electronics

MDPI

Article

Efficient Resource Allocation for Security-Aware Task Offloading in MEC System Using DVS

Yanli Wang [1], Wanli Zhang [2], Haiquan Deng [3] and Xianwei Li [4,*]

[1] School of Computer Science and Technology, Weinan Normal University, Weinan 714099, China
[2] School of Information Engineering, Suzhou University, Suzhou 234000, China
[3] China Mobile Group Hunan Company Limited, Chenzhou Branch, Chenzhou 423000, China
[4] School of Computer and Information Engineering, Bengbu University, Bengbu 233000, China
* Correspondence: lixianwei163@163.com

Abstract: With the Internet of Things (IoT) and communication technologies are snowballing, various applications (e.g., e-health and face recognition) are generated by IoT devices (IoTDs). Nevertheless, these IoTDs generally have constrained computation resources. By offloading the IoT applications to be processed by the MEC servers, mobile edge computing (MEC) is envisioned as a promising and effective solution to address this problem. Meanwhile, security is a critical issue for task offloading in MEC. While plenty of studies have focused on IoT tasks offloading, many of them ignored the security issue. Moreover, many previous works ignored the resource allocation of MEC servers. In addition, as dynamic voltage scaling (DVS) technology is flexible in the design of MEC systems, we integrate this technology with task offloading. In this paper, the problem of IoT applications offloading in an MEC system is studied, whose goal is to minimize computation overheads measured by the task processing delay and energy consumption of IoTDs. The AES cryptographic technique is adopted to make sure that the security of the data of the offloaded tasks is guaranteed. An optimization problem of security-aware task offloading is formulated and solved by proposing an efficient resource-allocation scheme. Experimental results are performed to evaluate and confirm the performance of the proposed security model.

Keywords: MEC; DVS; task offloading; security

Citation: Wang, Y.; Zhang, W.; Deng, H.; Li, X. Efficient Resource Allocation for Security-Aware Task Offloading in MEC System Using DVS. *Electronics* **2022**, *11*, 3032. https://doi.org/10.3390/electronics11193032

Academic Editor: Claus Pahl

Received: 14 August 2022
Accepted: 20 September 2022
Published: 23 September 2022

1. Introduction

In the past few years, the number of Internet of Things (IoT) devices (IoTDs), such as wearable devices, has risen explosively [1]. According to the IOT ANALYTCS, the number of active IoTDs will grow to 22 billion by 2025 [2], as Figure 1 illustrates. The growing number of IoTDs has resulted in various IoT applications, which are typically delay sensitive and require a lot of computation resources. However, most of these IoT devices usually have limited resources and constrained battery power [3,4], which imposes a bottleneck for processing IoT applications. Although the remote public cloud (e.g., Amazon EC2) can provide abundant computing resources for processing IoT applications, their data centers are far away from the IoTDs, which will cause a long transmission latency. Therefore, the distant public cloud may not be suitable for latency-sensitive IoT applications [5].

To this end, the mobile edge computing (MEC) platform has been put forward [6]. In contrast to the remote public cloud, the servers of the edge cloud are implemented at a base station (BS), which are much closer to the IoTDs. Hence, the transmission latency can be significantly reduced. Meanwhile, by way of offloading computation-intensive IoT tasks to the edge cloud, the energy consumption of IoTDs can also be reduced. Therefore, offloading IoT tasks to the edge cloud can lower the processing latency as well as energy consumption.

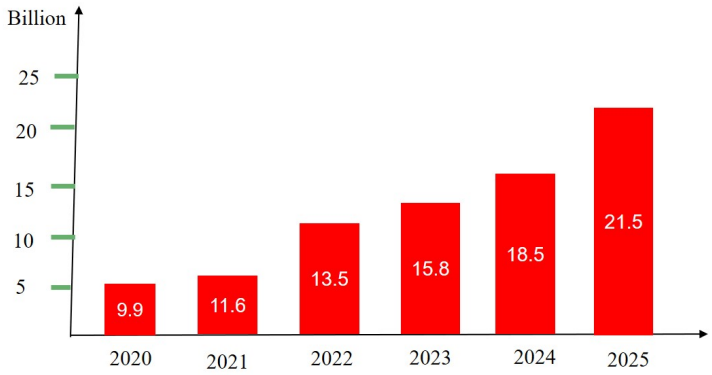

Figure 1. The active devices' number connections worldwide [2].

In an MEC system, designing efficient resource-allocation schemes for task offloading is a hot research topic studied by a significant amount of works, such as [7–9]. However, many of them ignored the issue of data security, as the IoT users are concerned about privacy leakage when offloading applications to the edge cloud [10]. Although some of the related works, such as [11,12], did not jointly consider the DVS technology and the computing resources that are hosted in the edge server. In [13], the authors investigated the joint allocation of power and computing resources in an MEC system with multiple cells and edge servers. They did not take the data-security issue and DVS technology into account.

In this article, we study IoTDs' task offloading in an MEC system by allocating the resources of the edge MEC server and adopting the DVS technology. The objective of our study is minimizing the execution time of tasks and the consumed energy by IoTDs, which is an optimization problem. As its objective function is non-convex, an algorithm is proposed to derive the optimal solution.

In a nutshell, we make the following contributions.

- We study the problem of IoTDs' task offloading in a security-aware MEC system by allocating resources of the MEC server and adopting DVS technology. In particular, we take security issues into account, and the IoTDs adopt the DVS technology to reduce the task's execution time and the consumed energy. Our objective is to obtain the minimization of the task-processing delay and the consumed energy for all IoTDs. We formulate this problem using an optimization problem.
- Since the objective function for the formulated problem is mixed-integer nonlinear programming (MINP), it is well-known as non-convex and NP-hard. This kind of problem is difficult to be solved. We propose an efficient algorithm by dividing the formulated problem into several sub-problems. The optimal solution for each sub-problem can be efficiently solved.
- The theoretical analysis and experimental results verify the performance of our proposed solution scheme.

2. Related Work

Recently, IoTDs' task offloading and resource allocation have gained an enormous amount of research interest in recent years. We present a review of some works according to their objectives.

The minimization of delay or energy consumption. In [14], Gao et al. proposed a two-stage computing method using the aggregative game to minimize the processing latency of tasks while considering the proper management of a server load. In [15], Han et al. studied power-consumption minimization with the constraints of average latency by considering the MEC system dynamics. Lyu et al. studied resource-allocation and task-admission problems to minimize the total energy consumption considering delay constraints in MEC [16]. Their formulated problem is an integer programming problem, and the problem

was solved by the proposed quantized DP algorithm. In [17], Wu et al. studied delay minimization for partial task offloading in the non-orthogonal multiple access (NOMA)-based MEC system. In [18], Wu et al. investigated multi-task multi-access MEC via NOMA to minimize the total energy consumption of entire tasks. They applied a two-step approach to studying the formulated problem. In [19], Tuong et al. investigated task offloading for the reduction of the average task delay by proposing a novel scheme, which can effectively improve the edge cloud service quality for all IoTDs by jointly optimizing the computation resource allocation (CRA) and sub-channel assignment (SA).

The minimization of both energy consumption and delay. In [6], a game-theoretic approach for mobile users is proposed to make task-offloading decisions by considering the overheads by the measurement of energy consumption and processing time in the multi-user MEC. In [20], Qian et al. investigated the joint resource allocation of communication and computation to minimize the delay and consumed energy by computing the resource allocation of edge servers in the NOMA-enabled multiaccess MEC with different edge servers. In [21], Zhang et al. studied data compression and the control speed of wireless transmission, aiming to minimize the energy consumption for wearable devices. An approximation algorithm was proposed such that the compression and transmission energy of wearable devices are minimized. Tang et al. studied partial task offloading in the MEC considering the overheads of time and energy [22]. The block coordinate descent method was proposed to solve the time and energy minimization problem. In [23], Feng et al. studied the trade-off of energy consumption and delay between the MEC system and the blockchain system by allocating the radio and communication resources in the blocked-enabled MEC systems. In [24], the authors focused on the study of sub-task offloading for minimizing both task-processing latency and energy consumption of the IoT devices with guarantees of dependency. They proposed an offloading scheme and compared it with different algorithms. However, this work did not consider resource allocation. In [25], Zhan et al. studied minimizing energy consumption and time for UAV applications in the MEC system enabled by UAV.

The works above did not consider security, an important issue when offloading tasks in MEC. To overcome this shortcoming, we present an efficient resource-allocation scheme for IoT task offloading in the MEC system by taking data security into account. Although some works considered the security issue of task offloading in the MEC system, many of them either overlooked the allocation of computing resources of the MEC server or the adoption of DVS without considering the joint study. In [11], Wu et al. studied power allocation in NOMA-enabled MEC networks aiming to minimize the energy consumption of two users while ensuring the task-offloading security. In [26], Elgendy et al. studied efficient and secure computation offloading for MEC-based mobile IoT networks minimizing the total energy consumed by the MDs. Although this security issue is considered in the two works, the work [11] did not take the allocation of computation resources into account, and the work [26] did not apply the DVS technology. Ref. [7] studied partial task offloading and the allocation of transmit power of MDs adopting the DVS technology. Two system-design goals, namely, minimizing energy consumption and task execution latency, were considered. However, allocating the computation resources of the MEC server and the security issue were not considered.

In order to present the existing challenges that this paper tries to address, we provide and summarize the limitations in the state-of-art literature in Table 1.

Table 1. Comparison.

Reference	DVS	Processing Time	Consumed Energy	Computation Resources	Security
[14]	No	Yes	No	No	No
[15]	No	No	Yes	No	No
[16]	No	No	Yes	Yes	No
[17]	No	Yes	No	No	No
[18]	No	No	Yes	No	No
[19]	No	Yes	No	No	No
[6]	No	Yes	Yes	No	No
[20]	No	Yes	Yes	No	No
[21]	No	No	Yes	No	No
[22]	No	Yes	Yes	No	No
[24]	No	Yes	Yes	No	No
[25]	No	Yes	Yes	No	No
[11]	No	No	Yes	No	Yes
[26]	No	Yes	Yes	Yes	Yes
[7]	Yes	Yes	Yes	No	No
This Study	Yes	Yes	Yes	Yes	Yes

3. System Model

In this section, the system model is introduced, and the optimization problem is then formulated. Suppose that an MEC system consisting of multiple IoTD users and an edge cloud integrated with one server providing computing resources for the number of N IoTDs, which is illustrated in Figure 2. The IoTDs can be the fire alarms, which are delay-sensitive. By analyzing the sensed data, these fire alarms will alarm if there are fire accidents. There is a base station (BS) where the edge cloud is deployed, through which each IoTD's task can be offloaded and processed in the MEC server. The task of IoTD i is denoted by $Q_i = (D_i, C_i)$ [27–29], where D_i denotes the task's data size, i.e., the input data, and C_i denotes the CPU cycles that should be needed to compute one bit of task data size. Let a_i denote the IoTD i's task offloading policy. If the user of IoTD i processes its task in the edge cloud, $a_i = 1$, otherwise, $a_i = 0$.

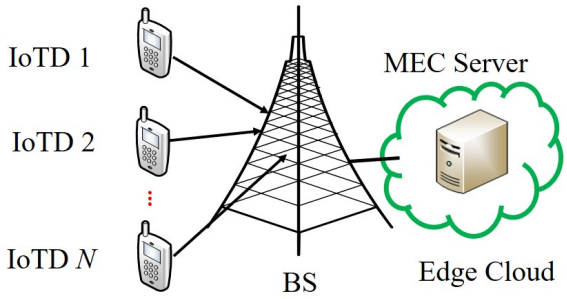

Figure 2. System model.

3.1. Local Model

In the local model, each IoTD user will execute the task by adopting the DVS technology to use the computing resources of its device. For the IoTD i, its executing time and the consumed energy for the local execution are, respectively, expressed as [6,26]:

$$T_{i,l} = \frac{C_i D_i}{f_i} \tag{1}$$

$$E_{i,l} = P_i C_i D_i, \tag{2}$$

where f_i represents the computing capability denoted by the number of CPU cycles that can be provided in one second, and P_i means the one CPU cycle power consumption.

The power consumed by one CPU cycle of the IoTD i is [6]

$$P_i = k_i f_i^2, \tag{3}$$

where k_i is a constant value parameter which is dependent on the chip architecture, and its value is 10^{-26} [30].

Therefore, we can obtain the computation overhead of the local model,

$$\begin{aligned} G_i &= w_{i,t} T_{i,l} + w_{i,e} E_{i,l} \\ &= w_{i,t} \frac{C_i D_i}{f_i} + w_{i,e} P_i C_i, D_i \end{aligned} \tag{4}$$

where $w_{i,t}$ and $w_{i,e} \in [0, 1]$ represent the weighting parameters of task executing time and the energy consumed by the IoTD i, respectively. If $w_{i,t} > w_{i,e}$, the IoTD i cares more about execution time. Otherwise, if $w_{i,t} < w_{i,e}$, then the IoTD i focuses more on its energy consumption.

3.2. Edge Cloud Model

In the edge cloud model, each IoTD will offload its task to the MEC server in the edge cloud through a wireless channel. When offloading tasks to the edge cloud, extra transmission time and energy consumption will be caused. For the IoTD i, its data rate in the uplink is

$$r_i = B_i \log_2(1 + \frac{p_i h_i}{B_i \varpi_0}), \tag{5}$$

where B_i is the bandwidth allocated to IoTD i, p_i is the transmission power, h_i means the channel gain expressed by d_i^{-2}, d_i denotes the distance from the IoTD i to the BS, and ϖ_0 denotes the background noise power.

Then, the transmission time and the energy consumed for transmission are, respectively, computed as [7,26],

$$T_{i,t} = \frac{D_i}{r_i} \tag{6}$$

$$E_{i,t} = p_i T_{i,t} = p_i \frac{D_i}{r_i} \tag{7}$$

When executed by the edge server, the computational time is calculated as

$$T_{i,e} = \frac{C_i D_i}{F_{i,e}}, \tag{8}$$

where $F_{i,e}$ is the allocated computing resources to the IoTD i.

3.3. Security Model

For the IoTD, i, the data of its task may be attacked by varieties of threats when it adopts the edge cloud model. Consequently, the AES cryptographic technique is adopted to protect the task against different types of threats [12,31,32]. In the security model, the tasks of IoTDs and their data will be encrypted before offloading. After receiving the data of these tasks, these data will be decrypted and executed by the edge servers. After accomplishing the execution of these tasks, the results will be sent back to the IoTDs.

Therefore, the additional time, including the encrypt time and decrypt time and energy consumption for securing data, are denoted, respectively, as [12,31,32],

$$T_{i,en} = \frac{C_i^e}{f_i} \tag{9}$$

$$T_{i,de} = \frac{C_i^d}{F_{i,e}} \tag{10}$$

$$T_{i,s} = T_{i,de} + T_{i,en} = \frac{C_i^e}{f_i} + \frac{C_i^d}{F_{i,e}} \tag{11}$$

$$E_{i,s} = k_i f_i^2 C_i^e \tag{12}$$

where C_i^e and C_i^d indicate the number of CPU cycles that are needed to encrypt and decrypt the task of IoTD i's data, respectively.

Therefore, the total computation overhead of the edge model is calculated as

$$\begin{aligned} H_i &= w_{i,t}(T_{i,t} + T_{i,e} + T_{i,s}) + w_{i,e}(E_{i,e} + E_{i,s}) \\ &= w_{i,t}(\frac{D_i}{r_i} + \frac{C_i D_i}{F_{i,e}} + \frac{C_i^e}{f_i} + \frac{C_i^d}{F_{i,e}}) \\ &+ w_{i,e}(\frac{p_i D_i}{r_i} + k_i f_i^2 C_i^e). \end{aligned} \tag{13}$$

For the analysis of convenience, the notations that are used in this paper are summarized in Abbreviations.

3.4. Problem Formulation

The task offloading and resource-allocation problem while considering data security is described in the following problem. The objective of it is to minimize the task-processing time and energy consumption of all IoTDs.

For the IoTD i, we let Z_i denote the execution time and energy consumption,

$$\begin{aligned} Z_i &= (1 - a_i)G_i + a_i H_i \\ &= (1 - a_i)[w_{i,t}\frac{C_i D_i}{f_i} + w_{i,e} k_i f_i^2 C_i D_i] + \\ & a_i[w_{i,t}(\frac{D_i}{r_i} + \frac{C_i D_i}{F_{i,e}} + \frac{C_i^e}{f_i} + \frac{C_i^d}{F_{i,e}}) \\ & + w_{i,e}(\frac{p_i D_i}{r_i} + k_i f_i^2 C_i^e)]. \end{aligned} \tag{14}$$

Hence, the formulated problem is expressed as,

Problem 1.

$$\begin{aligned} &\min_{f_i, F_{i,e}, a_i} \sum_{i=1}^{N} Z_i \\ &s.t. \quad 0 < f_i \le f_i^m \\ &\sum_{i=1}^{N} F_{i,e} \le F \\ &a_i \in \{0, 1\}, \end{aligned} \tag{15}$$

where the first constraint is the maximum computational capability of IoTD i, the second constraint means that the allocated resources to the IoTDs should not be more than the computation capability of the MEC server, and a_i is the task offloading policy.

Remark 1. *As the task-offloading policy is binary, and the resource allocation is continuous in Problem* ***P****, the above problem is a non-convex MINP problem, and it is notoriously challenging to find the optimal solution. Traditionally, the MINP problem can be solved by leveraging the Branch-and-Bound, the Dinkelbach, and the Alternating Direction Method of Multipliers (ADMM). However, the time complexity of applying these algorithms is prohibitive [16]. The authors in [33] obtain the solution of the MINP problem using these traditional algorithms and present the performances of these algorithms. To solve the task-offloading problem, Chen et al. presented a distributed learning-based framework [34]. In the following section, we will offer an efficient algorithm to obtain the optimal solution to Problem* ***P****.*

4. Algorithm Design

It is obvious that Problem 1 is MINLP, which is difficult to solve [7,35,36]. In this part, we put forward an efficient algorithm to obtain the solution to Problem 1. Before that, a lemma is firstly presented whose proof is in [35], according to which the solution is obtained.

Lemma 1. *We can always obtain*

$$\sup_{x,y} f(x,y) = \sup_{x} \tilde{f}(x),$$

where $\tilde{f}(x)$=$sup_y f(x,y)$.

By referring to Lemma 1, we have the conclusion that we could always solve a function from minimizing over some of the variables firstly, and minimize over the left ones later.

From Lemma 1, Problem 1 can be solved by minimizing p_i, F_i, and a_i sequentially. That is, we can first optimize the transmission power of IoTDs and computing resources of the edge server provided that the offloading decisions of IoTDs are given. Consequently,Problem 1 can be solved by dividing it into the following sub-problems:

(1) Local model computation problem;
(2) Edge model computing problem;
(3) Task offloading decision problem.

4.1. Local Model Computation Problem

Based on Problem 1, when $a_i = 0$, the IoTD users execute tasks on their own devices. Hence, the local model computing problem is written as follows,

Problem 2.

$$\begin{aligned} &\min_{f_i} G_i(f_i) \\ &s.t. \quad 0 < f_i \leq f_{i,m}, \end{aligned} \tag{16}$$

where $G_i(f_i)$ *is denoted as*

$$\begin{aligned} G_i &= w_{i,t} T_{i,l} + w_{i,e} E_{i,l} \\ &= w_{i,t} \frac{C_i D_i}{f_i} + w_{i,e} k_i f_i^2 C_i D_i \end{aligned} \tag{17}$$

From the second-order derivative of Equation (17),

$$\frac{\partial^2 G_i}{\partial {f_i}^2} = \frac{2 w_{i,t} C_i D_i}{f_i^3} + 2 C_i D_i w_{i,e}, \tag{18}$$

we know that G_i'' is positive in the domain of f_i; therefore, f_i is convex. Therefore, from the first derivative, we have

$$\frac{\partial G_i}{\partial f_i} = -\frac{w_{i,t} C_i D_i}{(f_i)^2} + 2 w_{i,e} k_i f_i C_i D_i = 0. \tag{19}$$

From Equation (19), the optimal solution is

$$f_i^* = \sqrt[3]{\frac{w_{i,t}}{2 k_i w_{i,e}}}. \tag{20}$$

It can be clearly observed that $G_i(f_i)$ monotonously increases if $f_i > f_i^*$ and monotonously increases when $f_i < f_i^*$.

Hence, the optimal solution to the local computation problem P2 is

$$G_i(f_i) = \begin{cases} G_i(f_{i,m}), f_i^* \geq f_{i,m} \\ G_i(f_i), f_i^* < f_{i,m} \end{cases} \tag{21}$$

4.2. Edge Model Computing Problem

According to Problem 1, when $a_i = 1$, the users of the IoTDs will execute their tasks in the edge cloud. Therefore, the allocation of the MEC server computing problem can be written as follows.

Problem 3.

$$\begin{aligned} &\min_{f_i, F_{i,e}} \sum_{i=1}^{N} H_i(f_i, F_{i,e}) \\ &s.t. \quad 0 < f_i \leq f_{i,m} \\ &\quad \sum_{i=1}^{N} F_{i,e} \leq F \\ &\quad F_{i,e} > 0, \end{aligned} \tag{22}$$

where $H_i(f_i, F_{i,e})$ *is denoted as*

$$\begin{aligned} H_i(f_i, F_{i,e}) = w_{i,t}\left(\frac{D_i}{r_i} + \frac{C_i D_i}{F_{i,e}} + \frac{C_i^e}{f_i} + \frac{C_i^d}{F_{i,e}}\right) \\ + w_{i,e}\left(\frac{p_i D_i}{r_i} + k_i f_i^2 C_i^e\right). \end{aligned} \tag{23}$$

By checking the objective function of Problem 3, we can further divide the above problem into the following two sub-problems: (1) Local computing resource-allocation problem; and (2) Edge cloud computing resource-allocation problem.

The problem of resource allocation in local model computing is formulated as follows.

Problem 4.

$$\begin{aligned} &\min_{f_i, F_{i,e}} \sum_{i=1}^{N} H_i(f_i) \\ &s.t. \quad 0 < f_i \leq f_{i,m}, \end{aligned} \tag{24}$$

where $H_i(f_i)$ *is denoted as*

$$H_i(f_i) = w_{i,t} \frac{C_i^e}{f_i} + w_{i,e} k_i f_i^2 C_i^e. \tag{25}$$

As function $H_i(f_i)$ has a similar structure to $G_i(f_i)$, the solution of function $H_i(f_i)$ can easily be obtained, which is expressed as

$$H_i(f_i) = \begin{cases} H_i(f_{i,m}), f_i^* \geq f_{i,m} \\ H_i(f_i), f_i^* < f_{i,m}, \end{cases} \tag{26}$$

where f_i^* is given as follows

$$f_i^* = \sqrt[3]{\frac{w_{i,t}}{2k_i w_{i,e}}}. \tag{27}$$

The edge cloud resource-allocation problem is as follows.

Problem 5.

$$\begin{aligned} &\min_{F_{i,e}} \sum_{i=1}^{N} H_i(F_{i,e}) \\ &s.t. \sum_{i=1}^{N} F_{i,e} \leq F_e \\ &F_{i,e} > 0, \end{aligned} \tag{28}$$

where $H_i(F_{i,e})$ *is denoted as*

$$H_i(F_{i,e}) = w_{i,t}(\frac{C_i D_i + C_i^d}{F_{i,e}}). \tag{29}$$

From

$$\frac{\partial^2 H_i}{\partial {F_{i,e}}^2} = \frac{2w_{i,t}(C_i D_i + C_i^d)}{F_{i,e}^3} > 0, \tag{30}$$

we know that H_i is convex [35]. Consequently, the Lagrangian function is

$$\begin{aligned} L(F_{i,e}, u) = \sum_{i=1}^{N} [&\frac{w_{i,t}(C_i + C_i^d)}{F_{i,e}} \\ &+ u(\sum_{i=1}^{N} F_{i,e} - F_e), \end{aligned} \tag{31}$$

where $u \geq 0$ is the Lagrange multiplier. Referring to the KKT conditions [35], we have

$$\frac{\partial L}{\partial F_{i,e}} = \frac{-w_{i,t}(C_i D_i + C_i^d)}{F_{i,e}^2} + u = 0 \tag{32}$$

$$u(\sum_{i=1}^{N} F_{i,e} - F_e) = 0 \tag{33}$$

$$u \geq 0. \tag{34}$$

According to Equation (33), it is obvious that $u > 0$. Hence, from Equation (32), we have

$$F_{i,e}^* = \sqrt{\frac{w_{i,t}(C_i D_i + C_i^d)}{u}}. \tag{35}$$

Substituting Equation (37) back into the Equation (33), we can obtain the optimal value of Lagrangian multiplier, which is

$$u = (\frac{\sum_{i=1}^{N} \sqrt{w_{i,t}(C_i D_i + C_i^d)}}{F_e})^2. \tag{36}$$

Therefore, by substituting Equation (36) into Equation (37), the optimal value of the allocated computing resource is

$$F_{i,e}^{*} = \frac{\sqrt{w_{i,t}(C_i D_i + C_i^d)}}{\sum_{i=1}^{N} \sqrt{w_{i,t}(C_i D_i + C_i^d)}} F_e. \tag{37}$$

By substituting Equation (37) into Equation (25), the optimal value of $H_i(F_{i,e})$ is

$$H_i^{*}(F_{i,e}) = \frac{[\sum_{i=1}^{N} \sqrt{w_{i,t}(C_i D_i + C_i^d)}]^2}{F_e}. \tag{38}$$

4.3. Offloading Decision Problem

In this subsection, an efficient resource-allocation and task-offloading algorithm in the security-aware MEC system is presented. The users of IoTDs decide to execute their tasks in the MEC server if the computation overheads are smaller than the computation overheads of the local model [6]. For each user of IoTD, whether offloading its task or not is set by making a comparison of the computation overhead of the local model and edge model.

$$o_i = \begin{cases} 1, G_i \geq H_i \\ 0, G_i < H_i \end{cases} \tag{39}$$

After obtaining the minimum computation cost of each IoTD, we can obtain the optimal system computation cost, which is

$$Z^{*} = \sum_{i=1}^{N} ((1 - a_i) G_i^{*} + a_i H_i^{*}). \tag{40}$$

An algorithm is presented to obtain an optimal solution for the offloading decision policies, which is shown in Algorithm 1. This algorithm provides a detailed process to obtain the optimal decision for each IoTD user. In Algorithm 1, all users of IoTDs firstly initialize their task-offloading policies as 0 to obtain their optimal solution to the CPU frequency. Then, all users of IoTDs set the task-offloading policies as 1 to reach their optimal solution to the allocated computing resources and the local CPU frequency. At last, all IoTD users make the optimal offloading policies by comparing the computation overheads of the two models.

4.4. Complexity Analysis

In the local model, assume that there are n IoTDs, the computation overhead needs n times computing. Therefore, the time complexity of steps 3–6 in Algorithm 1 is $O(n)$. In the edge model considering the security issue, the computation overhead needs $N - n$ times computing. Therefore, the time complexity of steps 7–8 in Algorithm 1 is $O(N - n)$. Consequently, the time complexity of Algorithm 1 is $O(N)$. In [33], Yu et al. studied power resource allocation and interference management in Fog computing by applying the combination of the traditional methods, namely, ADMM, benders decomposition, and the Dinkelbach algorithm. The time complexity of this work is $O(max\{N/(\varepsilon^2), 2^N\})$, where ε denotes the tolerance.

Algorithm 1 The Proposed Algorithm in the Security-Aware MEC System

Input:
N tasks of IoTDs;
Output:
The resources allocation, task offloading, and the system computation overhead;
Initialize $i \leftarrow 1$
repeat
$f_i* \leftarrow$ Obtain its computational resource allocation by solving subproblem **P2**;
$G_i \leftarrow$ Obtain its computation overhead in the local computing model based on Equation (21);
$F_{i,e}* \leftarrow$ Obtain its optimal allocation of computational resources by solving problem **P3**;
$H_i \leftarrow$ Obtain its the optimal computation overhead in the MEC server model from Equation (13);
if $G_i \geq H_i$ **then**
    $a_i \leftarrow 1$;
else
    $a_i \leftarrow 0$;
end if
$i \leftarrow i + 1$
until $i \leftarrow N$

5. Experimental Results

In this section, the experimental results are provided to demonstrate the performance of the proposed model. The proposed efficient resource allocation for the security-aware task-offloading model labeled as Secured Model is evaluated through simulation results by comparing with the following models.

Local Model: In this model, all the IoTDs' tasks do not offload and are executed using the resources of IoTDs.

Edge Model: In this model, all IoTDs will take the offloading strategies. Their tasks are all processed by the MEC server.

Unsecured Model: In this model, the IoTDs' tasks may be computed in the local model or offloaded to the MEC server according to the proposed offloading algorithms. This kind of model is widely studied in existing works, such as [6,37].

5.1. Simulation Parameter Settings

Assuming that some IoTDs and one server exist in a MEC system, the default number of the IoTDs is $N = 10$ unless otherwise specified. The distance from the IoTD i to the BS is 1000 m. The computational capability of the MEC server is $F = 10$ GHz, and the computational capability of each IoTD is in [0, 1] GHz. Each IoTD has a task processed by itself or in the MEC server. The IoTDs' tasks can be the face-recognition application, whose data size is uniformly distributed in [0, 10] Mb. The needed CPU cycles to complete one bit are set to 1000 cycles. As far as the IoTD i is concerned, its transmission power is set to 0.1 W, the bandwidth is 0.3 MHz, its channel gain h_i is 2.6×10^{-7}, and the background noise power is 10^{-7} W. The weighting parameter values $w_{i,t}$ and $w_{i,e}$ are set from $\{0.2, 0.5, 0.8\}$ and $w_{i,t} + w_{i,e} = 1$. Moreover, the number of CPU cycles to encrypt and decrypt the IoTD i's data are both set to 100 megacycles. A summary of the values of the main parameters is given in Table 2. We set these parameter values according to [6,12,16,38].

Table 2. Parameter values Settings.

Parameters	Values
The IoTDs' number N	10
The allocated bandwidth of IoTD i B_i	0.3 MHz
The IoTD is' data size D_i	[0, 10] Mb
The transmission power p_i	0.1 W
The needed CPU cycles to process one bit of data size C_i	1000 cycles
The number of CPU cycles to encrypt the IoTD i's data C_i^e	100 magacycles
The number of CPU cycles to decrypt the IoTD i's data C_i^d	100 magacycles
k_i	10^{-26}
The channel gain h_i	10^{-6}
The maximum computational capability $f_{i,m}$	1 GHz
The background noise power ϖ_0	10^{-7} W
The computational capability for the MEC server F	30 GHz

5.2. Simulation Results

Figure 3 shows the offloading percentage of IoTD users under different numbers of IoTD users considering the security model and the one without security. It is evident that the offloading percentage is lower when the security methodology is adopted. The offloading percentage of the two models decreases, respectively, versus the increasing number of IoTD users. For the secured model, the offloading percentage decreases more rapidly than the unsecured model. Moreover, we observe that the offloading percentage of IoTD users with the unsecured model remains at 100 percent when the number is less than 10.

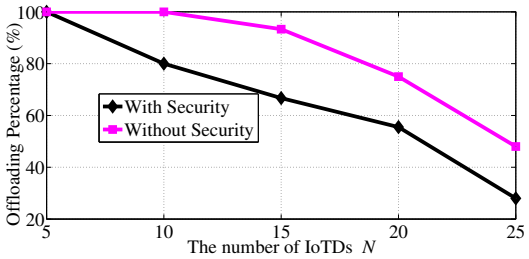

Figure 3. The offloading percentage of IoTD users.

Figure 4 shows the energy consumption of all tasks versus different numbers of IoTDs. Specifically, Figure 4a makes a comparison of the energy consumption under the models of Local Model, Edge Model, and the Unsecured Model, and Figure 4b also shows the consumed energy in the Secured model. It is evident from the two figures that processing tasks of IoTDs under the Local Model consumes the most energy of the four models. In contrast, less energy is consumed in the Edge Model than the other three models. It can also be observed that energy consumption significantly increases when the number of IoTDs rises. Furthermore, we can notice that, when the number of IoTD users is less than 15, the energy consumption under the Secured model and the proposed Unsecured model are both near to the Edge model. However, when the number of IoTD users exceeds 15, the gap will become larger.

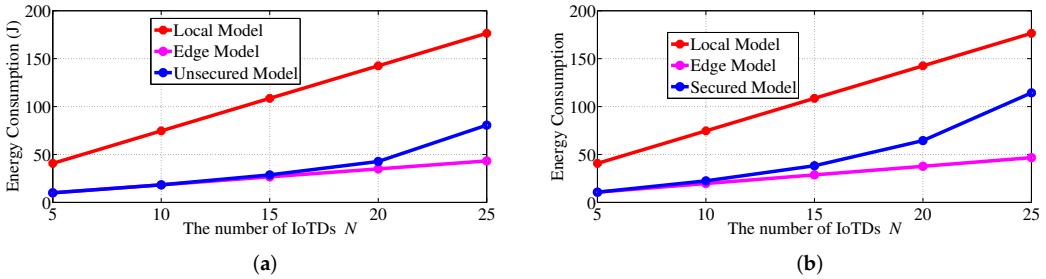

Figure 4. The energy consumption. (**a**) Unsecured model; (**b**) Secured model.

Figure 5 depicts the task-execution time of all the IoTDs versus different numbers of IoTDs under different models. In Figure 5a, a comparison among the models of Local Model, Edge Model, and the Unsecured Model is shown, while Figure 5b shows the task-execution time of the Secured model. By observing Figure 5, it is found that the task-execution time in the Local Model is the least among the four models, while processing in the Edge Model costs more time than the other three models. From the two sub-figures, we see that the total task execution time under the four models, respectively, increases if the number of IoTD users increases. By comparing Figures 4 and 5, we find that the proposed Secured Model consumes more energy than the baseline model Unsecured Model, while costing less time. In addition, it can be noticed that, when the number of IoTD users is less than 20, the task-execution time under the Secured model and the proposed Unsecured model are both near to the Edge model. However, when the number of IoTD users exceeds 20, the gap between the two models will become larger.

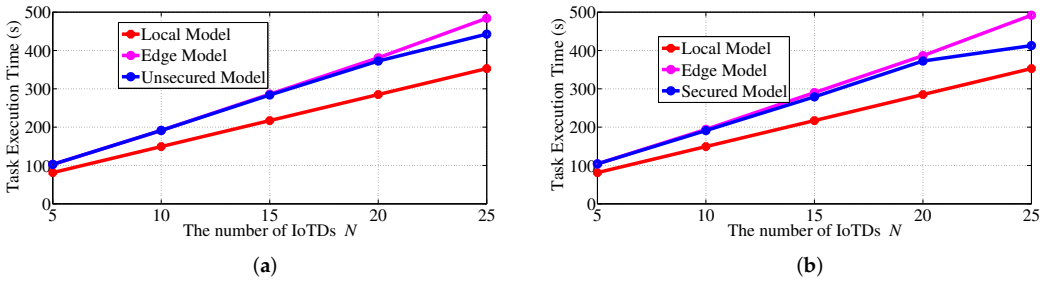

Figure 5. The task execution time. (**a**) Unsecured model; (**b**) Secured model.

Figure 6a,b show the total computation overheads under the Unsecured model and Secured model, respectively. It can be noticed from Figure 6a that the computation overheads under the Unsecured model are higher than under the Local Model and the Edge Model. It can be noticed from Figure 6b that the computation overheads under the proposed secured model are higher than under the Local Model and the Edge Model. By comparing Figure 6a and Figure 6b, we observe that the proposed secured model obtains higher computation overheads than the Unsecured model. However, the gap is not obvious. This may be the reason that encrypting and decrypting data of the tasks of IoTD users may cause extra computation overheads. In addition, it is noted that the computation overheads in the Local Model are higher than or near equal to those of the Edge Model when the number of IoTD users is less than 20. However, the computation overheads under the Edge Model will be higher than the Local Model when the number of IoTD users is over 20.

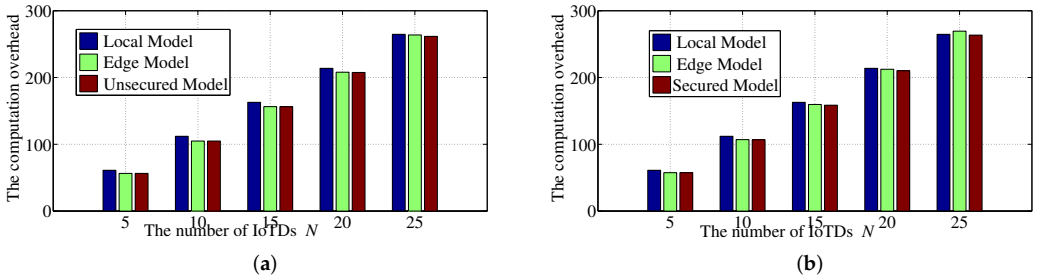

Figure 6. The computation overhead. (**a**) Unsecured model; (**b**) Secured model.

In the last section, we analyze the impact of the computational capacity of the MEC server on the offloading percentage of IoTD users under the unsecured model and the secured model. We set the number of IoTD users N as 15 and vary the computational capability of the MEC server from 10 GHz to 30 GHz. We present the corresponding results in Figure 7. It can obviously be found from Figure 7 that the offloading percentage of IoTD users will increase with the computational capabilities of MEC servers increasing. Moreover, the offloading percentage under the secured model is lower than that under the unsecured model. This is because taking the secure technique will cost additional processing time and energy.

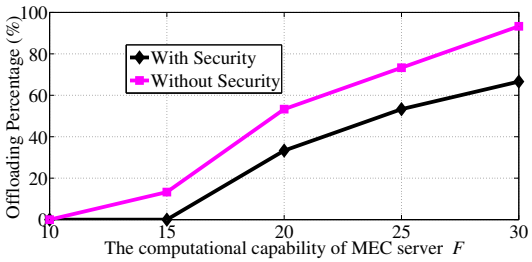

Figure 7. The offloading percentage of IoTD users versus different computational capacities of MEC server.

5.3. Discussions

In this paper, only the binary task offloading, that is, the task of IoTD that is either processed in the model or in the edge model by offloading, is considered. In our proposed Secured Model, as far as each IoTD is considered, since the proposed algorithm selects the smaller computation overhead, the system computation overhead is the lowest compared with the Local Model and the Edge Model. However, compared with the Unsecured Model, as the adoption of the secure technique costs additional processing time and energy, the Secured Model will cost additional computation overheads. This can be extended to the partial task-offloading case, namely, one part of the task for each IoTD is locally processed, and the other part of this task is offloaded.

According to the experimental results and the results' analysis, the proposed secured model can be applied to some IoT applications, such as face-recognition applications. However, for applications that need higher requirements for the delay, such as traffic safety applications, the edge cloud owner has to provide more resources to meet the applications of the requirements.

As far as the formulated MINP problem is concerned, as well as the proposed algorithm, other methods such as the dandelion algorithm (DA) [39] or the Bat algorithm [40], may be applied to obtain the optimal solution. Queuing models can be used to model the system model [41]. Besides, task Offloading in the air-ground integrated MEC Systems also needs to be further studied [42].

6. Conclusions

In this paper, the IoT task-offloading problem with security taking into account an MEC system is studied, whose objective is minimizing the task-executing time and energy consumption of IoTDs by allocating the computing resources while adopting the DVS technology. Meanwhile, to ensure the security the offloaded tasks, we adopt the AES cryptographic technique to secure the data of these tasks. An optimization problem considering the computation overheads of the IoTDs is formulated. For the optimization problem, since its objective function is not convex, the computation-overhead minimization problem is a MINP problem, and it is well-known as NP-hard. To obtain the solution to this problem, we first split it into three layered sub-problems and obtain the answer to each. The experimental simulation results are validated to show the effectiveness of the proposed task-offloading model. The experimental simulation results reveal the fact that the number of the IoTDs has an evident effect on the energy consumption, task-execution time, and offloading percentage of IoTD users. In a comparison to baseline models, the proposed secured model reveals its better performances.

Author Contributions: Conceptualization, Y.W., W.Z., H.D. and X.L.; methodology, Y.W., W.Z., H.D. and X.L.; software, W.Z. and H.D.; validation, Y.W. and H.D.; formal analysis, Y.W. and X.L.; investigation, Y.W. and X.L.; resources, Y.W. and H.D.; data curation, H.D. and X.L.; writing—original draft preparation, Y.W. and W.Z.; writing—review and editing, Y.W. and X.L.; visualization, Y.W., H.D. and X.L.; supervision, X.L.; project administration, X.L.; funding acquisition, Y.W., W.Z. and X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the soft science research program of Shaanxi Provincial Department of science and technology (2022KRM098), the Open project support of Anhui Provincial Key Laboratory of intelligent building and building energy efficiency in Anhui Jianzhu University (IBES2020KF03), the Shaanxi provincial plan for improving public scientific quality (2021PSL14), and the Teaching research project of computer basic education (2021-AFCEC-435).

Data Availability Statement: The data of this paper will be available by contacting the corresponding author.

Acknowledgments: The authors have benefited a lot from the comments of academic editor and reviewers, which have improved the quality of this paper significantly.

Conflicts of Interest: The authors of this paper declare that they have no conflict of interest.

Abbreviations

We use following abbreviations in this paper:

MEC	mobile edge computing
IoT	Internet of Things
IoT devices	IoTDs
DVS	Dynamic voltage scaling
MINP	mixed-integer nonlinear programming
Notations	
N	the IoTDs' number
f_i	the computation capability of the IoTD i
D_i	the data size of IoTD i's task in bits
C_i	the amount of CPU cycles needed to compute one bit of the data size of IoTD i
C_i^e	the number of CPU cycles needed to encrypt the data of the task of IoTD i
C_i^d	the number of CPU cycles needed to decrypt the data of the application of IoTD i
$w_{i,t}$	represent the weighting parameter of the execution time of the IoTD i
$w_{i,e}$	represent the weighting parameter the energy consumed by the IoTD i
λ	the lagrange multiplier
B_i	the uplink channel bandwidth of the IoTD i
p_i	the power transmission of IoTD i
h_i	the channel gain from the IoT device i to the BS

ϖ_0	the noise power
r_i	the achievable uplink rate for the IoTD i
a_i	the task offloading decision made by the IoTD i
F	the computation capability of the MEC server
$F_{i,e}$	the computation resources of the MEC server allocated to the IoTD i

References

1. Tan, L.; Yu, K.; Bashir, A.K.; Cheng, X.; Ming, F.; Zhao, L.; Zhou, X. Toward real-time and efficient cardiovascular monitoring for COVID-19 patients by 5G-enabled wearable medical devices: A deep learning approach. *Neural Comput. Appl.* **2021**, *2*, 1–14. [CrossRef]
2. Lueth, K.L. State of the IoT 2018: Number of IoT devices now at 7B Market accelerating. Available online: https://iot-analytics.com/ (accessed on 14 August 2022)
3. Chiang, M.; Zhang, T. Multiobjective optimization for computation offloading in fog computing. *IEEE Internet Things* **2018**, *3*, 854–864. [CrossRef]
4. Premsankar, M.; Francesco, M. D.; Taleb, T. Edge computing for the internet of things: A case study. *IEEE Internet Things* **2018**, *5*, 1275–1284. [CrossRef]
5. Deng, S.; Huang, L.; Taheri, J.; Zomaya, A.Y. Computation offloading for service workflow in mobile cloud computing. *IEEE T. Parall. Distr.* **2015**, *26*, 3317–3329. [CrossRef]
6. Chen, X.; Jiao, L.; Li, W.; Fu, X. Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Trans. Netw.* **2016**, *24*, 2795–2808. [CrossRef]
7. Wang, Y.; Sheng, M.; Wang, X.; Wang, L.; Li, J. Mobile-edge computing: Partial computation offloading using dynamic voltage scaling. *IEEE Trans. Commun.* **2016**, *10*, 4268–4282. [CrossRef]
8. Fang, W.; Ding, S.; Li, Y.; Zhou, W.; Xiong, N. Okra: Optimal task and resource allocation for energy minimization in mobile edge computing systems. *Wirel. Netw.* **2019**, *25*, 2851–2867. [CrossRef]
9. Chen, W.; Wang, D.; Li, K. Multiuser multi-task computation offloading in green mobile edge cloud computing. *IEEE Trans. Serv. Comput.* **2019**, *12*, 726–738. [CrossRef]
10. Liu, D.; Ding, W.; Atiquzzaman, M. A survey on secure data analytics in edge computing. *IEEE Internet Things* **2019**, *6*, 4946–4967. [CrossRef]
11. Hou, X.; Li, Y.; Liu, P. Energy-Efficient Edge Computing Service Provisioning for Vehicular Networks: A Consensus ADMM Approachs. *IEEE Trans. Veh. Technol.* **2019**, *68*, 5087–5099.
12. Elgendy, I.A.; Zhang, W.; Tian, Y.C.; Li, K. Resource allocation and computation offloading with data security for mobile edge computing. *Future Gener. Comput. Syst.* **2019**, *100*, 531–541. [CrossRef]
13. Tran, T.X.; Pompili, D. Joint task offloading and resource allocation for multi-server mobile-edge computing networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 856–868. [CrossRef]
14. Gao, M.; Shen, R.; Li, J.; Yan, S.; Li, Y.; Shi, J.; Han, Z. Computation Offloading With Instantaneous Load Billing for Mobile Edge Computing. *IEEE Trans. Veh. Technol.* **2022**, *15*, 1473–1485. [CrossRef]
15. Han, D.; Chen, W.; Fang, Y. Joint channel and queue aware scheduling for latency sensitive mobile edge computing with power constraints. *IEEE Wirel. Commun.* **2020**, *19*, 3938–3951. [CrossRef]
16. Lyu, X.; Tian, H.; Ni, W.; Zhang, Y.; Zhang, P.; Liu, R. Energy-Efficient Admission of Delay-Sensitive Tasks for Mobile Edge Computing. *IEEE Trans. Commun.* **2018**, *66*, 2603–2616. [CrossRef]
17. Wu, Y.; Qian, L.; Ni, K.; Zhang, C.; Shen, X. Delay-Minimization Nonorthogonal Multiple Access Enabled Multi-User Mobile Edge Computation Offloading. *IEEE J. Sel. Top. Signal Process.* **2019**, *13*, 392–407. [CrossRef]
18. Wu, Y.; Shi, B.; Qian, L.; Hou, F.; Cai, J.; Shen, X. Energy-Efficient Multi-task Multi-access Computation Offloading Via NOMA Transmission for IoTs. *IEEE Trans. Ind. Inform.* **2020**, *16*, 4811–4822. [CrossRef]
19. Tuong, V. D.; Noh, W.; Sungrae, C. Delay Minimization for NOMA-Enabled Mobile Edge Computing in Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2022**, *18*, 7321–7331. [CrossRef]
20. Qian, L.; Shi, B.; Wu, Y.; Sun, B.; Tsang, D.H.K. NOMA-Enabled Mobile Edge Computing for Internet of Things via Joint Communication and Computation Resource Allocations. *IEEE Internet Things* **2020**, *7*, 718–733. [CrossRef]
21. Zhang, W.; Fan, R.; Wen, Y.; Liu, F. Energy Optimal Wireless Data Transmission for Wearable Devices: A Compression Approach. *IEEE Trans. Veh. Technol.* **2018**, *67*, 9605–9618. [CrossRef]
22. Tang, Q.; Lyu, H.; Han, G.; Wang, J.; Wang, K. Partial offloading strategy for mobile edge computing considering mixed overhead of time and energy. *Neural Comput.* **2020**, *32*, 15383–15397. [CrossRef]
23. Feng, J.; Yu, F.R.; Pei, Q.; Du, J.; Zhu, L. Joint Optimization of Radio and Computational Resources Allocation in Blockchain-Enabled Mobile Edge Computing Systems. *IEEE Wirel. Commun.* **2020**, *19*, 4321–4334. [CrossRef]
24. Zhang, Y.; Chen, J.; Zhou, Y.; Yang, L.; H, B.; Yang, Y. Dependent task offloading with energy-latency tradeoff in mobile edge computing. *IET Commun.* **2022**, 1–9. [CrossRef]

25. Zhan, C.; Hu, H.; Sui, X.; Liu, Z.; Niyato, D. Completion Time and Energy Optimization in the UAV-Enabled Mobile-Edge Computing System. *IEEE Internet Things* **2020**, *8*, 7808–7822. [CrossRef]
26. Elgendy, I.A.; Zhang, W.; Zeng, Y.; He, H.; Tian, Y.; Yang, Y. Efficient and Secure Multi-User Multi-Task Computation Offloading for Mobile-Edge Computing in Mobile IoT Networks. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 2410–2422. [CrossRef]
27. Siriwardhana, Y.; Porambage, P.; Liyanage, M.; Ylianttila, M. A Survey on Mobile Augmented Reality With 5G Mobile Edge Computing: Architectures, Applications, and Technical Aspects. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1160–1192. [CrossRef]
28. Shakarami, A.; Ghobaei-Arani, M.; Shahidinejad, A. A survey on the computation offloading approaches in mobile edge computing: A machine learning-based perspective. *Comput. Netw.* **2020**, *182*, 107496. [CrossRef]
29. Portilla, J.; Mujica, G.; Lee, J.S.; Riesgo, T. The extreme edge at the bottom of the Internet of Things: A review. *IEEE Sens. J.* **2019**, *19*, 3179–3190. [CrossRef]
30. Zhang, J.; Hu, X.; Ning, Z.; Ngai, E.C.-H.; Zhou, L.; Wei, J.; Cheng, J.; Hu, B. Energy-Latency Tradeoff for Energy-Aware Offloading in Mobile Edge Computing Networks. *IEEE Internet Things* **2018**, *5*, 2633–2645. [CrossRef]
31. Elgendy, I.A.; Zhang, W.; Liu, C.; Hsu, C. An Efficient and Secured Framework for Mobile Cloud Computing. *IEEE Trans. Cloud Comput.* **2021**, *9*, 79–87. [CrossRef]
32. Zhang, W.; Elgendy, I.A.; Hammad, M.; Iliyasu, A.M.; Du, X.; Guizani, M.; El-Latif, A.A.A. Secure and Optimized Load Balancing for Multitier IoT and Edge-Cloud Computing Systems. *IEEE Internet Thing* **2021**, *8*, 8119–8132. [CrossRef]
33. Yu, Y.; Bu, X.; Yang, K.; Wu, Z.; Han, Z. Green large-scale fog computing resource allocation using joint benders decomposition, dinkelbach algorithm, admm, and branch-and-bound. *Comput. IEEE Internet Things* **2019**, *6*, 4106–4117. [CrossRef]
34. Chen, X.; Wu, C.; Liu, Z.; Zhang, N.; Ji, Y. Computation Offloading in Beyond 5G Networks: A Distributed Learning Framework and Applications. *IEEE Wirel. Commun.* **2021**, *28*, 56–62. [CrossRef]
35. Boyd, S.; Vandenberghe, L. *Convex Optimization*; Computational Cambridge University Press: Cambridge, UK, 2004.
36. Lyu, X.; Tian, H.; Sengul, C.; Zhang, P. Multiuser Joint Task Offloading and Resource Optimization in Proximate Clouds. *IEEE Trans. Veh Technol.* **2017**, *66*, 3435–3447. [CrossRef]
37. Li, X.; Zhao, L.; Yu, K.; Aloqaily, M.; Jararweh, Y. A cooperative resource allocation model for IoT applications in mobile edge computing. *Comput. Commun.* **2021**, *173*, 183–191. [CrossRef]
38. Elgendy, I.A.; Muthanna, A.; Hammoudeh, M.; Shaiba, H.; Unal, D.; Khayyat, M. Advanced Deep Learning for Resource Allocation and Security Aware Data Offloading in Industrial Mobile Edge Computing. *Big Data* **2021**, *9*, 265–278. [CrossRef]
39. Li, X.-G.; Han, S.-F.; Zhao, L.; Gong, C.-Q.; Liu, X.-J. New dandelion algorithm optimizes extreme learning machine for biomedical classification problems. *Comput. Intell. Neurosci.* **2017**, *2017*, 4523754. [CrossRef]
40. Lin, N.; Tang, J.; Li, X.; Zhao, L. A novel improved bat algorithm in uav path planning, Computers. *Comput. Mater. Contin.* **2019**, *61*, 323–344.
41. Meng, T.; Wolter, K.; Wu, H.; Wang, Q. A secure and cost-efficient offloading policy for mobile cloud computing against timing attacks. *Pervasive Mob. Comput.* **2018**, *45*, 4–18. [CrossRef]
42. Chen, X.; Wu, C.; Chen, T.; Liu, Z.; Zhang, H.; Liu, H.; Bennis, M.; Ji, Y. Information Freshness-Aware Task Offloading in Air-Ground Integrated Edge Computing Systems. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 243–258. [CrossRef]