Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 9162422, 10 pages
https://doi.org/10.1155/2022/9162422

Research Article

Empirical Matching-Based Computation Offloading Optimization for 5G and Edge Computing-Integrated EIoT

Hui Zhang, Huixia Ding, Yang Wang, Sachula Meng, Sicheng Zhu, Ling Teng, and Fangyun Dong

China Electric Power Research Institute, Beijing, China

Correspondence should be addressed to Hui Zhang; zhanghui@epri.sgcc.com.cn

Received 28 January 2022; Revised 24 February 2022; Accepted 1 March 2022; Published 15 April 2022

Academic Editor: Han Wang

Electric Internet of things (EIoT) that integrates 5G and edge computing can provide data transmission and processing guarantee for smart grid. However, computation offloading optimization including joint optimization of server selection and computation resource allocation still faces several challenges such as difficulty in tradeoff balance among various quality of service (QoS) parameters, coupling between server selection and computation resource allocation, and multi-device competition. To address these challenges, we propose an empirical matching-based computation offloading optimization algorithm for 5G and edge computing-integrated EIoT. The optimization objective is to minimize the computation offloading delay by jointly optimizing large timescale server selection and small timescale computation resource allocation. We first model the large timescale server selection problem as a many-to-one matching problem, which can be decoupled from small timescale computation resource allocation by establishing a matching preference list based on empirical performance. Then, the large timescale server selection problem is solved by pricing-based matching with a quota algorithm. Furthermore, based on the obtained suboptimal result of large timescale server selection, the small timescale computation resource allocation problem is subsequently solved by Lagrange dual decomposition, the result of which is used to update large timescale empirical performance. Finally, extensive simulations are carried out to demonstrate the superior performance of the proposed algorithm by comparing it with existing algorithms.

1. Introduction

In order to achieve the energy supply-demand balance and the safe and stable operation of the smart grid, massive electric Internet of things (EIoT) devices need to be deployed to support multiple types of real-time data collection, such as voltage, current, active/reactive power, electric energy, temperature, and humidity [1–3]. The integration of 5G and edge computing provides a viable solution for the real-time data monitoring. Specifically, 5G provides communication guarantee for real-time data collection and information interaction due to the advantages of high reliability, wide connection, and low delay [4]. Compared with ad hoc networks, 5G with central control functionality can achieve centralized and efficient resource scheduling and management for computation offloading optimization in EIoT. Edge computing provides computation guarantee for real-time data processing with the superiority of high-speed computation and low-delay transmission [5, 6]. Compared with cloud computing, edge computing alleviates the high transmission delay and network congestion by reducing transmission distance between devices and servers. In addition, it outperforms fog computing in terms of privacy, security, and computation capacity due to the unified security management and abundant computation resources.

Computation offloading plays a crucial role in 5G and edge computing-integrated EIoT, which includes server selection and computation resource allocation [7, 8]. First, devices select an appropriate server and offload the computation task to the selected edge server via 5G for real-time processing. Then, edge servers allocate computation resources based on service requirements and computation capacity to

reduce computation delay [9]. However, the computation offloading optimization for 5G and edge computing-integrated EIoT still faces several challenges.

(i) Difficulty in tradeoff balance among various QoS parameters: services in 5G and edge computing-integrated EIoT have differentiated quality of service (QoS) requirements in delay, reliability, and so on. The QoS parameters are mutually influencing; e.g., improving reliability by using extra signaling overhead may harm delay and vice versa. Hence, it is necessary to balance the tradeoff among various QoS parameters through computation offloading optimization.

(ii) Coupling between large timescale server selection and small timescale computation resource allocation: server selection in large timescale leads to changes in the characteristics and number of devices served by servers, which in turn affects the computation resource allocation in small timescale. The inefficient decisions of computation resource allocation in small timescale directly affect the data computation delay performance of devices, which in turn affects the evaluation of the empirical performance of the servers in large timescale and may result in frequent server selection switching.

(iii) Multi-device competition in server selection and computation resource allocation: the contradiction between limited computation resources and multiple devices not only leads to competition for powerful servers in large timescale server selection but also causes competition for CPU frequency in small timescale computation resource allocation. In particular, the performance of delay and reliability is severely reduced by the competition for the same server.

Computation offloading within multi-access edge computing (MEC) has drawn extensive research in academia. In [10], Zaman et al. provided a survey of mobility management-based task offloading in edge networks. The taxonomy of research work classification is based on objectives, constraints, models, scenarios, and so on, and several future research directions for offloading in edge computing were presented. Lagrange dual approach has been widely adopted for addressing computation offloading problems with convex properties. In [11], Zhou et al. studied a multi-access edge computing-based task offloading algorithm for lightweight user by combining machine learning with Lagrange dual approach to minimize offloading energy consumption. In [12], Wang et al. proposed a Lagrange dual method-based communication resource allocation algorithm to minimize the weighted sum-energy consumption in MEC. Nevertheless, Lagrange dual approach cannot handle nonconvex problems which are common in multi-access networks. As an alternative approach, game theory is explored to cope with multi-user task offloading problems in MEC. In [13], Apostolopoulos et al. proposed a noncooperative game-based distributed cognitive data offloading algorithm to maximize users' utilities. In [14], Wang et al. proposed a noncooperative game-based computation offloading algorithm to adjust the offloading probability of each user and maximize vehicular utility. The drawback of game theory lies on required prior knowledge to derive utilities of players, which are essential to solve both cooperative and noncooperative games. Compared with game theory, reinforcement learning is more suitable for addressing computation offloading problems with uncertain information. In [15], Dinh et al. studied a model-free reinforcement learning-based computation offloading mechanism to maximize the utility function of each mobile user. In [16], Gao et al. proposed a novel Q learning-based computation offloading scheme for MEC system to minimize the system loss function. However, the aforementioned works have not considered the difficulty in tradeoff balance among various QoS requirements such as delay and reliability. Moreover, they focus on single timescale computation offloading optimization, while the coupling and mutual interconnection between large timescale server selection and small timescale computation resource allocation are neglected. The elimination of multi-access competition requires cooperation among different devices, which imposes another challenge on reinforcement learning-based computation offloading.

Matching theory is an effective method to solve the optimization of mutual relationship between two sides [17]. In [18], Seng et al. proposed an efficient task virtual machine matching algorithm to coordinate the offloading problems among mobile users and edge servers. In [19], Zhou et al. proposed a low complexity and stable computation offloading mechanism to minimize the total network delay based on pricing-based matching. However, the existing works only consider the one-to-one matching, which is not suitable to the scenario that the same server can serve multiple devices at the same time. In [20], Liu et al. proposed a many-to-one matching-based channel allocation algorithm for mobile users, which achieves higher throughput performance. In [21], Wang et al. proposed a many-to-one matching theory-based subchannel allocation algorithm to reduce the energy efficiency. However, the joint optimization of reliability and delay is ignored. In [22], Zhang et al. proposed a joint optimization algorithm of computation offloading and computation resource allocation, which optimizes network stability and achieves the tradeoff between energy efficiency and average service delay. Nevertheless, the aforementioned works do not consider the coupling between small timescale computation resource allocation and large timescale sever selection.

Motivated by the aforementioned challenges including difficulty in tradeoff balance among various QoS parameters, coupling between large timescale server selection and small timescale computation resource allocation, and the multi-device competition in server selection and computation resource allocation, we propose an empirical matching-based computation offloading optimization algorithm for 5G and edge computing-integrated EIoT. The optimization objective is to minimize the computation offloading delay by jointly optimizing large timescale server selection and

small timescale computation resource allocation. We first model the large timescale server selection problem as a many-to-one matching problem, which can be decoupled from small timescale computation resource allocation by establishing matching preference lists based on empirical performance. Then, the large timescale server selection problem is solved by pricing-based matching with quota algorithm. Furthermore, based on the obtained suboptimal result of large timescale server selection, the small timescale computation resource allocation problem is subsequently solved by Lagrange dual decomposition, the result of which is used to update large timescale empirical performance. Finally, extensive simulations are carried out to demonstrate the superior performance of the proposed algorithm by comparing it with existing algorithms. The main contributions of this paper are summarized as follows:

(i) Differentiated QoS guarantee: differentiated QoS demands such as reliability and delay are guaranteed through the minimization of the long-term total delay of all devices under the constrains of server computation capacity and transmission signal-to-interference-plus-noise ratio (SINR). Specifically, reliability is enforced by eliminating servers which cannot satisfy the SINR constraint from the device preference lists.

(ii) Multi-timescale computation offloading optimization: server selection is optimized in large timescale based on empirical matching with empirical performance enabled matching preference list and pricing-based matching with quota enabled matching conflict elimination. On the basis, computation resource allocation is optimized in small timescale based on Lagrange dual decomposition to reduce computation delay.

(iii) Extensive performance evaluation: compared with two state-of-the-art algorithms, simulation results demonstrate that the proposed algorithm has superior performance in computation offloading delay. Moreover, the impacts of key parameters such as quota and SINR threshold on delay performance are revealed to provide guidance for the real-world implementation of computation offloading optimization in 5G and edge computing-integrated EIoT.

The rest of the paper is organized as follows. Section 2 demonstrates the system model of computation offloading for 5G and edge computing-integrated EIoT and formulates the multi-timescale joint optimization problem. Section 3 introduces the proposed empirical matching-based computation offloading optimization algorithm. The simulation results are shown in Section 4. Section 5 concludes the paper.

2. System Model

In this paper, we consider a scenario of computation offloading for 5G and edge computing-integrated EIoT, as shown in Figure 1. The set of M devices and N base stations (BSs) are represented as $\mathcal{M} = \{1, 2, \cdots, m, \cdots, M\}$ and $\mathcal{N} = \{1, 2, \cdots, n, \cdots, N\}$, respectively. The edge server and BS are located in the same place to provide computation and communication services. For simplicity, we denote the sets of BS and edge server as $\mathcal{N}$. An example is shown in Figure 1, where the process of computation offloading includes two stages, i.e., server selection and computation resource allocation. In the first stage, when devices 1, 2, and 3 select the same BS 1 with limited offloading quota of 2 devices, BS 1 chooses to allocate computation resources for devices 1 and 2 and reject the access request of device 3. Device 3 finally offloads the computation task to BS 2 with a larger quota. We assume that BSs receive the time synchronization signal from the 5G time synchronization network and broadcast to all edge servers and devices to ensure that computation offloading is performed on the basis of time synchronization [15, 23].

A slot model is adopted [24] where server selection is optimized in large timescale, i.e., epoch, and computation resource allocation is optimized in small timescale, i.e., slot. The total optimization period has G epochs, each of which consists of T_0 slots, i.e., $T = GT_0$. The sets of epochs and slots are denoted as $\mathcal{G} = \{1, \cdots, g, \cdots, G\}$ and $\mathcal{T} = \{1, \cdots, t, \cdots, T\}$. The slot set of the g-th epoch is given by

$$\mathcal{T}(g) = \{(g-1)T_0 + 1, (g-1)T_0 + 2, \cdots, gT_0\}. \quad (1)$$

The key notations used in this paper are summarized in Table 1.

2.1. Transmission Model. At each epoch, a device selects a server for computation offloading. Define $x_{m,n}(g)$ as the server selection variable, where $x_{m,n}(g) = 1$ represents that device m selects server n for computation offloading, and otherwise, $x_{m,n}(g) = 0$. Considering uplink data transmission, the SINR between device m and server n in the t-th slot is [25]

$$\gamma_{m,n}(t) = \frac{P_{TX}(t)h_{m,n}(t)}{B_n N_0 + \lambda_{m,n}(t)}, \quad (2)$$

where B_n is the transmission bandwidth. $P_{TX}(t)$ is the transmission power, $h_{m,n}(t)$ is the channel gain between device m and server n in the t-th slot, N_0 is the noise spectral power density, and $\lambda_{m,n}(t)$ is the electromagnetic interference power.

To ensure reliability requirement, SINR constraint is given by

$$\gamma_{m,n}(t) \geq \gamma_{\min}, \quad (3)$$

where $\gamma_{\min}$ is SINR threshold for data reliable transmission. The transmission rate between device m and server n in the t -th slot is given by

$$R_{m,n}(t) = B_n \log\left(1 + \gamma_{m,n}(t)\right). \quad (4$$

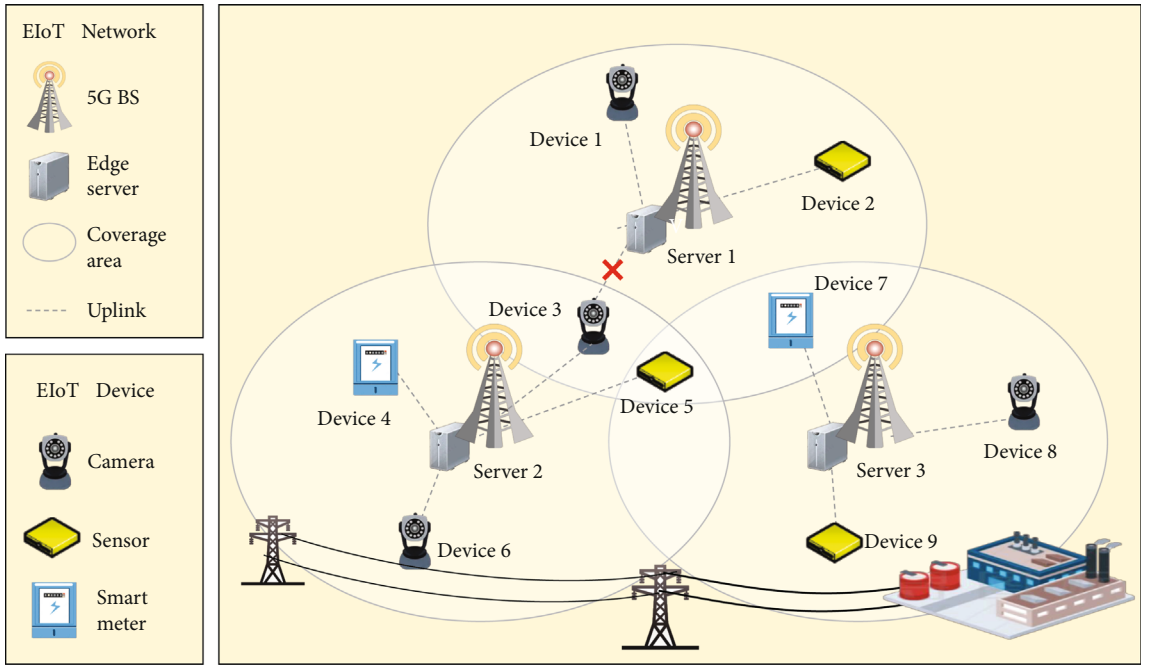

Figure 1: Computation offloading for 5G and edge computing-integrated EIoT.

Table 1: Summary of notations.

Notations	Definition	Notations	Definition
M	Number of devices	$\tau^{O}_{m,n}(t)$	Transmission delay of m to n in the t-th slot
N	Number of BSs	$\tau^{C}_{m,n}(t)$	Computation delay for processing $A_m(t)$ in the t-th slot
G	Number of epochs	$\tau_{m,n}(t)$	The total computation offloading delay
T	Number of slots	$f_{n,\max}$	The maximum computation resources
$\mathcal{G}$	Set of epochs	$f_{m,n}(t)$	Allocated computation resources by n to m
$\mathcal{T}$	Set of slots	$\mathcal{L}$	The preference list of devices
$\mathcal{M}$	Set of devices	$\gamma_{m,n}(t)$	SINR between m and n in the t-th slot
$\mathcal{N}$	Set of BSs or edge servers	$R_{m,n}(t)$	Transmission rate between m and n in the t-th slot
$x_{m,n}(g)$	Server selection variable	$\theta_{m,n}(g)$	Preference value of m towards n in the g-th epoch
B_n	Transmission bandwidth	$p_n(g)$	Server price of matching m with n
$P_{TX}(t)$	Transmission power	$\tilde{\tau}_{m,n}(g)$	Average offloading delay of m in the g-th epoch
$h_{m,n}(t)$	Channel gain between m and n	$\gamma_{\min}$	SINR threshold for data reliable transmission
N_0	Noise spectral power density	z	Iterative index of Lagrangian multiplier update
$\lambda_{m,n}(t)$	Electromagnetic interference power	$\rho_{\zeta_n(t)}(t,z)$	The update step length of $\zeta_n(t)$
$\zeta_n(t)$	Lagrangian sub vector	ε_m	Data computation density of m
Q_n	The quota of n	$A_m(t)$	The size of packets arriving at m in the t-th slot

We assume that the size of packets arriving at device m in the t-th slot is $A_m(t)$. Hence, the transmission delay of device m offloading $A_m(t)$ data to server n in the t-th slot is

$$\tau^{O}_{m,n}(t) = \frac{A_m(t)}{R_{m,n}(t)}. \tag{5}$$

2.2. Computation Model. Define $f_{m,n}(t)$ as the allocated computation resources by server n to device m. The computation delay for processing $A_m(t)$ data in the t-th slot [26] is given by

$$\tau^{C}_{m,n}(t) = \frac{\varepsilon_m A_m(t)}{f_{m,n}(t)}, \tag{6}$$

where ε_m is the data computation density of device m, i.e., the required CPU cycles to process one bit data.

The total allocated computation resource of server n should be lower than the maximum computation resource $f_{n,\max}$, i.e.,

$$\sum_{m=1}^{M} f_{m,n}(t) \le f_{n,\max}. \tag{7}$$

2.3. Problem Formulation. The total delay that device m selects server n for computation offloading is composed of the transmission delay and the computation delay, i.e.,

$$\tau_{m,n}(t) = \tau_{m,n}^{O}(t) + \tau_{m,n}^{C}(t). \tag{8}$$

The objective is to minimize the long-term total delay of all devices under the constrains of servers' computation capacity and transmission reliability, which can be expressed as

$$\begin{aligned}
\mathrm{P1}: & \min_{\{x_{m,n}(g)\},\{f_{m,n}(t)\}} \sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{n=1}^{N} x_{m,n}(g)\tau_{m,n}(t) \\
\text{s.t.} \quad & C_1: x_{m,n}(g) = \{0,1\}, \quad \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \forall g \in \mathcal{G}, \\
& C_2: \sum_{m=1}^{M} x_{m,n}(g) \le Q_n, \quad \forall n \in \mathcal{N}, \forall g \in \mathcal{G}, \\
& C_3: \sum_{n=1}^{N} x_{m,n}(g) = 1, \quad \forall m \in \mathcal{M}, \forall g \in \mathcal{G}, \\
& C_4: \sum_{m=1}^{M} f_{m,n}(t) \le f_{n,\max}, \quad \forall n \in \mathcal{N}, \forall t \in \mathcal{T}, \\
& C_5: \gamma_{m,n}(t) \ge \gamma_{\min}, \quad \forall m \in \mathcal{M}, \forall t \in \mathcal{T},
\end{aligned} \tag{9}$$

where Q_n is the quota of server n, C_2 represents that each edge server can be selected by at most Q_n devices at the same time, C_3 represents that each device can select only one edge server for computation offloading in an epoch, C_4 represents the edge server computation resource constraint, and C_5 represents the SINR constraint.

We neglect energy consumption because major delay-sensitive EIoT devices can continuously draw energy from the power grid. Nevertheless, the proposed algorithm can also be extended to the scenario where energy consumption is minimized.

3. Empirical Matching-Based Computation Offloading Optimization Algorithm for 5G and Edge Computing-Integrated EIoT

In this section, we introduce the problem transformation. Then, we propose the empirical matching-based large timescale server selection algorithm and Lagrange dual decomposition-based small timescale computation resource allocation for 5G and edge computing-integrated EIoT.

3.1. Problem Transformation. The long-term stochastic joint optimization problem P1 includes two subproblems, i.e., large timescale server selection problem and small timescale computation resource allocation problem. The large timescale server selection problem is first modeled as a many-to-one matching problem, which is decoupled from small timescale computation resource allocation by establishing matching preference list based on empirical performance of delay and reliability. Then, the large timescale server selection problem is solved by pricing-based matching with quota algorithm. Based on the obtained suboptimal result of large timescale server selection strategy, the small timescale computation resource allocation problem is subsequently solved by Lagrange dual decomposition, the result of which is used to update large timescale empirical performance.

3.2. Empirical Matching-Based Large Timescale Server Selection. The large timescale server selection optimization problem is transformed into a many-to-one matching problem, which is defined as a triple $(\mathcal{M}, \mathcal{N}, \mathcal{L})$, where $\mathcal{M}$ and $\mathcal{N}$ represent the sets of matching devices and servers, respectively, and $\mathcal{L}$ represents the preference list of devices.

Theorem 1. *A matching ϕ is a many-to-one correspondence of set $\mathcal{M}$ to set $\mathcal{N}$ based on the preference list $\mathcal{L}$. $\phi(m) = n$ represents the matching of device m and server n, i.e., $x_{m,n}(g) = 1$. Specially, server n can match at most Q_n devices at the same time.*

To decouple the large timescale server selection from the small timescale resource allocation, the preference list is established based on empirical performance of delay and reliability. The empirical matching-based large timescale server selection algorithm is summarized in Algorithm 1, including preference list establishment and the pricing-based iterative matching [27].

3.2.1. Preference List Establishment. In the initialization, each EIoT device first traverses all servers once to obtain the value of $\tilde{\tau}_{m,n}(g)$. At the beginning of the g-th period, the server selection decision is made based on the empirical information up to the current period. Thus, the preference value of device m towards server n in the g-th period is defined as $\theta_{m,n}(g)$, which is expressed as

$$\theta_{m,n}(g) = -\frac{\sum_{i=0}^{g-1} x_{m,n}(i)\tilde{\tau}_{m,n}(i)}{\sum_{i=0}^{g-1} x_{m,n}(i)} - p_n(g), \tag{10}$$

where $p_n(g)$ represents the server price of matching device m with server n, the initial value of which is set as zero and $\tilde{\tau}_{m,n}(g)$ represents the average offloading delay of device m in the g-th epoch, which is expressed as

$$\tilde{\tau}_{m,n}(g) = \frac{1}{T_0} \sum_{t=(g-1)T_0}^{gT_0} \sum_{n=1}^{N} x_{m,n}(t)\tau_{m,n}(t). \quad (11)$$

3.2.2. The Implementation Procedure of Pricing-Based Iterative Matching

3.2.3. Empirical Matching-Based Large Timescale Server Selection Algorithm

(Step 1) When $g = 0$, initialize $x_{m,n}(g) = 0$, $q_n = Q_n$, $p_n(g) = 0$, $\forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \forall g \in \mathcal{G}$ [1]. Define Θ as the set of unmatched devices, and set $\Theta = \mathcal{M}$ at the beginning. Define Γ_n as the set of devices proposed to server n, and set $\Gamma_n = \varnothing$ at the beginning. All devices traverse all servers once to complete the initialization of the preference list $g \le G$.

(Step 2) When $g \le G$, each $m \in \Theta$ calculates its preference value $\theta_{m,n}(g)$ towards n as (10) and sorts $\theta_{m,n}(g)$ in descending order to obtain the preference list $\mathcal{L}_m(g)$.

(Step 3) Each $m \in \mathcal{M}$ proposes its most preferred server in $\mathcal{L}_m(g)$. Each server, e.g., n, adds m into Γ_n if m proposes n. For $n \in \mathcal{N}$, if $|\Gamma_n| \le q_n$, n matches with the devices, e.g., m, from the proposed devices, i.e., $x_{m,n}(t) = 1$. Update $q_n = q_n - |\Gamma_n|$. Update $\Theta = \Theta \setminus m$. Otherwise, n updates the price $p_n(g)$ as (13). Unmatched devices execute steps $2 \sim 3$, until every device m has been matched with a server, i.e., $\phi(m) \neq \varnothing$, or there exists no available server for unmatched device m.

The implementation procedure of pricing-based matching consists of three phases, which are introduced as follows.

Step 1. When $g = 0$, initialize $x_{m,n}(g) = 0$, $q_n = Q$, $p_n(g) = 0$, $\forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \forall g \in \mathcal{G}$, the set of unmatched devices as $\Theta = \mathcal{M}$, and the set of devices which proposed to server n as $\Gamma_n = \varnothing$.

Step 2. Each $m \in \Theta$ calculates the preference value $\theta_{m,n}(g)$ for unmatched device m towards n as (10) and sorts $\theta_{m,n}(g)$ in a descending order to obtain the preference list $\mathcal{L}_m(g)$. We introduce $\succ_m$ to compare the preferences towards different servers. For example, $n \succ_m n'$ means that device m prefers server n than server n', which is expressed as

$$n \succ_m n' \Leftrightarrow \theta_{m,n}(g) > \theta_{m,n'}(g). \quad (12)$$

When a server is selected by multiple devices exceeding to its quota, the server increases the price $p_n(g)$ with increment Δp_n to eliminate matching conflicts, which is expressed as

$$p_n(g) = p_n(g) + \Delta p_n. \quad (13)$$

Then, the preference list $\mathcal{L}_m(g)$ is updated according to new preference value as (10) and (12).

Step 3. Each $m \in \mathcal{M}$ proposes to its most preferred server in $\mathcal{L}_m(g)$. Then, each server, e.g., n, adds m into Γ_n if m proposes to n. n matches with the devices, e.g., m, from the proposed devices, i.e., $x_{m,n}(g) = 1$. Repeat until every device m has been matched with a server.

Therefore, the optimal matching result can be obtained as

$$x_{m,n}(g) = 1 \Leftrightarrow \phi(m) = n. \quad (14)$$

3.2.4. Complexity Analysis. The computation complexity of the proposed server selection algorithm mainly depends on preference value calculation, preference list establishment, and pricing-based iterative matching process, where the complexities of these three steps are, respectively $\mathcal{O}(MN)$, $\mathcal{O}(MN \log (MN))$, and $\mathcal{O}(MN)$. Hence, the complexity of the proposed algorithm is $\mathcal{O}(2MN + MN \log (MN))$.

3.3. Lagrange Dual Decomposition-Based Small Timescale Computation Resource Allocation. Based on the many-to-one matching-based large timescale server selection, the small timescale computation resource allocation is subsequently optimized. For edge server n, the computation resource allocation problem in the t-th slot can be expressed as

$$\begin{aligned} &\mathrm{P2}: \min_{\{f_{m,n}(t)\}} \Psi_n(t) \\ &\text{s.t.} \quad C_3: \sum_{m=1}^{M} f_{m,n}(t) \le f_{n,\max}, \forall n \in \mathcal{N}, \forall t \in \mathcal{T}, \end{aligned} \quad (15)$$

where $\Psi_n(t)$ can be expressed as

$$\Psi_n(t) = \sum_{m=1}^{M} \tau_{m,n}^{C}(t) = \sum_{m=1}^{M} \frac{\varepsilon_m A_m(t)}{f_{m,n}(t)}. \quad (16)$$

Based on Jensen's inequality theorem [28], (16) satisfies $f(\mathbb{E}(\Psi_n(t)) \le \mathbb{E}(f(\Psi_n(t))))$ while $f(\cdot)$ is a convex function. Therefore, P2 is a convex function and can be solved by Lagrange dual decomposition [29]. Define $\zeta_n(t)$ as a Lagrangian subvector corresponding to constraint C_3. The Lagrange equation can be expressed as

$$L\left(f_{m,n}(t), \zeta_n\right) = \Psi_n\left(f_{m,n}(t)\right) + \zeta_n(t)\left[\sum_{m=1}^{M} f_{m,n}(t) - f_{n,\max}\right]. \quad (17)$$

Based on Lagrange dual decomposition, P2 can be transformed as

$$\max_{\zeta_n(t)>0} \min_{\{f_{m,n}(t)\}} L\left(f_{m,n}(t), \zeta_n(t)\right). \quad (18)$$

Based on Karush-Kuhn-Tucker conditions, it can obtain

$$f_{m,n}(t, z+1) = \max\left\{\arg_{f_{m,n}(t)} \nabla_{f_{m,n}}(t) = 0, 0\right\}, \quad (19)$$

where z is the iterative index of Lagrangian multiplier update. Lagrangian multiplier $\zeta_n(t)$ update can be expressed as

$$\zeta_n(t, z+1) = \max\left\{\zeta_n(t, z) + \left[\sum_{m=1}^{M} f_{m,n}(t) - f_{n,\max}\right]\rho_{\zeta_n(t)}(t, z), 0\right\}, \quad (20)$$

where $\rho_{\zeta_n(t)}(t, z)$ is the update step length.

4. Simulation Results

In this section, we introduce the simulation results and analysis. We consider a computation offloading scenario of EIoT with 20 devices and 5 servers. In the case of largescale fading, the channel gain is calculated according to $h_{m,n}(t) = 127 + 30\log(r_{m,n})$ [30], where $r_{m,n}$ is the distance between device m and server n, and distributed within [0.03,0.05] km. According to the actual environment of EIoT, the electromagnetic interference power varies from 28 dBm to 30 dBm. The specific simulation parameters are summarized in Table 2 [31–35]. Two existing algorithms are employed for comparison. The first one is the price matching-based computation offloading (PMCO) algorithm [27], where server-side computation resources are allocated equally. The second one is the greedy matching-based computation offloading (GMCO) algorithm [36], where large timescale server selection is solved based on greedy matching strategy and small timescale computation resource allocation is implemented once in the first time slot of each epoch. In addition, we extend the proposed algorithm to a large-scale computation offloading scenario with 100 devices to verify its effectiveness and robustness. The simulation software is MATLAB 2021. The real execution time of the proposed algorithm is about 49 ms per time slot, which achieves real-time or near-real-time. When adopting dedicated execution module such as DSP, the real execution time can be further reduced to millisecond level or even microsecond level.

4.1. Transmission Delay Performance. Figure 2 shows the box plots of transmission delay. Compared with PMCO and GMCO, the proposed algorithm can reduce the median transmission delay by 13.85% and 19.82% and reduce the transmission delay fluctuation by 65.79% and 68.29%, respectively. The proposed algorithm utilizes empirical matching-based large timescale server selection to reduce the transmission delay, thus performing the best. PMCO outperforms GMCO in terms of transmission delay because PMCO considers the server selection optimization based on price matching, thus effectively reducing the transmission delay.

4.2. Computation Delay Performance. Figure 3 show the box plots of computation delay. Compared with PMCO and GMCO, the median computation delay is decreased by 24.78% and 16.05%, and the computation delay fluctuation is decreased by 46.77% and 31.25%, respectively. The proposed algorithm utilizes Lagrange dual decomposition-based small timescale computation resource allocation to improve computation delay performance, thus performing the best. PMCO performs worse than GMCO in terms of computation delay. The reason is that GMCO considers the computation resource allocation, which leads to a lower computation delay.

Table 2: Simulation parameters.

Parameter	Value	Parameter	Value
M	20	N	5
Q_n	6	G	20
T	200	T_0	10
B_n	0.1 MHz	$\gamma_{\min}$	10 dB
A_m	[0.2,0.5] Mbits	P_{TX}	0.1 W
$f_{n,\max}$	[10, 30] GHz	N_0	−114 dBm
Δp_n	5	ε_m	1000 cycles

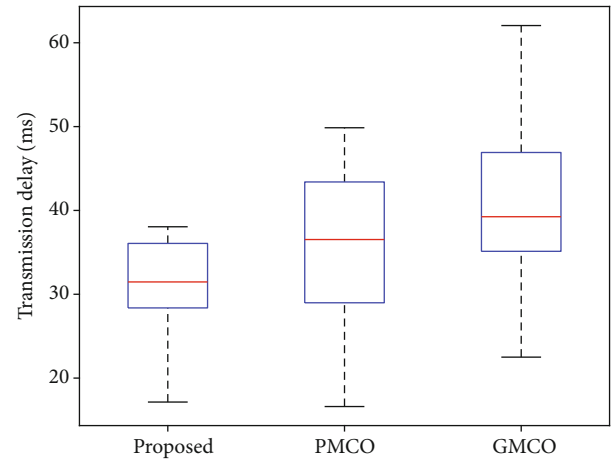

Figure 2: Transmission delay of different algorithms.

4.3. Total Computation Offloading Delay Performance. Figure 4 shows the total computation offloading delay versus time slot. The proposed algorithm outperforms PMCO and GMCO by 14.96% and 19.79%. The reason is that the proposed algorithm can avoid adversarial channels to improve delay performance through large timescale server selection. Then, given the large timescale server selection strategies, it further optimizes the small timescale computation resource allocation in each slot. However, PMCO neglects the optimization of computation resource allocation. The GMCO algorithm based on greedy matching and single-slot computation resource allocation cannot solve matching conflict from the perspective of the whole network and adjust the allocation strategies based on real-time information.

Figure 5 shows the total computation offloading delay versus number of devices. As the number of devices increases from 10 to 100, the total computation offloading

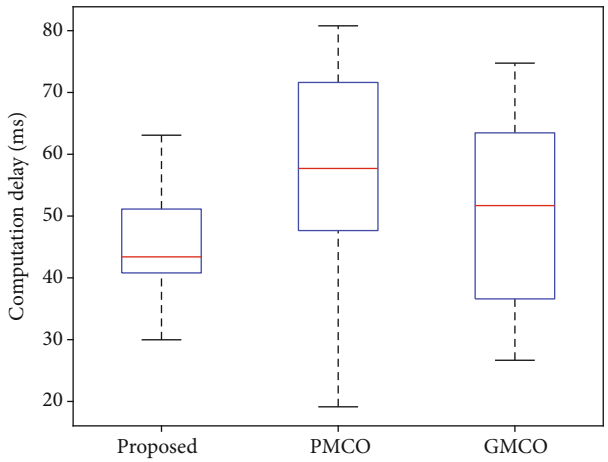

Figure 3: Computation delay of different algorithms.

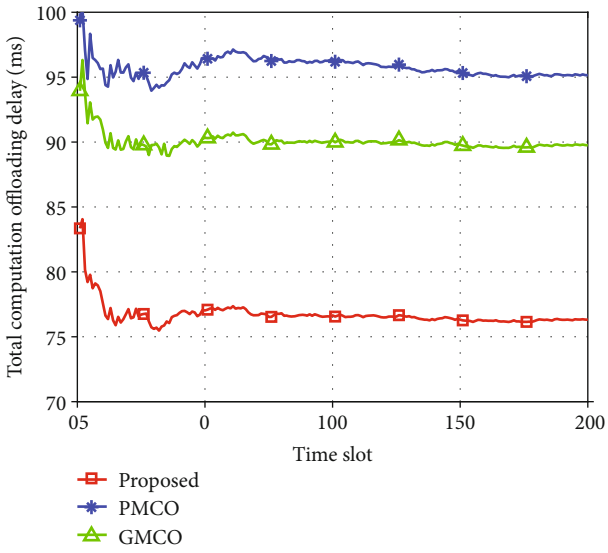

Figure 4: Total computation offloading delay versus time slot.

delay of PMCO and GMCO increases by 28.21% and 28.99 %, while the total computation offloading delay of the proposed algorithm is slightly increased by 10.14%. The reason why total computation offloading delay increases is that the conflicts caused by multiple devices selecting the same server increases, and the computation resource allocated to each device is poorer. Even in a multi-access scenario with 100 devices, the proposed algorithm can adaptively balance the performance return of all devices through pricing-based iterative matching and adjust the resource allocation strategies based on dynamic data arrival.

Figure 6 shows the impact of the quota Q_n on transmission delay and computation delay. The bars in the figure represent the transmission delay and the curves represent the computation delay. When Q_n increases from 4 to 8, the transmission delay and computation delay of PMCO and GMCO are decreased by 7.85%, 6.62%, 6.79%, and 11.13% , in which those of the proposed algorithm are obviously decreased by 14.11% and 11.13%. The reason for it is that the final delay performance depends on the worst computation offloading situation, where the larger Q_n means the greater offloading possibility to servers with better channels and computation resources for devices. In addition, the proposed algorithm can well coordinate the competition among multiple devices for servers and computation resources based on empirical information and real-time resource allocation.

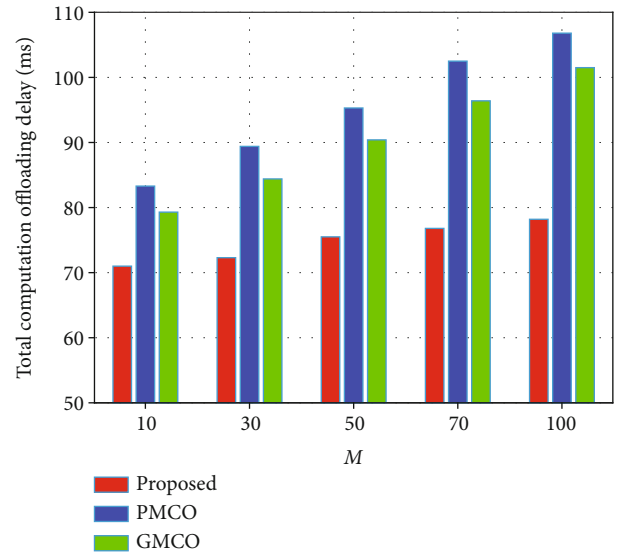

Figure 5: Total computation offloading delay versus number of devices M ($M = 10 \sim 100$).

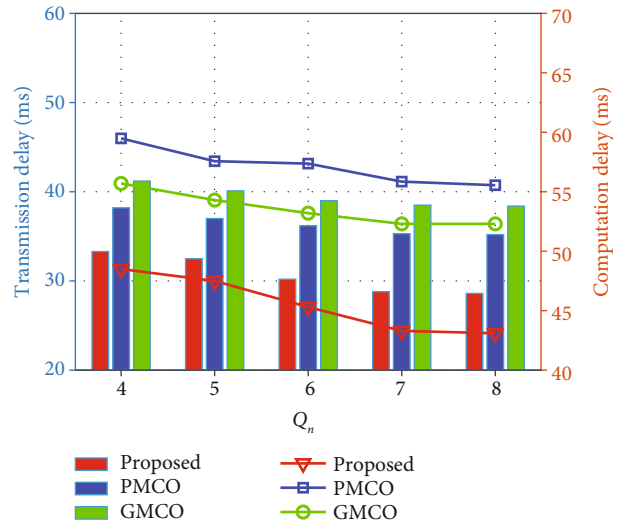

Figure 6: The impact of quota Q_n on transmission delay and computation delay.

Table 3 shows the impact of $\gamma_{\min}$ on total computation offloading delay. With the increasing of $\gamma_{\min}$, the total computation offloading delay decreases first and then increases. When $\gamma_{\min} = 12$ dB, the total computation offloading delay reaches the minimum value. The reason is that when $\gamma_{\min}$ increases from 8 to 12 dB, stringent reliability constraints impel devices to select servers with better empirical channel

Table 3: The impact of $\gamma_{\min}$ on total computation offloading delay.

$\gamma_{\min}$ (dB)	8	10	12	14	16
Proposed	75.24 ms	74.80 ms	73.25 ms	74.46 ms	75.82 ms
PMCO	95.80 ms	94.23 ms	93.15 ms	95.12 ms	95.60 ms
GMCO	92.73 ms	90.95 ms	89.82 ms	90.23 ms	92.86 ms

states, thus significantly reducing transmission delay at the beginning. However, as $\gamma_{\min}$ continues to increase from 12 to 16 dB, the servers with better channel states will be occupied by more devices resulting in a higher computation delay.

5. Conclusion

In this paper, we investigated a multi-timescale computation offloading optimization problem for 5G and edge computing-integrated EIoT. Specifically, we proposed an empirical matching-based computation offloading optimization algorithm, which includes the joint optimization of large timescale server selection and small timescale computation resource allocation. Compared with the existing PMCO and GMCO algorithms, the proposed algorithm could reduce the total computation offloading delay by 14.96% and 19.79%, respectively. It could also reduce the median transmission delay by 13.85% and 19.82%, the transmission delay fluctuation by 65.79% and 68.29%, the median computation delay by 24.78% and 16.05%, and the computation delay fluctuation by 46.77% and 31.25%. Finally, future research directions are outlined.

5.1. Computation Offloading under Extensive EIoT Scenarios with Incomplete Information. In large-scale EIoT scenarios, considering the prohibitive signaling overheads, global state information such as the channel state between devices and servers, the amount of available computation resources of servers, and the computation resource allocation policy of servers is unknown to devices. Server selection needs to be optimized under incomplete information. A feasible approach is to empower computation offloading for EIoT by learning through interaction with the environment by exploring advanced artificial intelligence algorithms such as deep reinforcement learning and federated learning.

5.2. Insecure Computation Offloading under Malicious Attack. With the extensive access of massive devices, EIoT faces various malicious attacks including distributed denial of services (DDoS), time synchronization attack, sinkhole attack, and tampering attack. These malicious attacks make computation offloading in EIoT insecure, which results in loss and tampering of important information, such as fault data, alarm data, and load fluctuation, endangering the supply-demand balance as well as stable and secure operation of power grid. To solve this problem, advanced security technologies such as blockchain, identity authentication, encryption, and trusted computing can be adopted to ensure the secure data transmission and processing in computing offloading.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Key R&D Program of China (2020YFB0905900) and is funded by the Science and Technology Project of SGCC (State Grid Corporation of China): The Key Technologies for Electric Internet of Things (SGTJDK00DWJS2100223).

References

[1] M. Ullah and J. Park, "Distributed energy trading in smart grid over directed communication network," *IEEE Transactions on Smart Grid*, vol. 12, no. 4, pp. 3669–3672, 2021.

[2] W. Costa, W. Santos, H. Rocha, M. Segatto, and J. Silva, "Power line communication based smartplug prototype for power consumption monitoring in smart homes," *IEEE Latin America Transactions*, vol. 19, no. 11, pp. 1849–1857, 2021.

[3] X. Li, Y. Tian, G. Ledwich, Y. Mishraand, and C. Zhou, "Minimizing multicast routing delay in multiple multicast trees with shared links for smart grid," *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5427–5435, 2019.

[4] P. Thiruvasagam, A. Chakraborty, A. Mathew, and C. Murthy, "Reliable placement of service function chains and virtual monitoring functions with minimal cost in softwarized 5G networks," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1491–1507, 2021.

[5] P. Zhang, Y. Zhang, H. Dong, and H. Jin, "Mobility and dependence-aware QoS monitoring in mobile edge computing," *IEEE Transactions on Cloud Computing*, vol. 9, no. 3, pp. 1143–1157, 2021.

[6] D. Wu, X. Huang, X. Xie, X. Nie, L. Bao, and Z. Qin, "LEDGE: leveraging edge computing for resilient access management of mobile IoT," *IEEE Transactions on Mobile Computing*, vol. 20, no. 3, pp. 1110–1125, 2021.

[7] F. Fang and X. Wu, "A win–win mode: the complementary and coexistence of 5G networks and edge computing," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 3983–4003, 2021.

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

[9] Z. Zhou, K. Ota, M. Dong, and C. Xu, "Energy-efficient matching for resource allocation in D2D enabled cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 5256–5268, 2017.

[10] S. K. U. Zaman, A. I. Jehangiri, T. Maqsood et al., "Mobility-aware computational offloading in mobile edge networks: a survey," *Cluster Computing*, vol. 24, no. 4, pp. 2735–2756, 2021.

[11] S. Zhou, W. Jadoon, and J. Shuja, "Machine learning-based offloading strategy for lightweight user mobile edge computing

tasks," *Complexity*, vol. 2021, Article ID 6455617, 11 pages, 2021.

[12] F. Wang, J. Xu, and Z. Ding, "Multi-antenna NOMA for computation offloading in multiuser mobile edge computing systems," *IEEE Transactions on Communications*, vol. 67, no. 3, pp. 2450–2463, 2019.

[13] P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, "Cognitive data offloading in mobile edge computing for internet of things," *IEEE Access*, vol. 8, no. 99, pp. 55736–55749, 2020.

[14] Y. Wang, P. Lang, D. Tian et al., "A game-based computation offloading method in vehicular multiaccess edge computing networks," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4987–4996, 2020.

[15] T. Q. Dinh, Q. D. La, T. Q. S. Quek, and H. Shin, "Learning for computation offloading in mobile edge computing," *IEEE Transactions on Communications*, vol. 66, no. 12, pp. 6353–6367, 2018.

[16] Z. Gao, W. Hao, Z. Han, and S. Yang, "Q-learning-based task offloading and resources optimization for a collaborative computing system," *IEEE Access*, vol. 8, no. 99, pp. 149011–149024, 2020.

[17] Z. Zhou, G. Ma, M. Dong, K. Ota, C. Xu, and Y. Jia, "Iterative energy-efficient stable matching approach for context-aware resource allocation in D2D communications," *IEEE Access*, vol. 4, pp. 6181–6196, 2016.

[18] S. Seng, C. Luo, X. Li, H. Zhang, and H. Ji, "User matching on blockchain for computation offloading in ultra-dense wireless networks," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1167–1177, 2021.

[19] Z. Zhou, H. Liao, X. Zhao, B. Ai, and M. Guizani, "Reliable task offloading for vehicular fog computing under information asymmetry and information uncertainty," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 9, pp. 8322–8335, 2019.

[20] X. Liu, Z. Qin, Y. Gao, and J. McCann, "Resource allocation in wireless powered IoT networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4935–4945, 2019.

[21] K. Wang, F. Fang, D. Costa, and Z. Ding, "Sub-channel scheduling, task assignment, and power allocation for OMA-based and NOMA-based MEC systems," *IEEE Transactions on Communications*, vol. 69, no. 4, pp. 2692–2708, 2021.

[22] Q. Zhang, L. Gui, F. Hou, J. Chen, S. Zhu, and F. Tian, "Dynamic task offloading and resource allocation for mobile-edge computing in dense cloud RAN," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3282–3299, 2020.

[23] T. Liu, Y. Zhang, Y. Zhu, W. Tong, and Y. Yang, "Online computation offloading and resource scheduling in mobile-edge computing," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6649–6664, 2021.

[24] H. Yu, Z. Zhou, Z. Jia, X. Zhao, L. Zhang, and X. Wang, "Multi-timescale multi-dimension resource allocation for NOMA-edge computing-based power IoT with massive connectivity," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 3, pp. 1101–1113, 2021.

[25] K. Wang, Y. Zhou, Q. Wu, W. Chen, and Y. Yang, "Task offloading in hybrid intelligent reflecting surface and massive MIMO relay networks," *IEEE Transactions on Wireless Communications*, p. 1, 2021.

[26] Z. Zhou, H. Liao, B. Gu, S. Mumtaz, and J. Rodriguez, "Resource sharing and task offloading in IoT fog computing: A contract-learning approach," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 3, pp. 227–240, 2020.

[27] C. Xu, J. Feng, Z. Zhou, Z. Chang, Z. Han, and S. Mumtaz, "Two-stage matching for energy-efficient resource management in D2D cooperative relay communications," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pp. 1–6, Singapore, 2017.

[28] C. Briat, "Convergence and equivalence results for the jensen's inequality— application to time-delay and sampled-data systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1660–1665, 2011.

[29] H. Liao, Z. Zhou, X. Zhao, and Y. Wang, "Learning-based queue-aware task offloading and resource allocation for space–air–ground-integrated power IoT," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5250–5263, 2021.

[30] H. Liao, Z. Zhou, W. Kong et al., "Learning-based intent-aware task offloading for air-ground integrated vehicular edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5127–5139, 2021.

[31] M. Tariq, M. Adnan, G. Srivastava, and H. Poor, "Instability detection and prevention in smart grids under asymmetric faults," *IEEE Transactions on Industry Applications*, vol. 56, no. 4, pp. 4510–4520, 2020.

[32] M. Tariq and H. Poor, "Electricity theft detection and localization in grid-tied microgrids," *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 1920–1929, 2018.

[33] X. Li, J. Li, Y. Liu, Z. Ding, and A. Nallanathan, "Residual transceiver hardware impairments on cooperative NOMA networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 680–695, 2020.

[34] X. Li, M. Zhao, M. Zeng et al., "Hardware impaired ambient backscatter NOMA systems: reliability and security," *IEEE Transactions on Communications*, vol. 69, no. 4, pp. 2723–2736, 2021.

[35] X. Li, Y. Zheng, M. D. Alshehri et al., "Cognitive AmBC-NOMA IoV-MTS networks with IQI: reliability and security analysis," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2021.

[36] S. Zhang, G. Cui, Y. Long, and W. Wang, "Joint computing and communication resource allocation for satellite communication networks with edge computing," *China Communications*, vol. 18, no. 7, pp. 236–252, 2021.