To evaluate the compatibility degree between pr_j^k and pp_i^l, firstly, it is necessary to ensure that the private data and its usage purpose are consistent, e.g., $pd_j^k = pd_i^l$, $pu_j^k = pu_i^l$; secondly, it is necessary to measure the compatibility degree between pr_j^k and pp_i^l in terms of the sensitivity attribute, operation attribute, and retention time attribute. Accordingly, we express pr_j^k's privacy attributes sd_j^k, OP_j^k, and re_j^k and pp_i^l's privacy attributes td_i^l, OP_i^l, and re_i^l as two three-dimensional vectors. The work in [32] adopts Euclidean distance to evaluate the Security Service-Level Agreement (Security-SLA) between cloud users and cloud service providers. Inspired by this work, we employ the Euclidean distance to measure the compatibility degree between the two privacy attribute vectors. More specifically, considering that the different privacy attributes play different roles in the measurement process, we use the weighted Euclidean distance to reflect the difference in the importance of different attributes. Based on the above analysis, the privacy compatibility degree $f_{i,j}^k$ is calculated by

$$f_{i,j}^k = \begin{cases} \sqrt{w_1 \times \left(f_{i,j}^{k,sd}\right)^2 + w_2 \times \left(f_{i,j}^{k,OP}\right)^2 + w_3 \times \left(f_{i,j}^{k,re}\right)^2}, & \text{if}\ pd_j^k = pd_i^l \wedge pu_j^k = pu_i^l, \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

where w_1, w_2, and w_3 are three weight parameters, $w_1 + w_2 + w_3 = 1$. $f_{i,j}^{k,sd}$, $f_{i,j}^{k,OP}$, and $f_{i,j}^{k,re}$ are the compatibility degrees of the sensitivity attribute, operation attribute, and retention time attribute, respectively. The compatibility degree of sensitivity attribute is obtained by

$$f_{i,j}^{k,sd} = \begin{cases} td_i^l - sd_j^k, & \text{if}\ td_i^l \geq sd_j^k, \\ 0, & \text{otherwise}. \end{cases} \tag{4}$$

The compatibility degree of operation attribute is obtained by

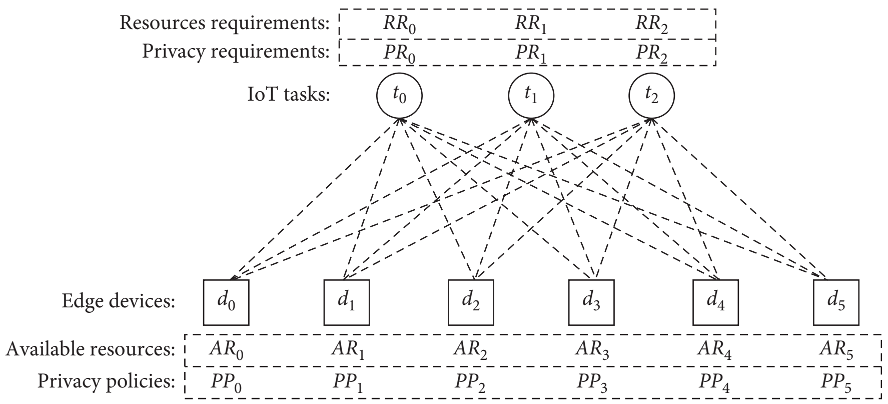


FIGURE 3: An illustration of the privacy-aware IoT task assignment model.

$$f_{i,j}^{k,OP} = \begin{cases} \dfrac{\left(\left|OP_j^k\right| - \left|OP_i^l\right|\right)}{\left|OP_j^k\right|}, & \text{if } OP_i^l \subseteq OP_j^k, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The compatibility degree of retention time attribute is obtained by

$$f_{i,j}^{k,re} = \begin{cases} \dfrac{\left(re_j^k - re_i^l\right)}{re_j^k}, & \text{if } re_i^l \leq re_j^k, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Example 2. Assume that the evaluation results of resource satisfaction and privacy compatibility between tasks and edge devices in Figure 3 are shown in Figures 4(a) and 4(b), respectively. Figure 4(c) shows the potential task assignments that satisfy qualification requirements, where the values on the dotted line represent the degree of privacy compatibility.

3.4. Problem Definition.

3.4. Problem Definition. Despite the ever-increasing resources of edge devices, they are still considered resource-constrained and often unable to execute complex data processing workflow [1]. Hence, the tasks of an IoT application need to be assigned to multiple edge devices for execution. During the task assignment process, if multiple tasks are assigned to an edge device, the available resources of the device may not be able to meet the resource requirements of these tasks. Furthermore, when the device undertakes multiple tasks at the same time, it will collect multiple pieces of private data from different tasks and may infer more privacy information through data mining and machine learning techniques [33]. To meet resource constraints and protect user privacy, in this paper, we assign only one task to each edge device.

Due to the dynamic and distributed nature of edge environments, unpredictable link/device failures and churn of mobile and portable devices often result in IoT applications that are not able to run stably and reliably [34, 35]. To enhance the reliability of the IoT applications, we consider assigning each task to multiple edge devices. that is, the task is backed up to multiple edge devices. When a device that undertakes the task cannot work, the backup device can also ensure the task is executed properly.

In summary, whether a task can be assigned to an edge device is a big issue. If and only if the device satisfies the task's resource requirements and privacy compatibility degree constraint, then the task can be assigned to this device. Given n tasks and m edge devices, the PITAE problem aims to find a solution with maximum privacy compatibility degree by assigning IoT tasks to qualified edge devices. To illustrate the PITAE problem, specific data structures can be formalized as follows:

(1) Lower bound vector of tasks B: It is an n-dimensional vector, where $B[j]$ $(0 \leq j < n)$ expresses how many edge devices must be assigned to task t_j. $B[j] > 1$ means that t_j requires multiple edge devices for execution.

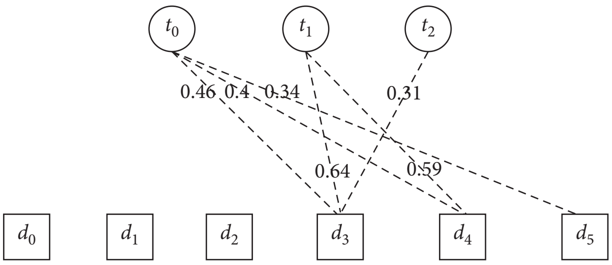
It is worth noting that the application developer does not know the failure and churn rates of edge devices when designing applications. Hence, how to properly set $B[j]$ is nontrivial, which is out of the scope of this paper. We may need to conduct a thorough investigation of this topic in the future. Here, we point out a few initial considerations that require attentions. To enhance the reliability of the IoT applications, each task generally needs to create 2-3 instances: a main task and 1-2 task replicas, and the main task and task replicas are assigned to different edge devices, i.e., $B[j] \leq 3$. We present a $B[j]$ setting scheme as follows: firstly, the application developer preliminarily estimates the average failure and churn rates of edge devices based on experience. Secondly, the application developer determines $B[j]$ by comprehensively considering the average failure and churn rates of the devices, and the criticality of the task t_j. Thirdly, during the task assignment process, if a feasible task assignment solution cannot be found due to some tasks being restricted by B, the application developer will adjust B for these tasks and start a new round of task assignment.

	t_0	t_1	t_2
d_0	0	0	0
d_1	0	0	0
d_2	1	0	0
d_3	1	1	1
d_4	1	1	1
d_5	1	0	0

(a)

	t_0	t_1	t_2
d_0	0.18	0.82	0.26
d_1	0.16	0.34	0.23
d_2	0.22	0.34	0.35
d_3	0.46	0.64	0.31
d_4	0.40	0.59	0.08
d_5	0.34	0.32	0.75

(b)



(c)

Figure 4: (a) Resource evaluation results. (b) Privacy evaluation results. (c) Potential task assignments that satisfy qualification requirements.

(2) Privacy compatibility matrix C: It is an $m \times n$ matrix, where $C[i, j] = f^{P}_{i,j}$ $(0 \leq i < m,\ 0 \leq j < n)$ denotes the privacy compatibility degree between the edge devices d_i and the task t_j.

(3) Evaluation matrix E: It is an $m \times n$ matrix, where $E[i, j](0 \leq i < m,\ 0 \leq j < n)$ expresses whether the edge device d_i satisfies the resource and privacy compatibility threshold constraints of task t_j, and $E[i, j] = 1$ means yes and 0 no. $E[i, j]$ is obtained by

$$E[i, j] = \begin{cases} 1, & \text{if } f^{P}_{i,j} \geq th \wedge f^{R}_{i,j} = 1, \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

where $th \in [0, 1]$ is the privacy compatibility threshold, which specifies the minimum privacy compatibility degree that the edge devices must have when executing tasks.

(4) Assignment matrix A: It is an $m \times n$ matrix, where $A[i, j](0 \leq i < m, 0 \leq j < n) \in \{0, 1\}$ expresses whether t_j is assigned to the edge device d_i ($A[i, j] = 1$) or not ($A[i, j] = 0$).

Given B, C, and E, the PITAE problem is to find a matrix A to Max:

$$\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} C[i, j] \times A[i, j]. \tag{8}$$

subject to

$$A[i, j] \in \{0, 1\} (0 \leq i < m, 0 \leq j < \text{n}), \tag{9}$$

$$\sum_{i=0}^{m-1} A[i, j] = B[j] (0 \leq j < n), \tag{10}$$

$$\sum_{j=0}^{n-1} A[i, j] \leq 1 (0 \leq i < m), \tag{11}$$

$$E[i, j] \times A[i, j] > 0 (0 \leq i < m, 0 \leq j < \text{n}), \tag{12}$$

where Constraint (9) specifies that the decision variables are binary; Constraint (10) guarantees that each task is assigned $B[j]$ edge devices; Constraint (11)ensures that each edge device can only be assigned to one task; and Constraint (12) ensures that each assigned edge device satisfies the resource and privacy compatibility threshold constraints.

Example 3. In an IoT audit application, the resource requirements of tasks and the available resources provided by edge devices are shown in Tables 1–2. Assume that the privacy compatibility threshold th is specified as 0.3, the lower bound vector of tasks $B = [1, 1, 2, 2, 1, 1]$, and the privacy compatibility matrix is shown in Figure 5(a). The evaluation matrix is obtained by Equation (7), as shown in Figure 5(b). Based on B and Figure 5(a) and 5(b), the assignment solution with the maximal privacy compatibility degree (5.03) should be $\{d_6, d_3, \{d_7, d_9\}, \{d_4, d_8\}, d_2, d_1\}$, and the assignment matrix is demonstrated in Figure 5(c).

4. Solutions to the PITAE Problem

The PITAE problem is a typical one-to-many task assignment problem. If the exhaustive search method is used to solve this problem, the solution space can be up to $O\ (m^n)$ [36]. Therefore, we first develop a task assignment solution based on the greedy search to solve this problem. Then, to improve the effectiveness of task assignment, we propose a task assignment solution based on the KM algorithm to find the optimal solution to the PITAE problem.

4.1. Greedy Search-Based Task Assignment (GSTA) Solution. The GSTA solution selects $B[j]$ the most qualified edge devices for each task in the task set Taccording to the privacy compatibility matrix C and the evaluation matrix E. Specifically, for each t_j belonging to $T$and d_i belonging to D, it first evaluates whether d_i satisfies the resource requirements and privacy compatibility threshold constraints of task t_j, e.g., $E[i, j] = 1$. Then, it determines whether d_i has been assigned a task, e.g., $S[i] = 1$. If yes, it skips d_i and examines the next edge device; otherwise, it adds the privacy compatible degree $C[i, j]$ to the candidate edge device vector V of t_j. Subsequently, it reversely sorts all candidate edge devices in V according to their privacy compatibility degrees and selects top $B[j]$ candidate edge devices for t_j from sorted candidate edge device vector SV. Finally, it sets the assignment $A[i, j]$ corresponding to the edge device d_i and task

$$\begin{bmatrix} 0.18 & 0.82 & 0.26 & 1.00 & 0.45 & 0.05 \\ 0.16 & 0.34 & 0.23 & 1.00 & 0.55 & 0.58 \\ 0.22 & 0.34 & 0.35 & 1.00 & 0.47 & 0.16 \\ 0.46 & 0.64 & 0.31 & 1.00 & 0.37 & 0.58 \\ 0.40 & 0.59 & 0.08 & 1.00 & 0.48 & 0.73 \\ 0.34 & 0.32 & 0.75 & 1.00 & 0.20 & 0.39 \\ 0.62 & 0.35 & 0.44 & 1.00 & 0.39 & 0.62 \\ 0.40 & 0.52 & 0.39 & 1.00 & 0.60 & 0.24 \\ 0.43 & 0.40 & 0.27 & 1.00 & 0.50 & 0.45 \\ 0.39 & 0.41 & 0.33 & 1.00 & 0.57 & 0.38 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

(a) (b) (c)

Figure 5: Matrixes. (a) The privacy compatibility matrix. (b) The evaluation matrix. (c) The assignment matrix.

t_j to 1, and updates the edge device selection vector S. The details of GSTA are described in Algorithm 1.

The time complexity of Algorithm 1 is $O(n \times m + n \times m \times \log_2 m)$, where $O(m \times \log_2 m)$ is the time complexity of a sorting operation.

4.2. KM Algorithm-Based Task Assignment (KMTA) Solution. In a data-intensive IoT application, its workflow is usually composed of tens of tasks, rarely hundreds or thousands [1]. The IoT audit application is a typical data-intensive IoT application, and we estimate its number of tasks to be on the order of tens of magnitudes. Moreover, to enhance the reliability of the application, each task generally needs to be assigned to 2-3 edge devices, e.g., $B[j] \leq 3$. Therefore, the total number of edge devices required for an IoT audit application should be around tens to two hundred. On the other hand, with the widespread application of the IoT technology, there are often hundreds of IoT devices connected to the edge network near the data source. Based on the above considerations, in the PITAE scenario, we believe that the number of edge devices can meet the needs of IoT tasks, e.g., $m > n$, and each task requires $B[j]$ edge devices to execute it, but each edge device can only be assigned to one task.

The well-known KM algorithm can quickly solve standard task assignment problems, i.e., one-to-one task assignment problems, and the time complexity is $O(m^3)$ [27, 28]. In addition, the KM algorithm always finds the solution with the smallest sum [29]. However, the PITAE problem needs to find a solution with the maximum privacy compatibility degree. Furthermore, the KM algorithm can always find a result for the PITAE problem, but the result may not be a feasible solution. For example, when the edge device d_i cannot satisfy the resource requirements or the privacy compatibility threshold constraints of t_j, i.e., $E[i, j] = 0$, the KM algorithm may produce incorrect task assignments, leading to an infeasible solution.

To deal with the limitations of the KM algorithm, the KMTA solution improves the KM algorithm to solve the PITAE problem by adding virtual tasks and adjusting the privacy compatibility degrees between tasks and edge devices. Concretely, first of all, for each d_i belonging to D and t_j belonging to T, it evaluates whether d_i satisfies the resource requirements and privacy compatibility threshold constraints of t_j, and adjusts the privacy compatibility value $C[i,j]$ according to the evaluation result. More specially, if d_i passes the evaluation, e.g., $E[i, j] = 1$, it adjusts $C[i, j]$ to mpc-$C[i, j]$; otherwise, it adjusts $C[i, j]$ to $\sum_{j=0}^{n-1} B[j]$. The adjustment operation ensures that KMTA can find the solution with the maximum privacy compatibility degree, because mpc is the maximum privacy compatibility value in C, $C[i, j] \in [0, 1]$, and the privacy compatibility degree of a solution never exceeds $\sum_{j=0}^{n-1} B[j]$. Secondly, it extends matrix C into an m rows and $\sum_{j=0}^{n-1} B[j]$ columns matrix C^*, where for each column j in C, there are $B[j]$ corresponding copy columns in C^*. If the number of rows of C^* is greater than the number of columns, i.e., $m > \sum_{j=0}^{n-1} B[j]$, it adds $m - \sum_{j=0}^{n-1} B[j]$ virtual columns to C^* and sets their privacy compatibility value to 0. Thirdly, it calls the KM algorithm to obtain a temporary matrix H and forms the assignment matrix A according to H. Finally, it checks whether A is a feasible assignment solution. If each assignment in A is correct and each task is assigned $B[j]$ edge devices, it returns success; otherwise, it returns failure. The details of KMTA are shown in Algorithm 2.

The time complexity of Algorithm 2 is determined by the following: (1) the time complexity of adjusting the C matrix is $O\ (m \times n)$; (2) the time complexity of extending the C matrix is $O\ (m \times n \times B[j]) + O\ (m \times (m - \sum_{j=0}^{n-1} B[j]))$; (3) the time complexity of calling the KM algorithm and forming the assignment solution is $O\ (m^3) + O\ (m \times n)$; and (4) the time complexity of judging the feasibility of the solution is $O\ (m \times n) + O\ (n)$. Thus, the overall complexity of Algorithm 2 is $O(m^3) + O\ (m \times n \times B[j]) + O(m^2) + O(m \times n) + O\ (m - \sum_{j=0}^{n-1} B[j]) + O\ (n)$. In the presented scenarios, $B[j]$ is a constant (typically less than 10), and $m > n$. Consequently, the time complexity of Algorithm 2 can be simplified as $O\ (m^3)$.

```
    Input:
    $T$: the tasks set; $D$: the edge devices set; $B$: the lower bound vector;
    $C$: the compatibility matrix; $E$: the evaluation matrix; $S$: the edge device selection vector.
    Output:
    $A$: the task assignment matrix.
for each task $t_j$ in $T$ do
    for each edge device $d_i$ in $D$ do
        if $E[i, j] = 1$ then
            if $S[i] = 1$ then;
                skip it and examine the next edge device;
            else
                $V \leftarrow C[i, j]$;
            end if
        end if
    end for
    $SV \leftarrow$ sorting $V$ based on privacy compatibility degree;
    Select Top-$B[j]$ edge devices from $SV$;
    Update $A[i, j]$ and $S[i]$;
end for
return $A$;
```

Algorithm 1: Greedy search-based task assignment.

```
Input:
$T$: the tasks set; $D$: the edge devices set; $B$: the lower bound vector;
$C$: the privacy compatibility matrix; $E$: the evaluation matrix.
Output:
    Success: $A$; failure: no feasible $A$ is obtained.
for each edge device $d_i$ in $D$ do
    for each task $t_j$ in $T$ do
        if $E[i, j] = 1$ then
            $C[i, j] \leftarrow mpc - C[i, j]$;
        else
            $C[i, j] \leftarrow \sum_{j=0}^{n-1} B[j]$;
        end if
    end for
end for
for each edge device $d_i$ in $D$ do
    $cindex \leftarrow 0$;
    for each task $t_j$ in $T$ do
        while $B[j] > 0$ do
            $C^*[i, cindex{+}{+}] \leftarrow C[i, j]$;
            $B[j] \leftarrow B[j] - 1$;
        end while
    end for
end for
if $m > \sum_{j=0}^{n-1} B[j]$ then
    Add $m - \sum_{j=0}^{n-1} B[j]$ virtual columns to $C^*$, and set their corresponding element values to 0;
end if
$H \leftarrow \mathrm{KM}(C^*)$;
Form the assignment matrix $A$ based on $H$;
if there is any incorrect assignment in $A$ then
    return Failure
end if
if for all columns of matrix $A$ satisfy $\sum_{i=0}^{m} A[i, j] = B[j]$ then
    return Success
else
    return Failure
end if
```

Algorithm 2: KM algorithm-based task assignment.

5. Experiments

In this section, we conducted four sets of simulation experiments to evaluate the effectiveness and efficiency of KMTA and GSTA. As far as we know, there is no other research directly related to our study. Hence, we implement a "Random (RNDM)" approach as a benchmark to compare with KMTA and GSTA. Given a set of tasks and a set of edge devices, RNDM randomly assigns each task to $B[j]$ edge devices that satisfy the resource requirements and privacy compatibility threshold constraints. All the experiments are performed on a Windows platform equipped with Intel Core i7-4790 @ 3.60 GHz and 8 GB RAM.

5.1. Experimental Setting. To comprehensively evaluate GSTA and KMTA, we have simulated various PITAE scenarios by changing the following parameters: (1) the number of edge devices (m); (2) the number of tasks (n); and (3) the privacy compatibility threshold (th). Specifically, in set #1, m changes from 30 to 300 with a step of 30, $n = m/3$, and th is set to 0.1. In set #2, m changes from 50 to 500 with a step of 50, $n = m/5$, and th is set to 0.1. In set #3, m and n are fixed at 150 and 50, respectively, and th changes from 0.1 to 0.5 with a step of 0.1. In set # 1.4, m is fixed at 250, and the other parameters are set as in set # 1.3. Each experiment is repeated 100 times, and the results are averaged. The detailed experimental settings are shown in Table 3.

Table 3: Experimental setting.

	m	n	th
Set #1	30, 60, . . ., 300	10, 20, . . ., 100	0.1
Set #2	50, 100, . . ., 500		
Set #3	150	50	0.1, 0.2, 0.3, 0.4, 0.5
Set #4	250		

In sets #1–4, $B[j]$ is randomly assigned from 1 to 3, and the resource requirements of each task and the available resources provided by each edge device are randomly generated following the uniform distribution. The details are shown in Table 4.

In sets #1–4, each task is randomly assigned 0-10 pieces of private data, and the privacy requirements and the privacy policies are randomly generated for private data. Specifically, for a privacy requirement $pr^k_j = < pd^k_j, sd^k_j, pu^k_j, OP^k_j, re^k_j >$, sd^k_j is assigned randomly with a value in [0.00, 1.00], pu^k_j is assigned randomly from 10 different purposes, OP^l_j is randomly generated from an operation set containing 5 different operations, and re^l_j is assigned randomly from 1 to 12 months. For a privacy policy $pp^l_i = < pd^l_i, td^l_i, pu^l_i, OP^l_i, re^l_i >$, the td^l_i, pu^l_i, OP^l_i, and re^l_i are the same as the setting of corresponding privacy attributes in pr^k_j.

5.2. Effectiveness Evaluation. Through comparison with RNDM, Figures 6 and 7 show the effectiveness of KMTA and GSTA in experiment sets #1-4 and the influence of three parameters, i.e., n, m, and th. On the whole, KMTA can find the optimal solution for the PITAE problem, and with the changes of n, m, and th, KMTA is significantly better than GSTA and RNDM in terms of privacy compatibility degree. Compared to KMTA, GSTA's privacy compatibility degree is lower than that of KMTA, especially in the case of stricter th constraints, but it is still significantly higher than RNDM in all cases.

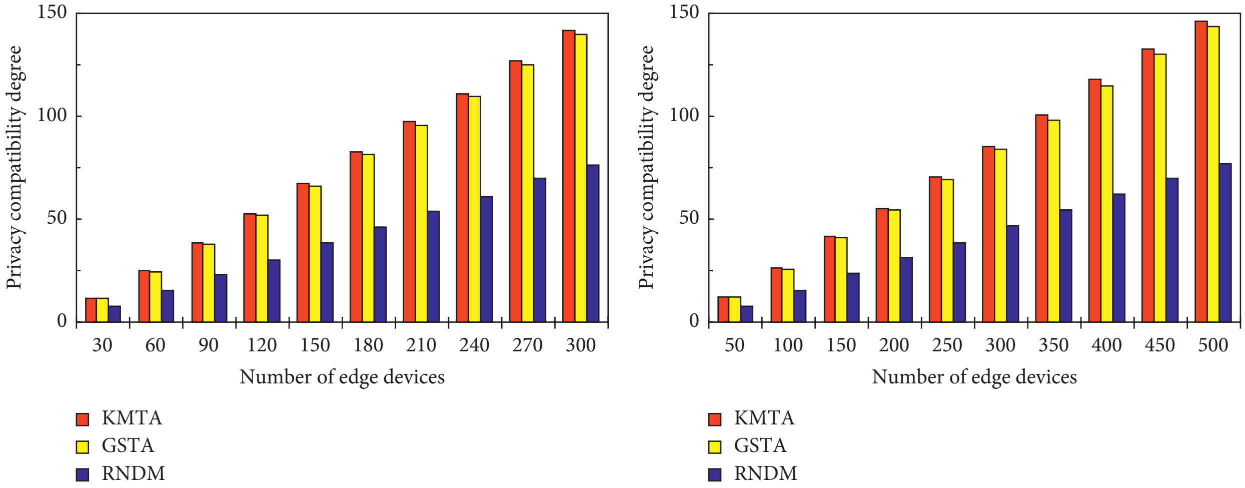
Figure 6 illustrates the effect of increasing m on privacy compatibility degree. As shown in Figure 6(a), as m increases, the privacy compatibility degrees of all the approaches increase rapidly. In all cases, KMTA shows the highest privacy compatibility degree, RNDM shows the lowest privacy compatibility degree, and GSTA's privacy compatibility degree is slightly lower than that of KMTA. The reason is that KMTA always assigns $B[j]$ qualified edge devices to each task globally to obtain the highest privacy compatible solution. Hence, it can find the optimal solution to the PITAE problem. GSTA always assigns $B[j]$ qualified edge devices with the highest privacy compatibility for each task locally, resulting in the privacy compatibility degree of the solution it finds slightly lower than that of KMTA. However, RNDM always randomly assigns each task to $B[j]$ qualified edge devices. Consequently, the solution it finds has the lowest privacy compatibility degree. For example, in Figure 6(a), the average privacy compatibility degrees of KMTA, GSTA, and RNDM are 75.59, 74.33, and 42.28, respectively.

In Figure 6(b), as m/n increases from 3 to 5, the average range of candidate edge devices for each task also enlarges. As a result, the privacy compatibility degrees of all the approaches have improved to varying degrees, and KMTA is still higher than GSTA and RNDM. For example, comparing Figure 6(b) with Figure 6(a), the average privacy compatibility degrees of KMTA, GSTA, and RNDM increase by 4.11%, 3.87%, and 1.31%, respectively.

Figure 7 demonstrates the effect of th on the privacy compatibility degree after fixing m and n. It can be seen from Figure 7(a) that when th increases from 0.1 to 0.5, the privacy compatibility degrees of KMTA and RNDM remain basically unchanged, but the privacy compatibility degree of GSTA shows a clear downward trend. It is because as th increases, the number of qualified edge devices for each task decreases. Due to that GSTA always selects edge devices locally for each task, it is most affected by th. For example, in Figure 7(a), the privacy compatibility degrees of KMTA and RNDM are kept at about 45 and 18, respectively, in all cases. However, GSTA's privacy compatibility degree is reduced from 44.89 to 17.56. When m/n increases from 3 to 5, and we compare Figure 7(b) with Figure 7(a), the privacy compatibility degrees of all the approaches show different degrees of improvement, but the privacy compatibility degree of GSTA still decreases with the increases of th. For example, in Figure 7(b), the privacy compatibility degrees of KMTA and RNDM maintains at about 47 and 19, respectively, in all cases. However, GSTA's privacy compatibility degree is reduced from 46.91 to 23.43.

Table 4: Resource requirements and available resources settings.

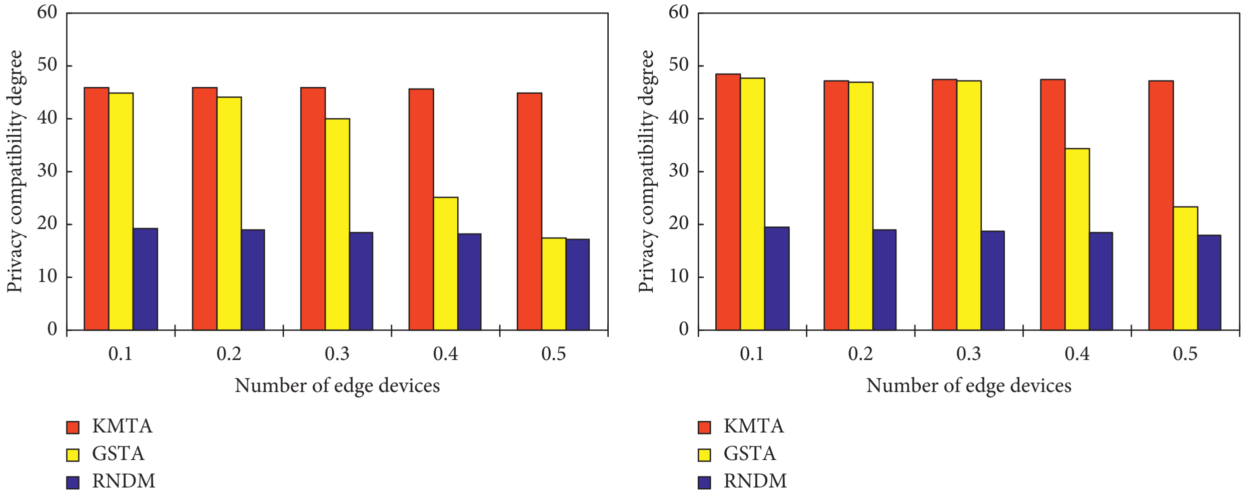
	CPU (GHz)	Memory (GB)	Storage (TB)	Bandwidth (Mbps)
Resources requirements	[1, 2]	[2, 8]	[0.2, 1]	[10, 20]
Available resources	[1, 3]	[2, 16]	[0.5, 2]	[10, 30]



(a)

(b)

Figure 6: Effectiveness vs. number of edge devices. (a) Set #1. (b) Set #2.



(a)

(b)

Figure 7: Effectiveness vs. number of edge devices. (a) Set #3. (b) Set #4.

5.3. Efficiency Evaluation. Figure 8 shows the times taken by KMTA, GSTA, and RNDM to find a solution. Since the solving time of the PITAE problem is mainly affected by n and m, we only compare the average execution time of all the approaches in experiment sets #1-2. In general, because KMTA is an optimal approach to solve the PITAE problem, it takes more execution time than GSTA and RNDM. Especially, when m and n are relatively large, this trend becomes more obvious.

As shown in Figure 8(a), when m is relatively small, e.g., $m < 120$, all the approaches consume basically the same time and increase slowly. However, when $m \geq 120$, KMTA consumes more time than GSTA and RNDM, and the consumed time by KMTA increases rapidly. For example, when m rises from 120 to 300, the execution time of KMTA increases from 10.91 ms to 320.77 ms, while the execution time of GSTA and RNDM is less than KMTA and remains below 15 ms. The results observed from Figure 8(b) show the influence of increasing m/n on time consumption. If we compare Figure 8(b) with Figure 8(a), we notice that the consumed time of all the approaches increases to different degrees. In addition, similar to Figure 8(a), in Figure 8(b), when m is

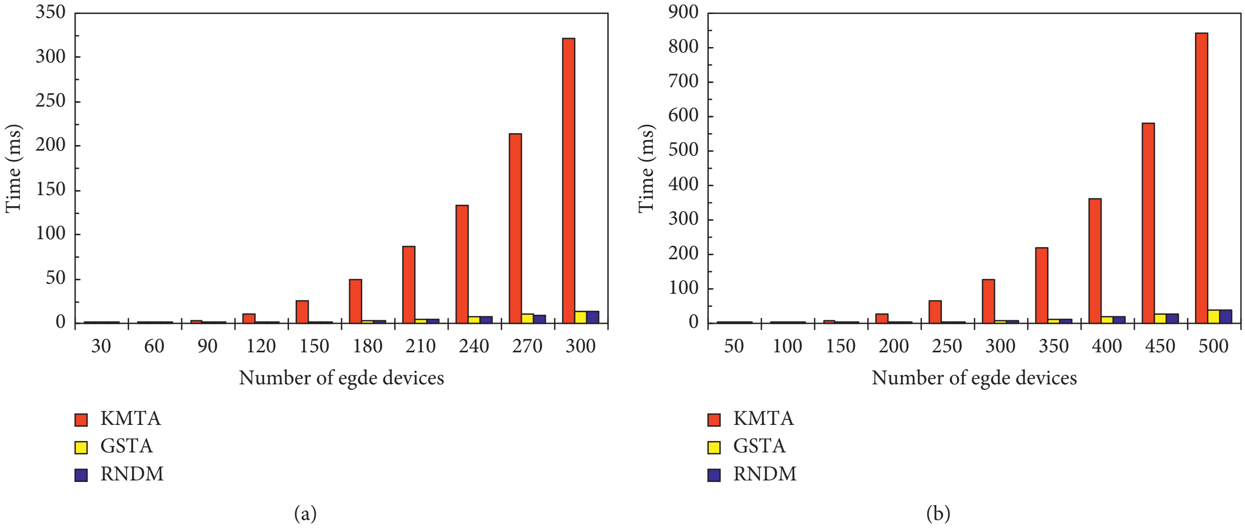


Figure 8: Average time consumption vs. number of edge devices. (a) Set #1. (b) Set #2.

relatively small, all the approaches consume basically the same time and increase slowly, while when $m \geq 200$, KMTA consumes more time than GSTA and RNDM, and the consumed time by KMTA increases rapidly. For example, in Figure 8(b), when m rises from 200 to 500, the time taken by KMTA increases from 29.29 ms to 840.79 ms, while GSTA and RNDM take less time than KMTA and keep the consumed time below 40 ms.

5.4. Discussion. From the above experimental results, we can make the following conclusions.

(1) In terms of effectiveness, KMTA and GSTA have significant advantages over RNDM. In addition, in all cases, KMTA can find a solution with a higher privacy compatibility degree than GSTA, especially in cases with stricter privacy constraints; e.g., th is relatively large, and the advantages of KMTA are more obvious.

(2) In terms of performance, the execution time of GSTA and RNDM is basically the same in all cases. In the case where m and n are relatively small, the execution time of KMTA is basically the same as that of GSTA and RNDM. However, in the case where m and n are relatively large, the execution time of KMTA is much longer than that of GSTA and RNDM.

(3) Although expanding m/n can improve the privacy compatibility degrees of all the approaches, it also brings more time consumption.

(4) In cases where m and n are relatively small or th is relatively large, KMTA outperforms GSTA and RNDM significantly. However, when m and n are relatively large, the overall performance of GSTA is better than that of KMTA and RNDM. In short, KMTA and GSTA can beat RNDM in different cases. Therefore, we can choose KMTA or GSTA to assign tasks according to different m, n, and th scenarios.

6. Related Work

With the emergence of a large number of edge devices with sensing, actuation, and computing capabilities in the urban environment, it has become more complicated to assign IoT tasks to edge devices for execution [8, 12]. Many research efforts have been focusing on task assignment based on vertical offloading technology and horizontal offloading technology. The former relies on a centralized coordinator to place simple task processing on local edge devices, while offloading complex data analysis tasks to fog/cloud nodes. The latter offloads tasks to multiple edge devices that are as close as possible to the data source, and these devices execute tasks in a distributed manner.

To serve IoT applications at the edge, Farhadi et al. [22] proposed a joint optimization method for service placement and request scheduling, and developed polynomial time algorithms to solve the placement and scheduling problems. Aiming at the task allocation problem in collaborative edge and cloud environment, Long et al. [21] proposed a non-cooperative game model between multiple agents and solved the task allocation problem with QoS constraints through a series of algorithms. Considering the latency and bandwidth requirements of IoT applications, Antonio et al. [20] proposed a QoS-aware application deployment method in fog computing. The proposed method models the deployment requirements of IoT applications, describes the available resources and quality of fog nodes, and develops optimization algorithms for the application deployment problem. Cheng et al. [37] proposed a task assignment method in a data sharing mobile edge computing system and designed three algorithms to deal with the holistic and divisible task assignment problem.

The above work uses vertical offloading technology to assign tasks for IoT applications. Recently, some new work has also emerged in the aspect of horizontal task offloading. The work in [1] clusters heterogeneous edge devices to process data-intensive IoT applications. The proposed method first decomposes an IoT application into a set of simple tasks, then automatically discovers qualified edge devices, and finally assigns tasks to appropriate edge devices. Similarly, Avasalcai et al. [2] proposed a decentralized resource management framework for deploying delay-sensitive IoT applications at the edge of the network and found deployment solutions that meet the requirements through satisfiability modulo theory (SMT) technology.

The above work mainly focuses on the task allocation problem of resource and QoS constraints, and rarely considers user privacy requirements. With the widespread adoption of IoT applications, users are increasingly concerned about the privacy of their personal data. Some research contributions focus on the privacy-aware task assignment for IoT applications.

Aiming at the privacy protection problem in socially aware edge computing, Zhang et al. [23] proposed a privacy-aware task allocation method. The proposed method uses generalization techniques to reduce the accuracy of private data and develops a game theory model to optimize the QoS of the application while ensuring that the user's privacy requirements are satisfied. To protect user privacy in IoT data, Mian et al. [24] proposed a privacy-aware task offloading method in fog computing. The method first divides the IoT tasks into different small fragments according to the security requirements of the data, then these task fragments are offloaded to multiple fog nodes that meet security requirements, and finally a dynamic programming algorithm is used to obtain the task offloading solution that meets the security and delay requirements. Considering the privacy leakage of sensing data in mobile crowd sensing systems, Dai et al. [25] proposed a privacy preservation task assignment scheme and designed a user location privacy protection algorithm based on the differential privacy method. To avoid the privacy disclosure of the datasets due to data acquisition by different operators, Xu et al. [26] took the privacy conflict of different datasets as the optimization goal, formulated the application deployment problem in cyber-physical cloud systems as a multi-objective optimization problem, and used an improved differential evolution technology to solve it.

Although the above work has advantages, the privacy-aware task assignment for IoT applications is still an open issue. The above work employs various privacy technologies to control access to private data, but does not consider how the data will be used after being accessed, such as the purpose of data use, the retention time of the data, and the operations executed on the data. Our approach can fully support these requirements and can also measure the compatibility degree between privacy requirements and privacy policies.

Group Role Assignment (GRA) [29, 30, 36, 38, 39] has been proposed for modeling general assignment problems by solving different engineering problems. The solution to the GRA provides inspiration to this research. The creation of a qualification matrix of GRA is a prerequisite way to model various assignment problems in edge computing.

7. Conclusion

The edge computing paradigm has a great potential to support a wide variety of IoT applications. In this paper, we propose a privacy-aware task assignment approach for IoT applications, which assigns tasks to edge devices close to the data source in a distributed manner, thereby reducing latency and effectively protecting user privacy. Firstly, we model the resource and privacy requirements of the tasks and assess whether the edge devices satisfy the resource and privacy constraints. Secondly, we formalize the PITAE problem as an integer programming optimization problem and propose two task assignment solutions to solve the PITAE problem. Finally, we compare the proposed approaches with the baseline approach. Experimental results show that (1) when m and n are relatively small or th is relatively large, KMTA outperforms GSTA and RNDM significantly; and (2) when m and n are relatively large, the overall performance of GSTA is better than that of KMTA and RNDM. In short, KMTA and GSTA can beat RNDM in different cases.

For future work, we intend to extend our work with QoS constraints, such as response time (communication latency between edge devices and processing latency on edge devices) and energy consumption (transmission energy between edge devices and processing energy on edge devices), in order to provide a more effective task assignment solution that can meet diverse requirements. In addition, considering the privacy protection requirements of edge devices for various resource information and willingness to undertake tasks, we also plan to integrate these requirements into our current privacy model, so as to achieve privacy protection for users and edge devices at the same time.

Another direction is to specify and solve problems related to privacy protection in edge computing along with the development of GRA with constraint (GRA+) model [36, 38, 39], which provides different ways in modeling various constraints, such as time, space, and coupling between agents (resources) and roles (tasks).

Data Availability

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

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was funded by the Natural Sciences and Engineering Research Council of Canada (Grant no. RGPIN-2018-04818), National Natural Science Foundation of China (Grant no. 61772270), and Jiangsu Province Planning Subject for the 13th Five-Year Plan of Education Sciences (Grant no. 2016-GH0303-00022).

References

[1] R. Dautov and S. Distefano, “Automating IoT data-intensive application allocation in clustered edge computing,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 1, pp. 55–69, 2021.

[2] C. Avasalcai, C. Tsigkanos, and S. Dustdar, “Resource management for latency-sensitive IoT applications with satisfiability,” *IEEE Transactions on Services Computing*, p. 1, 2021.

[3] D. Liu, Q. Jiang, H. Zhu, and B. Huang, “Distributing UAVs as wireless repeaters in disaster relief via group role assignment,” *International Journal of Cooperative Information Systems*, vol. 29, no. 01n02, Article ID 2040002, 2020.

[4] Gartner, “Gartner Forecasts Worldwide IoT-Enabled Software in 2019-2025,” 2021, https://www.gartner.com/en/documents/4009207-forecast-iot-enabled-software-worldwide-2019-2025.

[5] E. Ahmed, I. Yaqoob, A. Gani, M. Imran, and M. Guizani, “Internet-of-Things-Based smart environments: state of the art, taxonomy, and open research challenges,” *IEEE Wireless Communications*, vol. 23, no. 5, pp. 10–16, 2016.

[6] S. Eugene, T. Thanassis, and H. Wendy, “Analytics for the Internet of Things: a survey,” *ACM Computing Surveys*, vol. 51, no. 4, pp. 74:1–74:36, 2018.

[7] R. Casadei, G. Fortino, D. Pianini, W. Russo, C. Savaglio, and M. Viroli, “Modelling and simulation of opportunistic IoT services with aggregate computing,” *Future Generation Computer Systems*, vol. 91, pp. 252–262, 2019.

[8] G. Fortino, C. Savaglio, G. Spezzano, and M. Zhou, “Internet of Things as system of systems: a review of methodologies, frameworks, platforms, and tools,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 223–236, 2021.

[9] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: vision and challenges,” *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[10] M. Satyanarayanan, “The emergence of edge computing,” *Computer*, vol. 50, no. 1, pp. 30–39, 2017.

[11] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: the communication perspective,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[12] M. Dias de Assunção, A. da Silva Veith, and R. Buyya, “Distributed data stream processing and edge computing: a survey on resource elasticity and future directions,” *Journal of Network and Computer Applications*, vol. 103, pp. 1–17, 2018.

[13] R. Dautov and S. Distefano, “Stream processing on clustered edge devices,” *IEEE Transactions on Cloud Computing*, vol. 10, no. 2, pp. 885–898, 2020.

[14] M. Sun, Z. Zhou, X. Xue, W. Zhang, and W. Gaaloul, “Adaptive configuration of service-based smart sensors in edge networks,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 4, pp. 2674–2683, 2022.

[15] N. Fernando, S. W. Loke, and W. Rahayu, “Computing with nearby mobile devices: a work sharing algorithm for mobile edge-clouds,” *IEEE Transactions on Cloud Computing*, vol. 7, no. 2, pp. 329–343, 2019.

[16] E. Ahmed, A. Ahmed, I. Yaqoob, and J. A. M. M. Shuja, “Bringing computation closer toward the user network: is edge computing the solution?” *IEEE Communications Magazine*, vol. 55, no. 11, pp. 138–144, 2017.

[17] C. Tsigkanos, C. Avasalcai, and S. Dustdar, “Architectural considerations for privacy on the edge,” *IEEE Internet Computing*, vol. 23, no. 4, pp. 76–83, 2019.

[18] A. Alwarafy, K. A. Al-Thelaya, M. Abdallah, J. Schneider, and M. Hamdi, “A survey on security and privacy issues in edge-computing-assisted Internet of Things,” *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4004–4022, 2021.

[19] A. Atheer, B. Masoud, R. Omer, and P. Charith, “Privacy laws and privacy by design schemes for the Internet of Things: a developer’s perspective,” *ACM Computing Surveys*, vol. 54, no. 5, pp. 102:1–102:38, 2021.

[20] A. Brogi and S. Forti, “QoS-aware deployment of IoT applications through the fog,” *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1185–1192, 2017.

[21] S. Long, W. Long, Z. Li, K. Li, Y. Xia, and Z. Tang, “A game-based approach for cost-aware task assignment with QoS constraint in collaborative edge and cloud environments,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 7, pp. 1629–1640, 2021.

[22] V. Farhadi, F. Mehmeti, T. He, and T. F. L. H. S. K. S. K. Porta, “Service placement and request scheduling for data-intensive applications in edge clouds,” *IEEE/ACM Transactions on Networking*, vol. 29, no. 2, pp. 779–792, 2021.

[23] D. Zhang, Y. Ma, X. Sharon Hu, and D. Wang, “Toward privacy-aware task allocation in social sensing-based edge computing systems,” *IEEE Internet of Things Journal*, vol. 7, no. 12, Article ID 11384, 2020.

[24] M. R. Mian, T. Byungchul, L. Peng, and G. Mohsen, “Privacy-aware collaborative task offloading in fog computing,” *IEEE Trans. Comput. Soc. Syst.*vol. 9, 2022.

[25] M. Dai, J. Li, Z. Su, W. Chen, Q. Xu, and S. Fu, “A privacy preservation based scheme for task assignment in Internet of Things,” *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2323–2335, 2020.

[26] X. Xu, R. Mo, X. Yin et al., “PDM: privacy-aware deployment of machine-learning applications for industrial cyber-physical cloud systems,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5819–5828, 2021.

[27] H. W. Kuhn, “The Hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[28] J. Munkres, “Algorithms for the assignment and transportation problems,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[29] H. Zhu, M. Zhou, and R. Alkins, “Group role assignment via a kuhn-munkres algorithm-based solution,” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 42, no. 3, pp. 739–750, 2012.

[30] H. Zhu, *E-CARGO and Role-Based Collaboration: Modeling and Solving Problems in the Complex World*, Wiley-IEEE Press, Hoboken, NJ, USA, 2021.

[31] E. Union, *GeneralData Protection Regulation*, European Union, Maastricht, Netherlands, 2018.

[32] T. Halabi and M. Bellaiche, “A broker-based framework for standardization and management of cloud security-SLAs,” *Computers & Security*, vol. 75, pp. 59–71, 2018.

[33] L. Liu, H. Zhu, S. Chen, and Z. Huang, “Privacy regulation aware service selection for multi-provision cloud service composition,” *Future Generation Computer Systems*, vol. 126, pp. 263–278, 2022.

[34] D. T. Nguyen, H. T. Nguyen, N. Trieu, and V. K. Bhargava, “Two-stage robust edge service placement and sizing under demand uncertainty,” *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 1560–1574, 2022.

[35] J. Xu, B. Palanisamy, and Q. Wang, “Resilient stream processing in edge computing,” in *Proceedings of the 21st IEEE/*

ACM International Symposium on Cluster, Cloud and Internet Computing, (CCGrid), Melbourne, Australia, May 2021.

[36] H. Haibin Zhu and M. MengChu Zhou, "Role transfer problems and algorithms," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 38, no. 6, pp. 1442–1450, 2008.

[37] S. Cheng, Z. Chen, J. Li, and H. Gao, "Task assignment algorithms in data shared mobile edge computing systems," in *Proceedings of the IEEE Int. Conf. Distributed Comput. Syst., (ICDCS)*, Dallas, TX, USA, July 2019.

[38] H. Zhu, "Avoiding conflicts by group role assignment," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 4, pp. 535–547, 2016.

[39] H. Zhu, D. Liu, S. Zhang, S. Teng, and Y. Zhu, "Solving the group multirole assignment problem by improving the ILOG approach," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 12, pp. 3418–3424, 2017.