Xu *et al. EURASIP Journal on Wireless Communications and Networking* (2019) 2019:236
https://doi.org/10.1186/s13638-019-1560-8

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

A time-efficient data offloading method with privacy preservation for intelligent sensors in edge computing

Zhanyang Xu[1,2], Xihua Liu[1,2*], Gaoxing Jiang[1,2] and Bowei Tang[1,2]

Abstract

Over the past years, with the development of hardware and software, the intelligent sensors, which are deployed in the wearable devices, smart phones, and etc., are leveraged to collect the data around us. The data collected by the sensors is analyzed, and the corresponding measures will be implemented. However, due to the limited computing resources of the sensors, the overload resource usage may occur. In order to satisfy the requirements for strong computing power, edge computing, which emerges as a novel paradigm, provides computing resources at the edge of networks. In edge computing, the computing tasks could be offloaded from the sensors to the other sensors for processing. Despite the advantages of edge computing, during the offloading process of computing tasks between sensors, private data, including identity information and address, may be leaked, which threatens personal security. Hence, it is important to avoid privacy leakage in edge computing. In addition, the time consumption of offloading computing tasks affects the using experience of customers, and low time consumption makes contributions to the development of applications which are strict with time. To satisfy the above requirements, a time-efficient offloading method (TEO) with privacy preservation for intelligent sensors in edge computing is proposed. Technically, the time consumption and the offloading of privacy data are analyzed in a formalized way. Then, an improved of Strength Pareto Evolutionary Algorithm (SPEA2) is leveraged to optimize the average time consumption and average privacy entropy jointly. At last, abundant experimental evaluations are conducted to verify efficiency and reliability of our method.

Keywords: Intelligent sensors, Edge computing, Task offloading, Privacy leakage, Time consumption

1 Introduction

Nowadays, in order to make the world more intelligent and more informative, smart phones, autonomous vehicles, unmanned aerial vehicles, etc. are developed for the convenient lives of human beings [1, 2]. Many accessories make up these products, and one of the important accessories is the sensor [3, 4]. With the developments of hardware and software, the sensors become more and more intelligent. Nowadays, the intelligent sensor, which is a detecting device, senses the measured information and transforms the sensed information into required forms to meet the requirements of transmission, process, and storage. In addition, the intelligent sensors have different types, including gas sensor, humidity sensor, and radiation sensor, and each type sensor has its own function [5–7].

The accuracy of analyzing the results is based on the information volume, which is collected from different intelligent sensors [8, 9]. In order to guarantee the precision to make corresponding judgements, different types of intelligent sensors are deployed in various products and areas to collect all sorts of data based on their functions [10, 11]. However, due to the limited computing resources of sensors, the sensors cannot process all the collected data. Fortunately, edge computing, which emerges as a novel computing paradigm, provides computing resources at the edge of the networks. In edge computing paradigm, the computing tasks of sensors could be offloaded to the other sensors for processing. In this way, the efficiency of processing the collected data is greatly improved [12–14].

Despite the benefits, the time consumption is a weak point of edge computing. Though the delay, which is

*Correspondence: liuxihua710@gmail.com
[1] School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, People's Republic of China
[2] Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing, China

caused by the time consumption, in edge computing performs better than that in traditional cloud computing, the time consumption still makes effects on the using experience of the customers and the development of the industries with strict time limitation [15, 16]. For example, the sensors used in autonomous vehicles are strict with time. If the delay cannot meet the restrictions, traffic accidents will happen, and the safety of the passengers cannot be guaranteed [17, 18]. On a condition that the time consumption is low, more applications and features could be developed and put into industry [19]. Therefore, during the task offloading between sensors, how to minimize the time consumption is an important problem to be solved.

In addition, it is possible that the privacy information in the computing tasks is leaked during the transmission. Nowadays, the personal information is significant to everyone, and people have a strong awareness of privacy protection [20–22]. However, based on the present privacy-protecting techniques leveraged in edge computing, the privacy leakage can hardly be avoided [23, 24]. It is more likely that customers have the possibility to be harassed and endanger personal safety when the personal sensitive information in the computing tasks, which are offloaded between sensors in edge computing, is leaked [25, 26]. Besides, the data collected by the sensors are full of the privacy information of customers and have affected all the aspects in our lives, including the critical infrastructures (e.g., emergency systems), daily life (e.g., transportation), and personal information (e.g., address). On this condition, more sound privacy and security mechanisms need to be considered comprehensively.

The benefit from the edge computing paradigm is that the services of the intelligent sensors could be offloaded the other sensors for processing. In this way, the burden of processing a large quantity of services produced by all the sensors in real time could be eased. However, during the transmission of the services, the time consumption and privacy protection are two important aspects which need to be improved. On the one hand, the time consumption affects the using experience of customers and the development of the industries which are strict with time. The long-time consumption leads to the waste of time and the displeasure of customers. On the other hand, if the content of the offloading data is stolen during the transmission process, the private information in the tasks may be leaked. Therefore, how to release the privacy and security problems is waiting to be solved for the intelligent sensors in edge computing.

In this paper, the main contributions are listed as follows:

- Analyze the data offloading problem, which is defined as a multi-objective optimization problem, with the time consumption and the privacy entropy.
- The improving Strength Pareto Evolutionary Algorithm (SPEA2) is leveraged to make optimization of the time consumption and privacy entropy at the same time. Then, simple additive weighting (SAW) and multiple criteria decision-making (MCDM) methods are adopted to obtain the optimal schedule strategy.
- Systematic experiments are carried out to demonstrate the effectiveness and efficiency of our proposed method.

The reminder of this paper is organized as follows. In Section 2, the mathematic modeling and the formulation are described. Section 3, a time-efficient offloading method with privacy preservation, named TEO, is developed for intelligent sensors in edge computing. In Section 4, simulation experiments and comparison analysis are presented. In Section 5, related work is summarized. Finally, in Section 6, conclusion and future work are outlined.

2 System model and problem formulation

In this section, basic definitions and concepts for sensors in edge computing with privacy preservation are introduced. In addition, the time consumption and privacy entropy are also analyzed. Key terms and descriptions are shown in Table 1.

2.1 Resource model

The edge computing paradigm has the ability to satisfy the calculating problems of sensors in edge computing, which cannot avoid the privacy leakage during the trans-

Table 1 Key parameters and descriptions

Terms	Descriptions
N	The number of sensors
MD	The set of sensors, $MD = \{md_1, md_2, \ldots, md_N\}$
md_n	The nth sensor in MD
N	The number of computing tasks
K	The total types of divided data
CT	The set of computing tasks, $CT = \{ct_1, ct_2, \ldots, ct_N\}$
SD	The set of divided data in computing tasks, $SD = \left\{sd_{n,1}, sd_{n,2}, \ldots, sd_{n,K}\right\}$
$s_{n,k}$	The size of transmitted data with the kth type for ct_n
$p_{n,k}$	The probability of transferring kth type data for ct_n
$t_{n,k}$	The time of kth data transmission for ct_n
$ct_{n,k}$	The calculation time of kth type data
V_1	The data transmission rate between sensors
V_2	The data calculation rate of sensors
T	The average time of data transmission

mission of computing tasks between sensors. In addition, the data transmission rate in edge computing is denoted as V, and T represents the average time of data transmission. Figure 1 describes the system framework for data offloading of intelligent sensors in edge computing. In this framework, we consider there are N sensors, denoted as $MD = \{md_1, md_2, \ldots, md_N\}$, and M computing tasks corresponded with the sensors, denoted as $CT = \{ct_1, ct_2, \ldots, ct_N\}$, in this scenario. In each computing task, there are K types of divided data, and the probability of transferring kth type data is denoted as p_k. In addition, the data transmission rate in edge computing is denoted as V, and T represents the average time of data transmission.

2.2 Time consumption model

During the offloading of computing tasks, the time consumption is an important value, which needs to be taken into consideration. The time consumption mainly consists of the time consumed by the transmission and the calculation of the services offloaded from the intelligent sensors. Calculating the time consumption of ct_n is based on the size of transmitted data with the kth type for ct_n and the possibility of transferring kth type data.

The possibility of offloading the types of data is according to the Poisson distribution. The possibility of offloading kth type data in ct_n is denoted as $p_{n,k}$, and $p_{n,k}$ is calculated by

$$p_{n,k} = \frac{\lambda^{ct_{n,k}}}{ct_{n,k}!} e^{-\lambda}, k = 0, 1, 2, \ldots, K. \tag{1}$$

The transmission time of kth type data is calculated by

$$t_{n,k} = \frac{1}{V_1} s_{n,k}. \tag{2}$$

The transmission time of each computing task is calculated by

$$t_n = \sum_{k=1}^{K} p_{n,k} t_{n,k}, n = 0, 1, 2, \ldots, N. \tag{3}$$

The calculation time of kth type data is calculated by

$$ct_{n,k} = \frac{1}{V_2} s_{n,k}. \tag{4}$$

The calculation time of each computing task is based on the possibility of offloading kth type data and the time of kth type data, which is calculated by

$$ct_n = \sum_{k=1}^{K} p_{n,k} ct_{n,k}, n = 0, 1, 2, \ldots, N. \tag{5}$$

Finally, the average time consumption of the computing tasks is calculated by

$$T = \frac{1}{N} \cdot \sum_{n=1}^{N} (t_n + ct_n). \tag{6}$$

2.3 Privacy entropy model

To avoid the privacy leakage during the offloading of the computing tasks, the privacy entropy is leveraged to protect the privacy data. The computing tasks are divided into several types data to measure the uncertainty of the computing tasks. The larger the privacy entropy is, the more confusing the computing tasks are. Therefore, the contain of the computing tasks is safe.

All of the computing tasks are denoted as $CT_n = \{ct_{n,1}, ct_{n,2}, \ldots, ct_{n,k}\}$, where $ct_{n,k}$ represents the computing tasks in md_n, $p_{n,k}$ represents the probability of offloading kth type data in md_n. In addition, the relationship between CT_n and P_n is described by.

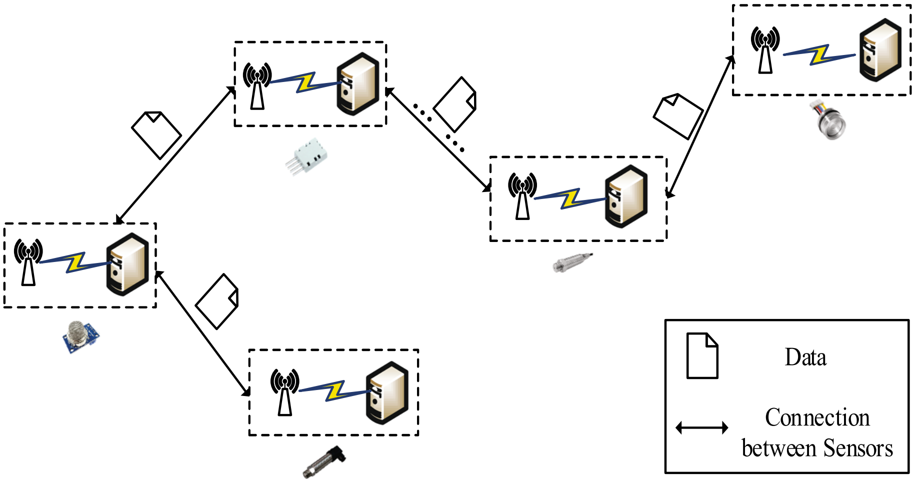

Fig. 1 A framework of data offloading for intelligent sensors in edge computing

$$\begin{pmatrix} CT_n \\ P_n \end{pmatrix} = \begin{pmatrix} ct_{n,1} & ct_{n,2} & \cdots & ct_{n,k} \\ p_{n,1} & p_{n,2} & \cdots & p_{n,k} \end{pmatrix}. \tag{7}$$

The privacy entropy of each computing task is calculated by

$$H(X_n) = -\sum_{k=1}^{K} p_{n,k}\log_2 p_{n,k}. \tag{8}$$

The privacy entropy of computing tasks is the summary of each computing task, which is calculated by

$$H(X) = \sum_{n=1}^{N} H(X_n). \tag{9}$$

At last, the average privacy entropy of the computing tasks is calculated by

$$H = \frac{1}{N} H(X). \tag{10}$$

2.4 Problem definition

In this paper, our goal is to minimize the time consumption of the computing tasks in (6) and maximize the privacy entropy in (10). The problem is formalized as follows

$$\max H(X),\ \min T. \tag{11}$$

$$s.t. \sum_{k=1}^{K} p_{n,k} = 1. \tag{12}$$

The constraint is shown in (12), which represents the probability of all the divided data equals one.

3 Data offloading method for intelligent sensors in edge computing

In this paper, a multi-objective optimization problem is proposed to improve the time consumption and privacy entropy. Compared with traditional algorithms such as genetic algorithm (GA), weighted coefficient method, SPEA2 has been widely employed to work out the optimization problems due to its parallel processing mechanism, global optimization, and good robustness. In addition, SPEA2 investigate on the calculation of the fitness of each individual, and SPEA2 leverage clustering to reduce the extra points. Therefore, SPEA2 is leveraged to find the optimal strategy. Finally, SAW and MCDM methods are used to obtain the optimal solutions.

3.1 Encoding

First, the number of the divided types of the computing tasks, which are waiting to be offloaded, is encoded in this section. In SPEA2, the gene represents the number of the computing tasks. All the genes make up the chromosome which represents the solution of the optimization problem. In addition, the chromosome is encoded in integer in this paper as shown in Fig. 2. Six genes, which correspond with the number of divided computing tasks, make up the sample chromosome. For example, the fourth gene is 6, and this gene indicates that the fourth computing task of the sensor is divided into six types.

3.2 Fitness functions and constraints

In this paper, the fitness functions include two categories: the time consumption (6) and the privacy entropy (10). The fitness functions are leveraged to evaluate the pros and cons of individuals. Based on the fitness functions, the optimal strategy could be obtained. As is shown in (11), our proposed method aims to make optimization of the time consumption and privacy entropy. Besides, the constraint is shown in (12). The constraint means that all the probabilities of the divided data should be equal to 1.

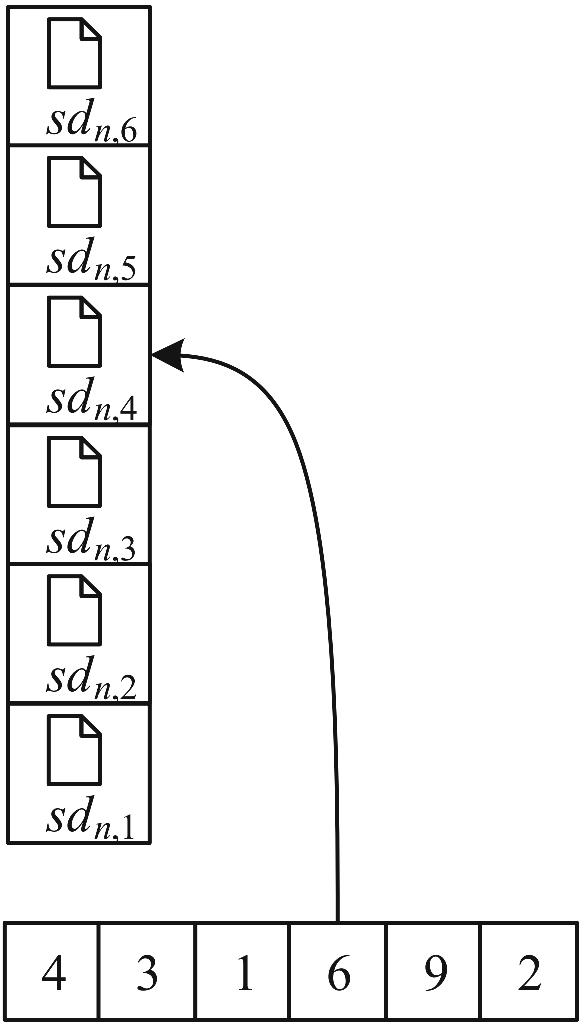

Fig. 2 An example of encoding operation

In Algorithm 1, the average time consumption of data could be calculated at last. The $s_{n,k}$, V_1, and V_2 need to be input into this algorithm. The transmission time of each computing task is calculated by line 3, and the calculation time of each computing task is calculated by line 4. Then, all the time consumption of the computing tasks is calculated by line 5, and the average time consumption of computing tasks is calculated by line 6 (Figs. 3 and 4).

In Algorithm 2, the average privacy entropy for all the offloading tasks is calculated at last. The pn,k needs to be input into this algorithm. The privacy entropy of each computing task is calculated by line 3. Then, all the privacy entropy of computing tasks is calculated by line 4, and the average privacy entropy for all the offloading tasks is calculated by line 7.

3.3 Initialization

In this part, some paraments need to be determined. The paraments are the size of population *SP*, the probability of crossover *PC*, the probability of mutation *PM*, the size of archive *SA*, and the number of iterations *NI*.

3.4 Selection

In this operation, the individuals, who have with better fitness, are selected from the evolutionary group into the mating pool. The following crossover and mutation operations just obtain the individuals from the mating pool to generate better populations.

3.5 Crossover and mutation

The crossover operation, which has two procedures, is to combine two parental chromosomes to generate two new chromosomes. The first procedure is to pick a crossover point randomly form one to the number of genes in chromosome. The second procedure is to change the two chromosomes at this crossover point. Fig. 3 illustrates an example of crossover operation.

If the offspring chromosome cannot perform better than their parental chromosome but is not the global optimal solution, the premature convergence will take place. The mutation operation is utilized to guarantee the individual diversity. In addition, the probability of each mutated gene is the same. Fig. 4 illustrates an example of mutation operation.

Algorithm 1 The Average Time of Data Transmission Obtaining

```
Require: $s_{n,k}$, $v$
Ensure: $T$
1: for $n = 1$ to $N$ do
2:     for $k = 1$ to $K$ do
3:         Calculate $t_n$ by formula (3)
4:         Calculate $ct_n$ by formula (5)
5:         $T = t_n + ct_n$
6:     end for
7: end for
8: $T = T/N$
9: return $T$
```

3.6 Schedule selection using sAW and mCDM

In this paper, the positive criterion is privacy entropy, and the negative criterion is time consumption. With the increase of the positive criterion, the method will perform well. Otherwise, with the increase of the negative criterion, the method will perform badly. The probability composition set is denoted as *PC*, and each probability strategy in *PC* affects the time consumption and the privacy entropy. The time consumption value is denoted as $TT = (TT_{i,j}, 1 \le i \le Q)$, and the privacy entropy value is denoted as $PE = (PE_{i,l}, 1 \le i \le Q)$. The scaling value of the negative criterion of time consumption is denoted as $TT_{i,j}$, and the scaling value of the positive criterion of privacy entropy is denoted as $PE_{i,l}$. In addition, $TT_{i,j}$ and $PE_{i,l}$ are calculated by

$$TT_{i,j} = \left\{ \begin{array}{ll} \frac{N^{\max}-N^{i,j}}{N^{\max}-N^{\min}} & ,\text{if } N^{\max}-N^{\min} \neq 0 \\ 1 & ,\text{if } N^{\max}-N^{\min} = 0 \end{array} \right\}, \tag{13}$$

$$PE_{i,l} = \left\{ \begin{array}{ll} \frac{P^{i,l}-P^{\min}}{P^{\max}-P^{\min}} & ,\text{if } P^{\max}-P^{\min} \neq 0 \\ 1 & ,\text{if } P^{\max}-P^{\min} = 0 \end{array} \right\}, \tag{14}$$

where $N^{\max}$, $N^{\min}$, $P^{\max}$, and $P^{\min}$ represent the maximum time consumption, the minimum time consumption, the maximum privacy entropy, and the minimum privacy entropy, respectively.

Finally, based on the formulas (13) and (14), the utility value is calculated by

$$UV_i = TT_{i,j} \cdot wt + \mathrm{PE}_{i,l} \cdot wp, \tag{15}$$

where *wt* and *wp* represent the weight of the transmission time and the privacy entropy in the probability composition, respectively.

Based on the above analysis, the utility value of each strategy is calculated. Then, the MCDM method is leveraged to select the best one in the utility value. The composition of the number of the computing tasks, which has the maximum utility value, is the optimal strategy.

3.7 Method review

This paper aims to make optimization of the time consumption and privacy entropy for the data offloading of intelligent sensors in edge computing. SPEA2 is leveraged to solve this problem because SPEA2 performs well in multi-optimization problems. At first, the offloading probability of each divided data needs to be encoded. Afterwards, the fitness functions and the constraints should be listed for our optimization problem. Then,

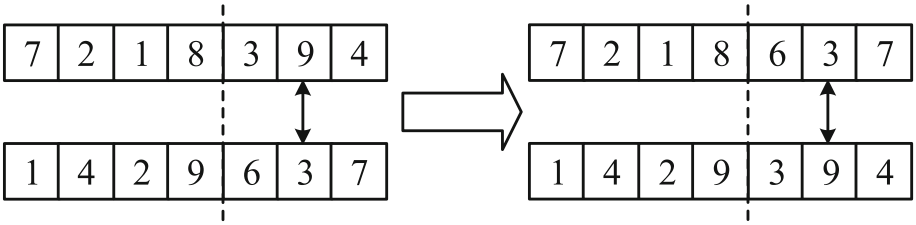

Fig. 3 An example of crossover operation

the premature convergence will be avoided by using the crossover and mutation operations, and new individuals will be generated. At last, the optimal transmission probability of the divided data will be obtained after leveraging the SAW and MCDM methods.

4 Experimental evaluation

In this section, comprehensive and abundant simulations and experiments are conducted to make evaluations of the performance of our proposed edge computing offloading method TEO. More specifically, simulation setups are introduced, including the statements of comparative methods and simulation parameter settings. Afterwards, the influence of different computing task scales on the time consumption and privacy entropy performance between the compared methods and TEO method is evaluated.

4.1 Experimental context

In this experiment, the server is based on DELL latitude7390, and the configurations are listed as follows: i7 CPU, 256 G solid-state drive and 8 GB memory. The power of the server and VM set as 64 W and 6 W, respectively. In Table 2, three basic parameters, which are used in our experiment, are listed in it. In order to ensure the effectiveness of the experiments, five different numbers of divided computing tasks are set to generate five different scale datasets, and the computing task scales are set as 5, 10, 15, 20, and 25, respectively.

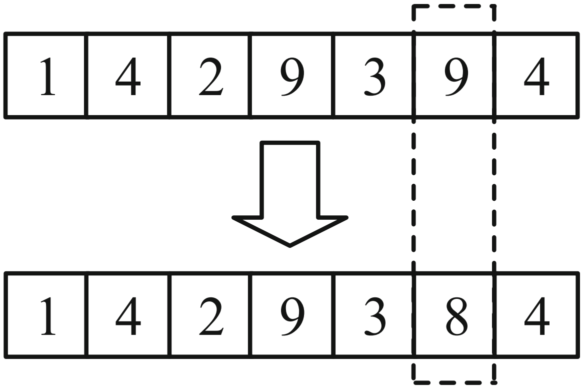

Fig. 4 An example of mutation operation

In order to carry out the comparison analysis, one basic method is selected to compare with our TEO method. The comparative method is Benchmark: The computing tasks is divided to K data. The probability of transferring kth data is equal, and the probability of transferring all the data equals 1 too.

4.2 Comparison of the time consumption

Technically, the transmission time and the calculation time make up the time consumption. Based on the analysis of the experiment data, as the number of the computing tasks increases, the time consumption will increase, and the growth rate increases faster in both of the two methods. The time consumption in our proposed method is 0.64, 1.52, 2.42, 3.34 and 4.41 (s), and the time consumption in Benchmark is 0.98, 1.78, 2.94, 3.98 and 5.3 (s) when the number of computing tasks is 5, 10, 15, 20 and 25, respectively. It is concluded that thetime consumption in our proposed method performs better than that in Benchmark, and the difference of the time consumption between the two methods is shown in Fig. 5.

4.3 Comparison of the transmission time

The transmission time mainly depends on the transmission rate between sensors. Therefore, in order to optimize the transmission time, the technique of improving

Algorithm 2 The Average Privacy Entropy for Offloading Tasks Obtaining

```
Require: $p_{n,k}$
Ensure: $H$
for $n = 1$ to $N$ do
    for $k = 1$ to $K$ do
        Calculate $H(X_n)$ by formula (8)
        $H(X) += H(X_n)$
    end for
end for
$H = H(X)/N$
return $H$
```

Table 2 Parameter settings

Parameter description	Value
The number of divided computing tasks	{5, 10, 15, 20, 25}
The transmission rate between sensors	120 Mb/s
The calculation rate of sensors	480 Mb/s
The size of divided computing tasks (m)	[1,50]

the transmission rate needs to be investigated further. In Fig. 6, the difference of the transmission time between the two methods is just a bit. With the increase of the computing tasks, the gap between the two methods becomes clearer. The transmission time in our proposed method is 0.51, 1.13, 1.84, 2.53 and 3.28 (s), and the time consumption in Benchmark is 0.75, 1.33, 2.18, 2.95 and 4.02 (s) when the number of computing tasks is 5, 10, 15, 20 and 25, respectively. The difference of the transmission time between the two methods is shown in Fig. 6.

4.4 Comparison of the time consumption

The calculation time depends on the calculation rate of sensors. On this condition that the processing ability of the sensors is improved, the calculation time will be improved too. Based on the analysis of Fig. 7, the calculation time is less than the transmission time. Same as the transmission time, with the increase of the computing tasks, the difference of the calculation time between the two methods becomes clearer too. The calculation time in our proposed method is 0.14, 0.39, 0.59, 0.81 and 1.13 (s), and the time consumption in Benchmark is 0.23, 0.45, 0.76, 1.03 and 1.29 (s) when the number of computing tasks is 5, 10, 15, 20 and 25, respectively. The difference of the calculation time between the two methods is shown in Fig. 7.

Algorithm 3 Time-Efficient Data Offloading Method with Privacy Preservation

```
Require: $N$
Ensure: $P$
for $n = 1$ to $N$ do
    $n = 1$
    while $n <= N$ do
        Crossover and mutation operation
        for the individuals in the population do
            Calculate the time of data transmission by formula (1)-(6)
            Calculate the information entropy by (7)-(10)
        end for
        Selection operation to ensure the child generation
        $n = n + 1$
    end while
    Evaluate utility function to pick out the optimal schedule strategy $P$
end for
return $P$
```

4.5 Comparison of the privacy entropy

In this experiment, the privacy entropy indicates the probability of causing the privacy leakage. With the increase of the privacy entropy, the safety of the migration is upgraded. The high privacy entropy makes the computing tasks more dispersed, which protects the safety of the important data in the tasks. In Fig. 8, with the increase of the tasks scales, the privacy entropy will increase, and the growth rate in the two methods is almost linear. However, the difference of the privacy entropy between the two method is almost same. The privacy entropy of our proposed method is 15.8, 30.96, 46.16, 61.35 and 76.28 and the privacy entropy in Benchmark is 15, 30, 45, 60 and 75 when the number of computing tasks is 5, 10, 15, 20 and 25, respectively. Though the privacy entropy is quite the same, our proposed method performs better in terms of the time consumption. The difference of the privacy entropy between the two methods is shown in Fig. 8.

4.6 Comparison of the average time consumption

In this experiment, the average time consumption is calculated by the number of computing tasks and the transmission rate. The low time consumption indicates that the time consumption of each computing tasks is saved. From the analysis of the average time consumption, the average time consumption in our proposed method is lower than that in Benchmark. The average time consumption of our proposed method is 0.06, 0.15, 0.25, 0.35, and 0.44 s when the number of computing tasks is 5, 10, 15, 20, and 25, respectively. The difference of the average time consumption between the two methods is shown in Fig. 9.

4.7 Comparison of the average privacy entropy

In this experiment, the average privacy entropy is calculated by the total privacy entropy and the number of computing tasks, and it represents the complexity of each computing task. The high value of the average privacy entropy also indicates the high-level safety of the transmission as the value of the privacy entropy. From the analysis of the average privacy entropy, it is concluded that our proposed method performs better than the Benchmark. The average privacy entropy of our proposed method is 1.6, 3.21, 4.81, 6.42, and 8.02 when the number of computing tasks is 5, 10, 15, 20, and 25, respectively. The difference of the average privacy entropy between the two methods is shown in Fig. 10.

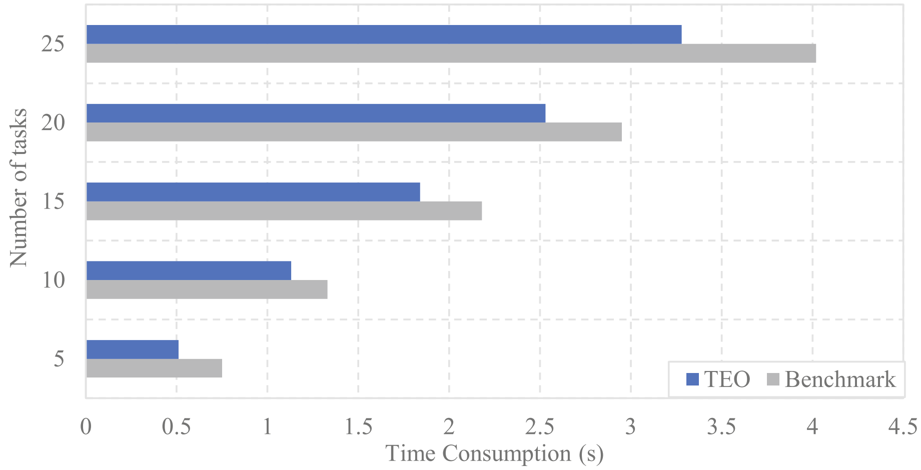

Fig. 5 Comparison of the time consumption

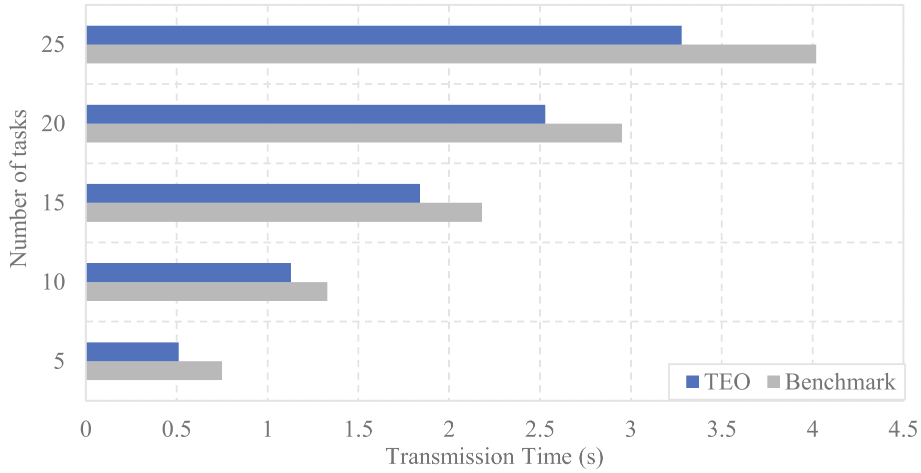

Fig. 6 Comparison of the transmission time

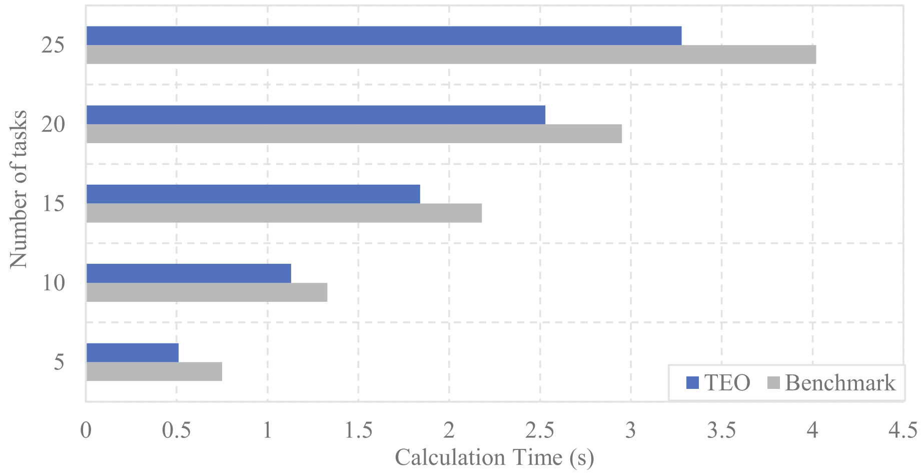

Fig. 7 Comparison of the calculation time

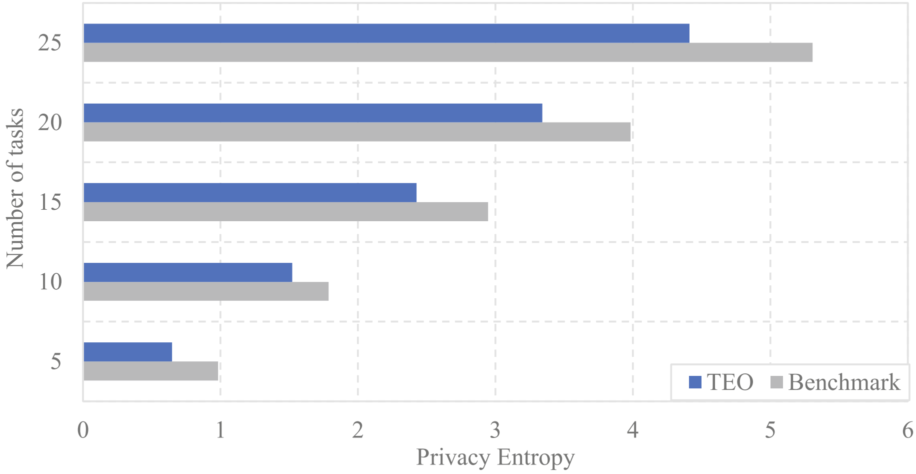

Fig. 8 Comparison of the privacy entropy

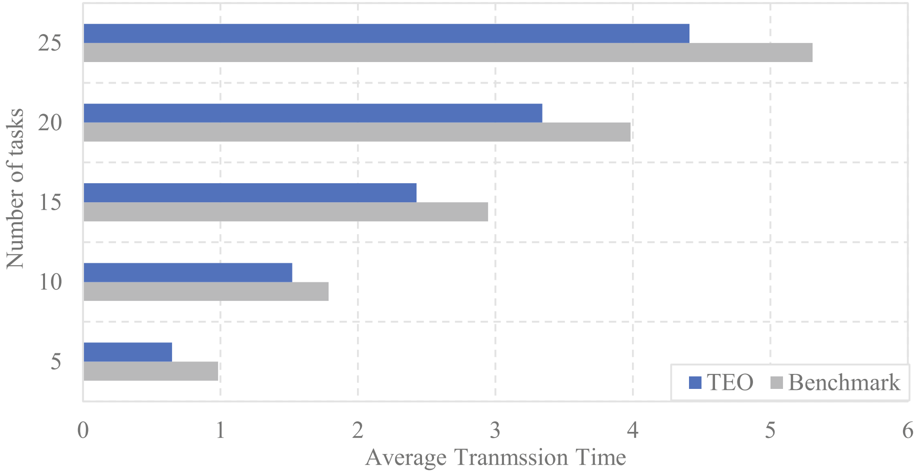

Fig. 9 Comparison of the average time consumption

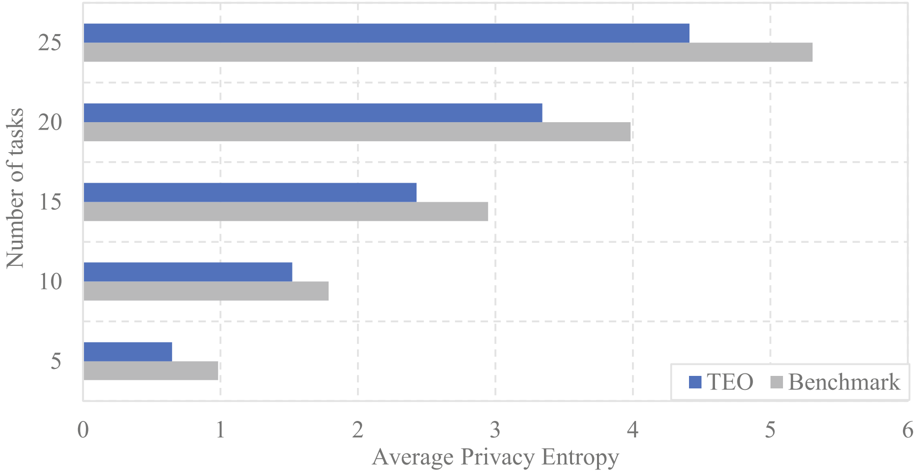

Fig. 10 Comparison of the average privacy entropy

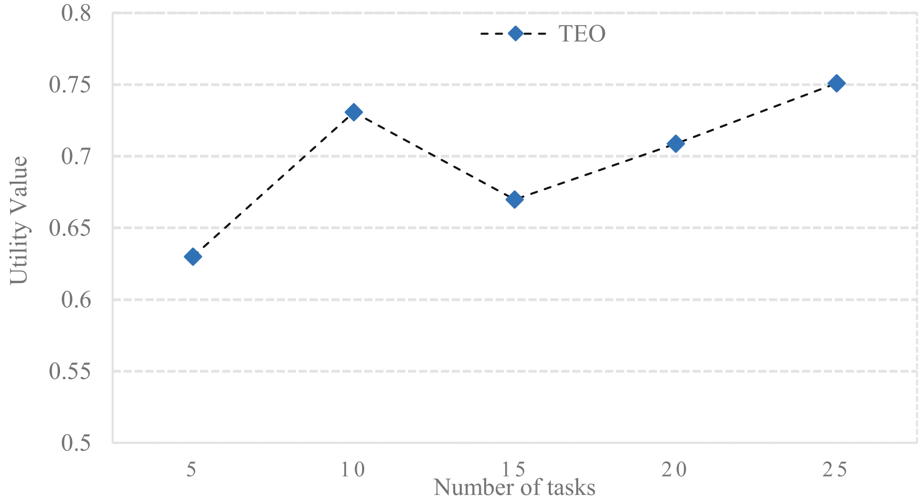

Fig. 11 Evaluation of the utility value

4.8 Evaluation of the utility value

In this experiment, the utility value is calculated by the SAW and MCDM methods. The high-utility value represents the good performance of the strategy. It is concluded that with the increase of the number of tasks, the utility value of the strategy will increase too. When the number of the tasks is 10, it performs better than the other four scales. The utility value of our proposed method is 0.63, 0.73, 0.66, 0.70, and 0.75 when the number of computing tasks is 5, 10, 15, 20, and 25, respectively. The utility value in our method is shown in Fig. 11.

5 Related work

With the rapid development of mobile applications, cloud computing can no longer meet the need of customers for real-time computing [27–29]. Edge computing has emerged as an outcome paradigm to release this problem. Due to its good performance, edge computing paradigm is widely accepted. Due to the limited CPU resources and computing power, the data-processing capacity of sensors is scant [30]. Therefore, more and more mobile users could offload their computing tasks to the other sensors for processing in edge computing [31, 32]. In this way, the local computing power and the time consumption are saved.

However, the time consumption of task offloading is a problem waiting to be processed. The time consumption affect the experience of mobile users directly [33, 34]. In order to make full use of the computing resources in edge computing reasonably, an appropriate resource schedule need to be designed.

In [35], the task offloading problem is formulated by Chen et al. as a mixed integer nonlinear program. In addition, this program is NP-hard. The optimization problem, which aims to minimize the time consumption and the energy consumption, was transformed into resource allocation problem and task placement problem. In [36], Tao et al. investigated to reduce energy consumption while guaranteeing the performance in mobile edge computing. They applied KKT conditions to work out this problem and proposed an offloading scheme. In [37], in order to make the computing resources closer to edge than cloud computing, Sun et al. introduced a cloudlet network architecture. In this architecture, each user equipment could make communication with its Avatar, which is a clone of the software located in the cloudlet, and reduce the delay between user equipment. In [38], Hao et al. focused on the computation offloading and introduced a new task-caching concept. Besides, they proposed task caching and offloading (TCO) method, and this method is based on the alternating iterative algorithm. In [39], Wang et al. formulated an optimization problem, which aims to improve the content caching strategy, resource allocation, and computation offloading decision, while taking the revenue of network into consideration. In addition, in order to work out this problem in an efficient way, the problem was transformed into a convex problem.

On the other hand, the privacy leakage is another problem, which may lead to the information crime. The computing tasks may contain some privacy information, including personal data and current locations [40, 41]. Therefore, it is of utmost significance to avoid privacy leakage during the task offloading in edge computing.

In [42], Duan et al. proposed a platform called SDN into 5G, which aims to satisfy the authentication handover and privacy protection. In order to simplify the authentication handover, the platform is to share the security information between the related access points. In [43], Eiza et al. formulated a novel system model, and this model is leveraged in 5G-enabled vehicular network. This system model could offer users with privacy-aware, reliable, and real-time video service. In [44], Norrman et al. introduced a method, which set up a pseudonym between user equipment and home network, to make protection of the IMSI in 5 G. In [45], Shen et al. aimed to optimize the task acceptance rate while avoiding the privacy leakage of participants by leveraging edge nodes. In addition, they proposed a privacy-preserving task allocation framework (P2TA) to improve the edge computing-enhanced MCS. In [46], Lu et al. formulated a lightweight privacy-preserving data aggregation scheme in the fog computing-enhanced IoT framework. In [47], Gai et al. aimed to solve the conflict problem between the computing efficiency and privacy protection. Besides, they proposed a new approach to offer safe transmission based on the multi-channel communication technology.

To the best of our knowledge, few studies and researches have verified the performance of edge computing. In addition, due to the importance of the transmission time and privacy security, it is necessary to consider both of them. The greater the privacy entropy is, the more confusing the computing tasks are. Thus, this paper uses the privacy entropy to address the privacy problem while minimizing the transmission time of computing tasks.

6 Conclusion and future work

In recent years, edge computing has emerged as an important paradigm to provide strong computing power for mobile devices and sensors. Guaranteed by the edge computing, many novel technologies have been introduced to improve the quality of the services. Edge computing has the advantages of processing the computing tasks effectively to provide real-time services. In order to minimize the time consumption and maximize the privacy entropy, a time-efficient offloading method (TEO) with privacy preservation is proposed for sensors in edge computing. First, the computing tasks of sensors, which are waiting to be offloaded, are divided into several types. Each type data has different probability values to offload these tasks. Then, a time-efficient offloading method is proposed. In this method, SPEA2 is leveraged to make optimization of the time consumption and privacy entropy jointly, and SAW and MCDM methods are used to find the optimal offloading strategy. The experimental evaluations verify the reliability of our proposed method.

For future work, we will adjust and extend our proposed method to the real-world scenario. In addition, we will try to analyze the energy consumption; thus, we improve the time consumption, privacy protection, and energy consumption at the same time.

Abbreviations

MCDM: Multiple criteria decision-making; SAW: Simple additive weighting; SPEA2: Improving the Strength Pareto Evolutionary Algorithm

Acknowledgements

This research is supported by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund,and Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology (CICAEET).

Authors' contributions

ZX, XL, GJ, and BT conceived and designed the study. ZX and XL performed the simulations. GJ and BT wrote the paper. All authors reviewed and edited the manuscript. All authors read and approved the final manuscript.

Availability of data and materials

The dataset supporting the conclusions of this article is available, which can downloaded at https://pan.baidu.com/s/1RB5uBejvJsyCD2GkD4jiiA.password: xwaa.

Competing interests

The authors declare that they have no competing interests.

Received: 2 July 2019 Accepted: 11 September 2019
Published online: 24 October 2019

References

1. I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, S. U. Khan, The rise of "big data" on cloud computing: Review and open research issues. Inf. Syst. **47**, 98–115 (2015)
2. Z. Gao, D. Wang, S. Wan, H. Zhang, Y. Wang, Cognitive-inspired class-statistic matching with triple-constrain for camera free 3D object retrieval. Futur. Gener. Comput. Syst. **94**, 641–653 (2019)
3. X. Xu, Y. Li, T. Huang, Y. Xue, K. Peng, L. Qi, W. Dou, An energy-aware computation offloading method for smart edge computing in wireless metropolitan area networks. J. Netw. Comput. Appl. **133**, 75–85 (2019)
4. A. Botta, W. De Donato, V. Persico, A. Pescapé, Integration of cloud computing and internet of things: A survey. Futur. Gener. Comput. Syst. **56**, 684–700 (2016)
5. X. Xu, D. Li, Z. Dai, S. Li, X. Chen, A heuristic offloading method for deep learning edge services in 5G networks. IEEE Access. **7**, 67734–67744 (2019). https://doi.org/10.1109/access.2019.2918585
6. Y. Yuan, W. Banzhaf, Arja: Automated repair of Java programs via multi-objective genetic programming. arXiv preprint arXiv:1712.07804 (2017)
7. M. Satyanarayanan, The emergence of edge computing. Computer. **50**(1), 30–39 (2017)
8. Z. Gao, H.-Z. Xuan, H. Zhang, S. Wan, K.-K. R. Choo, Adaptive fusion and category-level dictionary learning model for multi-view human action recognition. IEEE Int. Things J., 1–1 (2019). https://doi.org/10.1109/jiot.2019.2911669
9. S. Wan, Y. Zhao, T. Wang, Z. Gu, Q. H. Abbasi, K.-K. R. Choo, Multi-dimensional data indexing and range query processing via Voronoi diagram for internet of things. Futur. Gener. Comput. Syst. **91**, 382–391 (2019)
10. N. Abbas, Y. Zhang, A. Taherkordi, T. Skeie, Mobile edge computing: A survey. IEEE Int. Things J. **5**(1), 450–465 (2017)
11. L. Wang, H. Zhen, X. Fang, S. Wan, W. Ding, Y. Guo, A unified two-parallel-branch deep neural network for joint gland contour and segmentation learning. Futur. Gener. Comput. Syst. **100**, 316–324 (2019)
12. Y. Yuan, Y.-S. Ong, A. Gupta, H. Xu, Objective reduction in many-objective optimization: Evolutionary multiobjective approaches and comprehensive analysis. IEEE Trans. Evol. Comput. **22**(2), 189–210 (2017)
13. Y. Mao, J. Zhang, K. B. Letaief, Dynamic computation offloading for mobile-edge computing with energy harvesting devices. IEEE J. Sel. Areas Commun. **34**(12), 3590–3605 (2016)
14. T. H. Luan, L. Gao, Z. Li, Y. Xiang, G. Wei, L. Sun, Fog computing: Focusing on mobile users at the edge. arXiv preprint arXiv:1502.01815 (2015)
15. S. Ding, S. Qu, Y. Xi, S. Wan, A long video caption generation algorithm for big video data retrieval. Futur. Gener. Comput. Syst. **93**, 583–595 (2019)
16. W. Li, X. Liu, J. Liu, P. Chen, S. Wan, X. Cui, On improving the accuracy with auto-encoder on conjunctivitis. Appl. Soft Comput. **81**, 105489 (2019). https://doi.org/10.1016/j.asoc.2019.105489

17. T. X. Tran, A. Hajisami, P. Pandey, D. Pompili, Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges. arXiv preprint arXiv:1612.03184 (2016)
18. X. Xu, S. Fu, Y. Yuan, Y. Luo, L. Qi, W. Lin, W. Dou, Multiobjective computation offloading for workflow management in cloudlet-based mobile cloud using NSGA-II. Comput. Intell. **35**(3), 476–495 (2018). https://doi.org/10.1111/coin.12197
19. X. Xu, Y. Xue, L. Qi, Y. Yuan, X. Zhang, T. Umer, S. Wan, An edge computing-enabled computation offloading method with privacy preservation for internet of connected vehicles. Futur. Gener. Comput. Syst. **96**, 89–100 (2019)
20. Y. Yuan, H. Xu, B. Wang, B. Zhang, X. Yao, Balancing convergence and diversity in decomposition-based many-objective optimizers. IEEE Trans. Evol. Comput. **20**(2), 180–198 (2015)
21. T. Taleb, S. Dutta, A. Ksentini, M. Iqbal, H. Flinck, Mobile edge computing potential in making cities smarter. IEEE Commun. Lett. **55**(3), 38–43 (2017). https://doi.org/10.1109/mcom.2017.1600249cm
22. X. Xu, Y. Chen, Y. Yuan, T. Huang, X. Zhang, L. Qi, Blockchain-based cloudlet management for multimedia workflow in mobile cloud computing. Multimedia Tools Appl., 1–26 (2019). https://doi.org/10.1007/s11042-019-07900-x
23. K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, Y. Zhang, Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks. IEEE Access. **4**, 5896–5907 (2016)
24. J. Liu, W. Wang, D. Li, S. Wan, H. Liu, Role of gifts in decision making: An endowment effect incentive mechanism for offloading in the IOV. IEEE Int. Things J. **6**(4), 6933–6951 (2019). https://doi.org/10.1109/jiot.2019.2913000
25. J. Liu, Y. Mao, J. Zhang, K. B. Letaief, in *2016 IEEE International Symposium on Information Theory (ISIT)*. Delay-optimal computation task scheduling for mobile-edge computing systems (IEEE, 2016), pp. 1451–1455. https://doi.org/10.1109/isit.2016.7541539
26. Z. Yang, Y. Huang, X. Li, W. Wang, F. Wu, X. Zhang, W. Yao, Z. Zheng, L. Xiang, W. Li, et al., Efficient secure data provenance scheme in multimedia outsourcing and sharing. Comput. Mater. Contin. **56**(1), 1–17 (2018)
27. X. Xu, S. Fu, L. Qi, X. Zhang, Q. Liu, Q. He, S. Li, An IOT-oriented data placement method with privacy preservation in cloud environment. J. Netw. Comput. Appl. **124**, 148–157 (2018)
28. S. Sardellitti, G. Scutari, S. Barbarossa, Joint optimization of radio and computational resources for multicell mobile-edge computing. IEEE Trans. Signal Inf. Process. Over Netw. **1**(2), 89–103 (2015)
29. A. C. Baktir, A. Ozgovde, C. Ersoy, How can edge computing benefit from software-defined networking: A survey, use cases, and future directions. IEEE Commun. Surv. Tutor. **19**(4), 2359–2391 (2017)
30. X. Xu, Q. Cai, G. Zhang, J. Zhang, W. Tian, X. Zhang, A. X. Liu, An incentive mechanism for crowdsourcing markets with social welfare maximization in cloud-edge computing. Concurr. Comput. Pract. Experience, e4961 (2018). https://doi.org/10.1002/cpe.4961
31. F. Wang, J. Xu, X. Wang, S. Cui, Joint offloading and computing optimization in wireless powered mobile-edge computing systems. IEEE Trans. Wirel. Commun. **17**(3), 1784–1797 (2017)
32. B. Varghese, N. Wang, S. Barbhuiya, P. Kilpatrick, D. S. Nikolopoulos, in *2016 IEEE International Conference on Smart Cloud (SmartCloud)*. Challenges and opportunities in edge computing (IEEE, 2016), pp. 20–26. https://doi.org/10.1109/smartcloud.2016.18
33. X. Xu, S. Fu, Q. Cai, W. Tian, W. Liu, W. Dou, X. Sun, A. X. Liu, Dynamic resource allocation for load balancing in fog environment. Wirel. Commun. Mob. Comput. **2018**, 1–15 (2018). https://doi.org/10.1155/2018/6421607
34. X. Xu, R. Huang, R. Dou, Y. Li, J. Zhang, T. Huang, W. Yu, Energy-efficient cloudlet management for privacy preservation in wireless metropolitan area networks. Secur. Commun. Netw. **2018**, 1–13 (2018). https://doi.org/10.1155/2018/8180451
35. M. Chen, Y. Hao, Task offloading for mobile edge computing in software defined ultra-dense network. IEEE J. Sel. Areas Commun. **36**(3), 587–597 (2018)
36. X. Tao, K. Ota, M. Dong, H. Qi, K. Li, Performance guaranteed computation offloading for mobile-edge cloud computing. IEEE Wirel. Commun. Lett. **6**(6), 774–777 (2017)
37. X. Sun, N. Ansari, in *2016 IEEE International Conference on Communications (ICC)*. Primal: Profit maximization avatar placement for mobile edge computing (IEEE, 2016), pp. 1–6. https://doi.org/10.1109/icc.2016.7511131
38. Y. Hao, M. Chen, L. Hu, M. S. Hossain, A. Ghoneim, Energy efficient task caching and offloading for mobile edge computing. IEEE Access. **6**, 11365–11373 (2018)
39. C. Wang, C. Liang, F. R. Yu, Q. Chen, L. Tang, Computation offloading and resource allocation in wireless cellular networks with mobile edge computing. IEEE Trans. Wirel. Commun. **16**(8), 4924–4938 (2017)
40. X. Xu, X. Liu, L. Qi, Y. Chen, Z. Ding, J. Shi, Energy-efficient virtual machine scheduling across cloudlets in wireless metropolitan area networks. Mob. Netw. Appl., 1–15 (2019). https://doi.org/10.1007/s11036-019-01242-6
41. X. Xu, Q. Liu, Y. Luo, K. Peng, X. Zhang, S. Meng, L. Qi, A computation offloading method over big data for IOT-enabled cloud-edge computing. Futur. Gener. Comput. Syst. **95**, 522–533 (2019)
42. X. Duan, X. Wang, Authentication handover and privacy protection in 5G HetNets using software-defined networking. IEEE Commun. Mag. **53**(4), 28–35 (2015)
43. M. H. Eiza, Q. Ni, Q. Shi, Secure and privacy-aware cloud-assisted video reporting service in 5G-enabled vehicular networks. IEEE Trans. Veh. Technol. **65**(10), 7868–7881 (2016)
44. K. Norrman, M. Näslund, E. Dubrova, in *Proceedings of the 9th EAI International Conference on Mobile Multimedia Communications*. Protecting IMSI and user privacy in 5G networks, (2016), pp. 159–166. https://doi.org/10.4108/eai.18-6-2016.2264114
45. H. Shen, G. Bai, Y. Hu, T. Wang, P2ta: Privacy-preserving task allocation for edge computing enhanced mobile crowdsensing. J. Syst. Archit. **97**, 130–141 (2019). https://doi.org/10.1016/j.sysarc.2019.01.005
46. R. Lu, K. Heung, A. H. Lashkari, A. A. Ghorbani, A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced IOT. IEEE Access. **5**, 3302–3312 (2017)
47. K. Gai, M. Qiu, Z. Xiong, M. Liu, Privacy-preserving multi-channel communication in edge-of-things. Futur. Gener. Comput. Syst. **85**, 190–200 (2018)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.