

Research Article

A Reputation Value-Based Task-Sharing Strategy in Opportunistic Complex Social Networks

Jia Wu ^{1,2}, Fangfang Gou,^{1,2} Wangping Xiong ¹ and Xian Zhou ¹

¹School of Computer, Jiangxi University of Chinese Medicine, NanChang 330004, JiangXi, China

²School of Computer Science and Engineering, Central South University, Changsha 410083, China

Correspondence should be addressed to Wangping Xiong; 20030730@jxutcm.edu.cn and Xian Zhou; 20030731@jxutcm.edu.cn

Received 10 October 2021; Accepted 5 November 2021; Published 26 November 2021

Academic Editor: Xuyun Zhang

Copyright © 2021 Jia Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As the Internet of Things (IoT) smart mobile devices explode in complex opportunistic social networks, the amount of data in complex networks is increasing. Large amounts of data cause high latency, high energy consumption, and low-reliability issues when dealing with computationally intensive and latency-sensitive emerging mobile applications. Therefore, we propose a task-sharing strategy that comprehensively considers delay, energy consumption, and terminal reputation value (DERV) for this context. The model consists of a task-sharing decision model that integrates latency and energy consumption, and a reputation value-based model for the allocation of the computational resource game. The two submodels apply an improved particle swarm algorithm and a Lagrange multiplier, respectively. Mobile nodes in the complex social network are given the opportunity to make decisions so that they can choose to share computationally intensive, latency-sensitive computing tasks to base stations with greater computing power in the same network. At the same time, to prevent malicious competition from end nodes, the base station decides the allocation of computing resources based on a database of reputation values provided by a trusted authority. The simulation results show that the proposed strategy can meet the service requirements of low delay, low power consumption, and high reliability for emerging intelligent applications. It effectively realizes the overall optimized allocation of computation sharing resources and promotes the stable transmission of massive data in complex networks.

1. Introduction

In recent years, with the deep integration and development of IoT technologies and industries, various revolutionary mobile devices have penetrated into infrastructure, life services, national defense, and military, giving rise to new IoT smart applications such as smart home, driverless, augmented reality/virtual reality (AR/VR), and face recognition [1]. These applications generate large amounts of data, and at the same time, they are computationally intensive and time-sensitive. In particular, some new applications based on big data and artificial intelligence have high requirements for low-latency transmission of large-capacity data, making mobile devices face huge challenges in terms of computing resources and computing capabilities.

In complex opportunistic social networks, both mobile smart terminal devices and servers can be considered as social nodes. These social nodes can communicate and share

computing and storage resources [2–4]. However, frequent communication between these nodes leads to the surge in traffic and high multivariate of data-type, which poses a challenge to network management. When processing computation-intensive and latency-sensitive services, mobile terminals can share the computation task to servers with greater computing power at base stations, instead of performing computation by themselves. Then, the servers can provide the communication, storage, and computation resources needed to process these functions and services through networks. Meanwhile, through the management and mining of the hidden knowledge in the massive data, the end-users can obtain high-quality, low-latency, low-consumption, and highly personalized services [5, 6].

However, in the process of sharing a large number of computing tasks from the terminal device to the server, how to allocate the server's computing resources to ensure the service performance of smart mobile terminals is an urgent

problem to solve [7]. In addition, wireless sensor networks have some inherent characteristics, such as limited node energy, storage space, and computational processing capabilities. These characteristics can record nodes movement, so the bad behavior nodes [8, 9] are prone to exist in the network. Bad behavior nodes are further divided into selfish behavior nodes and malicious attack behavior nodes. The selfish behavior node uses the network resources as much as possible, and it does not make any/makes very little contribution to the network. The sole purpose of malicious nodes is to attack other nodes in the network or the entire peer-to-peer network. Malicious nodes forge false hotspot resources and provide malicious resources to other nodes in the network to download or forward them to achieve the purpose of invading, controlling nodes, or even destroying the entire network. Malicious nodes are the direct source of security problems in peer-to-peer networks. Mass data have higher requirements for data transmission on complex networks. How to exclude malicious nodes from a large number of nodes is a big challenge.

These problems can be solved in two steps: firstly, to solve the decision problem of whether and when to share the tasks under latency and energy consumption demand. Secondly, to solve the problem of reliable sharing allocation arising from malicious competition for computing resources. By using the cooperation between nodes, the reputation value of each node is updated and the malicious node is finally derived. Therefore, we try to allocate server computing resources based on the reputation value of smart mobile terminals to prevent malicious behaviors and realize the optimal allocation of overall resources from the perspective of the benefits of a larger smart mobile terminal, not just from the perspective of the economic benefits of a larger service provider.

To solve the above problems, this paper proposes an advanced task-sharing model (DERV), which consists of a task-sharing decision model that takes into account the delay and energy consumption requirements of new IoT applications, and a resource game allocation model that allocates server computing resources based on reputation values. Among them, the shared decision model adopts an improved PSO algorithm to realize multi-task sharing among multiple mobile terminals, which meets the low latency and low energy consumption requirements of new applications of the Internet of Things. The resource allocation model aims to prevent unreasonable resource allocation caused by malicious terminal competition, and it is mainly realized by the Lagrangian multiplier method. In this way, computing resources are allocated reasonably and the overall utility is maximized. The DERV model realizes the stable transmission of massive data in complex social networks.

The contributions to this paper are listed as follows:

- (1) We focus on providing low-latency, low-energy, and highly reliable service quality guarantees for time-delay and energy-sensitive, computing-intensive smart mobile terminals in the big data environment. We first propose a network model composed of

users, MEC servers, and trusted authorities. Then, we formulate the optimal configuration of the overall resources as how to allocate the computing resources of the MEC server. Finally, we transform the optimization problem as offloading decision and offloading allocation subproblems.

- (2) For offloading decision issues, this paper proposes a task-sharing decision model for multiple smart mobile terminals in a complex opportunistic social network environment. It considers time delay and energy consumption comprehensively. The purpose is to achieve an optimal task-sharing solution with low latency and low energy consumption.
- (3) For offloading allocation issue, a dynamic task-sharing allocation game based on the reputation value of smart mobile terminals is proposed, and it is achieved by the Lagrange multiplier method.

The rest of the paper is arranged as follows: The second part introduces the related research of data transmission routing algorithms in opportunistic social networks, the third part introduces related theoretical concepts and algorithm models, and the fourth part verifies the performance of the model on various standards through simulation experiments. At the end of the paper, we discussed and summarized the full text.

2. Related Work

Since the task-sharing strategy promises to solve the performance bottlenecks faced by mobile smart terminals in opportunistic complex social networks when dealing with novel smart applications such as computation-intensive and latency-sensitive applications, it has received widespread attention from scholars and has gradually become a research hotspot. Therefore, in this section, we will give a brief introduction to the task-sharing study of whether and how much mobile smart terminals share computational tasks and how to allocate the macro base station computational resources.

2.1. Task-Sharing Model. In terms of task sharing, there are a large number of researches' results on whether mobile smart terminals share tasks to macro base stations. Task-sharing decision schemes often use delay and energy consumption as benchmarks. Different environments and systems have different requirements for task-sharing decisions, and some applications require a good balance between latency and energy consumption. Paper [10] presents the Lyapunov optimization-based dynamic computation offloading algorithm (LODCO), a dynamic computational task-sharing algorithm based on Lyapunov's optimization theory. It optimizes task-sharing decisions in terms of both task running delay and task running failure, minimizing task processing delay, and ensuring the success rate of the data transfer process, but ignoring influencing factors such as energy consumption and cost. Paper [11] adopts an artificial fish swarm algorithm to design a task-sharing strategy for

energy consumption optimization under delay constraints. The strategy takes full account of the link conditions in the task data transmission network and effectively reduces the energy consumption of the device, but the disadvantage is that the algorithm is too complex. The paper [12] designs an energy-efficient computational task-sharing scheme, which is based on deep learning, to solve the problem of selective sharing of mobile application components. Experimental results show that the solution has high accuracy and low energy consumption, but the computationally intensive task remote interaction brings about high latency problems. The influential factors considered in the above paper in the task-sharing decision modeling process are not comprehensive enough, which affects the rational execution of task-sharing decisions.

The paper [13] considers a multi-user MEC system in which a single MEC server can handle the computational tasks shared by multiple user devices over wireless channels. This solution has a significant reduction in the sum of delay and energy consumption. Paper [14] presents a Lyapunov optimization theory combined with an adaptive e-learning approach to the problem of optimal sharing of trade-offs between response latency and energy costs in IoT scenarios. Paper [15] optimizes multi-user mobile edge computing task-sharing systems, constructs Markov decision problems with the long-run average overhead of delay and power consumption as optimization goals, and solves them using convex optimization theory. Although the above paper treats the delay and energy consumption as important components in the computational task-sharing process, the method of reducing energy consumption by constraining the delay makes the task-sharing strategy lack generality. We noticed that there are many excellent energy-saving strategies in the Internet of Things (IoT) paradigm. The energy consumption models proposed by scholars have all been proved to be effective in important IoT applications such as organizational collaboration, staff track, and logistics positioning. Many strategies emphasize the realization of potential benefits in terms of energy and cost, and have been implemented on real test beds [16–18]. Therefore, based on summarizing previous studies, this paper introduces delay and energy demand coefficients to consider more comprehensively the delay and energy consumption of intelligent terminal node computing task sharing.

2.2. Resource Allocation Model. In terms of resource allocation, the current research work focuses on the design of an allocation strategy for the problem of how to allocate computing resources for macro base stations. Paper [19] enabled MEC's LTE-V network using a deep Q-Learning approach and proposed an optimization goal of maximizing the utility of the sharing system under a given delay constraint. The results show that the scheme can share vehicle tasks with optimal utility while also satisfying the reliability and wait time constraints but ignores the important resource allocation aspect of the task-sharing problem. Huang et al. [20] proposed a computing task plan based on mobile user security and cost awareness in a mobile edge computing

environment. Its goal is to minimize the total cost under the constraint of risk probability and provide security and cost efficiency for mobile users. However, it ignores the actual needs of the end nodes, which is not conducive to the overall optimization of resource allocation. The paper [21] studies the sharing strategies for computationally intensive tasks (data processing tasks and blockchain mining tasks) in blockchain scenarios, addressing the failure of traditional task-sharing strategies (e.g., auction and game theory strategies) to adjust the sharing strategies to changes in the environment, but the efficiency and reliability of the average resource allocation scheme are difficult to meet the quality of service (QoS) needs of users.

The above paper has achieved some results in the study of resource allocation in opportunistic complex networks, but it ignores the problem of smart mobile terminal nodes competing for resources, which makes it difficult to achieve a reasonable and reliable allocation of computing resources in opportunistic social IoT systems. If the malicious resource competition behavior of terminal nodes is regulated and the overall optimal allocation of computational task-sharing resources is achieved, low latency, low energy consumption, and highly reliable QoS guarantees can be provided for experimental and energy-sensitive terminal nodes [22]. We have noticed that trust models of wireless sensor networks (WSNs) security have flourished due to the day-to-day attack challenges, which are most popular for the Internet of things [23]. Many strategies have introduced a reputation mechanism to solve this problem, and achieved good results [24, 25]. For example, Han et al. [24] supplemented the fence-sitter group to the existing rumor dissemination model, and then proposed a novel SIFR rumor dissemination model, which effectively realizes the security monitoring of rumors dissemination in the network. Therefore, this paper innovatively introduces smart mobile terminal node reputation values to allocate computational resources in macro base station servers to effectively achieve the optimal allocation of computational resources [26].

To summarize, this paper builds a model (DERV) to address the shortcomings of task-sharing decision-making and resource allocation methods in complex social networks of things. The model takes into account the time delay and energy consumption of computational tasks and allocates computational resources based on the reputation value, which effectively achieves overall optimization, and rational and reliable allocation of computational resources.

3. System Model Design

3.1. Network Model. The MEC server allocates computing resources to perform the different computational task-sharing processes for users in the environment of opportunistic social networks of things based on different computational task requirements and reputation values of smart terminals [27]. The mobile edge intelligent computing network model based on reputation value consists of users holding smart mobile terminals (VR/AR, smart cars, video game consoles, PC monitors, drones, smart home, etc.), MEC servers, and trusted institutions. These devices

generate massive amounts of data, posing a challenge to network management. The network model is shown in Figure 1. And, a list of all acronyms used in the paper is shown in Table 1.

3.1.1. Users. The users who need to perform computation-intensive computing tasks in this paper have the computation, GPS, and wireless communication modules. The computing module performs computational tasks, the GPS module acquires location information in real-time, and the wireless communication module enables data transmission. When a user needs to run emerging applications such as augmented reality, image processing, etc., and it is difficult for the computation module to complete the corresponding computation and storage tasks, the computation task-sharing request is initiated by the wireless communication module and the computation task is sent to the specified MEC server.

3.1.2. MEC Servers. Distributed MEC servers are deployed in the vicinity of users and are responsible for allocating computation resources to perform different computation shares and returning data to users after completing computation tasks. Users have different reputation requirements for computing task sharing. In the case of partial sharing, when more computing tasks are shared, a small base station is set up to store the queued waiting tasks, and the stored tasks are uploaded to the macro base station when the macro base station is idle. In this paper, we use the full sharing decision, so we do not consider the case of deploying a small base station, but only the macro base station of the MEC server, and each user chooses a different task-sharing method according to different requirements.

3.1.3. Trusted Authority. A trusted authority is responsible for recording events and updating the reputation value database. All users must register their legal identity with the trustworthy authority, and users with legal identity have valid reputation value and obtain the public and private key pairs and certificates required for secure communication. The trusted authority updates the reputation value in real-time through the user's behavior records at different times and provides the service provider with access to the whole network reputation value database.

3.2. Mathematical Model. The nodal users connected to the complex social networks of things have different computational task-sharing needs, and in addition to performing local computation, they can also share the computational task to the macro base station where the MEC server is deployed [30]. In a multi-user participation scenario, whether to choose to share and how to allocate resources effectively after task sharing are issues that need to be addressed [31]. To solve these problems, this paper proposes an intelligent computing task-sharing method that jointly considers delay and energy consumption. We select an appropriate task-sharing strategy according to the

performance benchmarks of different smart mobile terminals held by users and the needs of smart mobile terminals to realize low-latency, low-energy-consumption computing task-sharing decision-making. In addition, to achieve the overall optimal allocation of computing resources, this paper proposes a reliable computing resource allocation model to realize a bargaining game based on user reputation values and maximize user benefits [32].

3.2.1. Shared Decision-Making Models. In this paper, a single cellular network model with multiple users and the deployment of MEC servers for macro base stations is constructed. Assuming $M = \{1, 2, \dots, m\}$ denotes m users holding different kinds of smart mobile terminals. Each user has one computation-intensive or latency-sensitive computation task. The computational task owned by user i is denoted as

$$U_i = \{A_i, B_i, T_i^{\max}\}, \quad i \in M. \quad (1)$$

In this equation, U_i indicates the computational task owned by user i . A_i indicates the data size of the computational task. B_i indicates the number of CPU cycles required to complete the computational task, and T_i^{\max} indicates the maximum latency that user i can tolerate completing the computational task.

According to the different needs of smart mobile terminal users, delay-sensitive smart mobile terminal users have a higher demand for time delay, while energy-sensitive smart mobile terminal users have a higher demand for energy-saving due to their power shortage, so a trade-off mechanism is introduced, and the time demand coefficient ∂_i^t and energy demand coefficient ∂_i^e indicate the bias degree of the user's demand for delay sensitivity and energy-saving, respectively. The computational tasks for each user can either be shared with the MEC server or performed locally. This paper introduces the decision mechanism $d_{i,j} = \{0, 1\}$, where $j = \{1, 2\}$ indicates the decision mode, $j = 1$ indicates that the user chooses the local computational model to perform the computational task, and $j = 2$ indicates that the user chooses to share the computational model to the macro base station to perform the computational task. In this paper, we construct a task-sharing decision model with the goal of global system consumption involving multiple smart mobile devices, as shown in equation (2).

$$\min \sum_{i=1}^m \{ \partial_i^t (d_{i,1} F_1 + d_{i,2} F_2) + \partial_i^e (d_{i,1} G_1 + d_{i,2} G_2) \}. \quad (2)$$

In this equation, F_1 and F_2 denote the total latency of local and shared to macro base station calculations, respectively, G_1 and G_2 denote the total energy consumption of local and shared to macro base station calculations, respectively. The task-sharing decision process is shown in Figure 2.

(1) Local Computational Model. The local computing power of different smart mobile devices held by users varies, and the computational delay $T_{i(z)}^P$ and computational energy

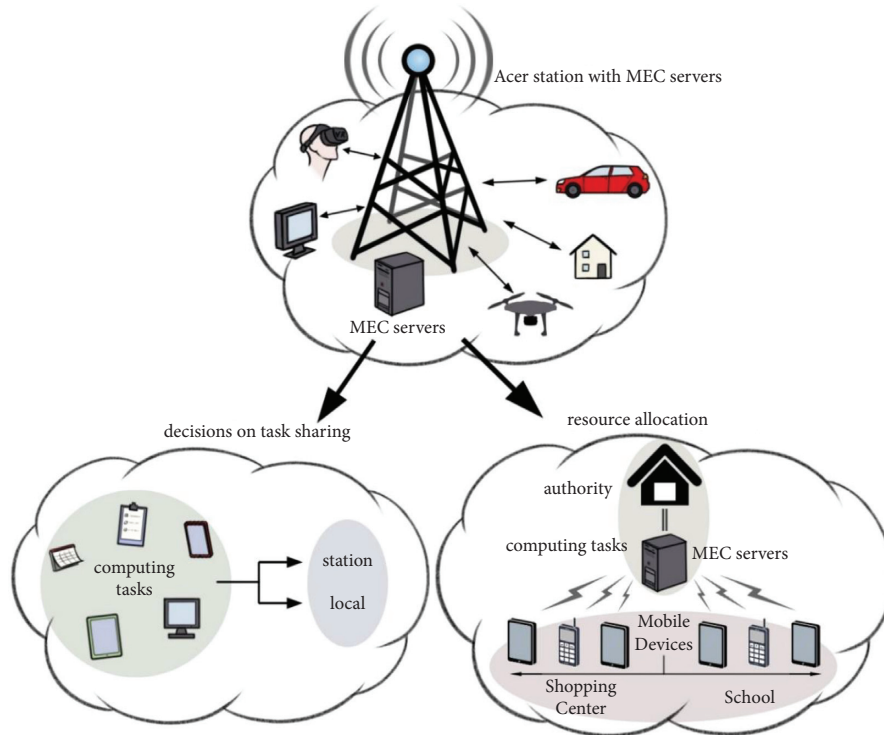


FIGURE 1: This diagram depicts the network model on which this paper is based.

TABLE 1: List of all acronyms used in the paper.

Acronym	Meaning
AR	Augmented reality
VR	Virtual reality
MILECR	Message importance based low energy consumption routing algorithm [28]
FCNS	Fuzzy routing-forwarding algorithm exploiting comprehensive node similarity [29]
DERV	The model we proposed in this paper
LODCO	The Lyapunov optimization-based dynamic computation offloading algorithm

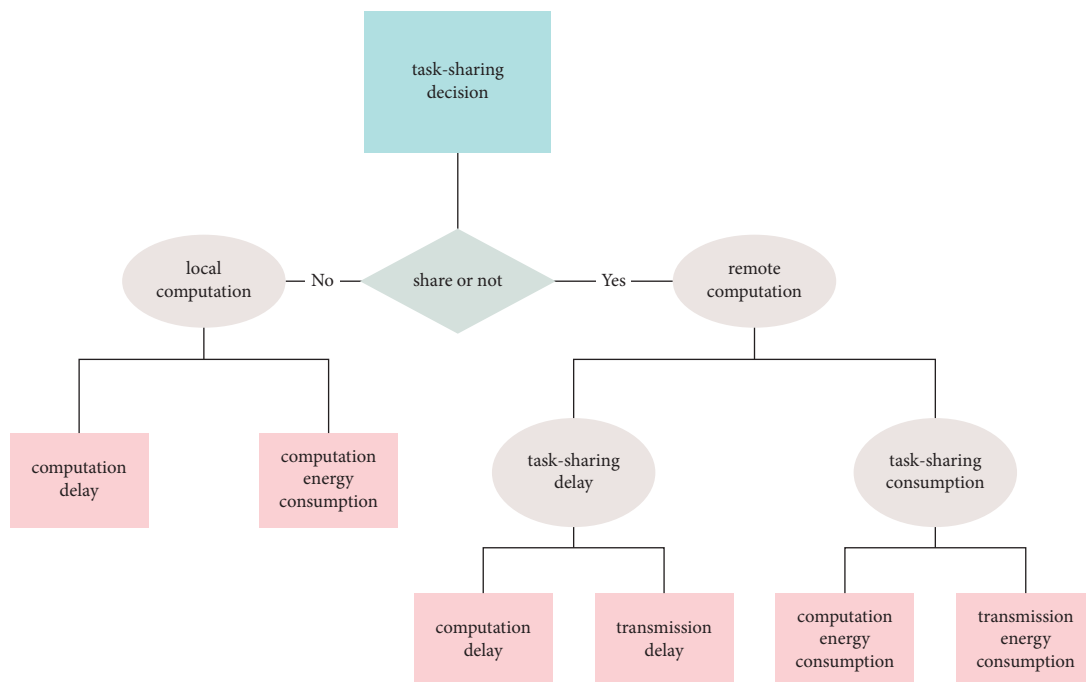


FIGURE 2: This diagram depicts the task-sharing decision-making process.

consumption $N_{i(z)}^P$ of the smart mobile device i when computing is

$$T_{i(z)}^P = \frac{B_i}{J_i^P}, \quad (3)$$

$$N_{i(z)}^P = \frac{B_i}{K_i^P}. \quad (4)$$

In this equation, J_i^P denotes the local computing power of the smart mobile terminal held by user I , and K_i^P denotes the energy consumption of this terminal i in a single local CPU cycle. For the local computing mode, since there is no other form of time consumption and energy consumption, equations (3) and (4) denote the total time delay and total energy consumption of user i 's local computing, respectively.

(2) *The Task-Sharing Model.* The MEC model constructed in this paper is a heterogeneous network with orthogonal frequency division multiplexing, where the channels between users accessing the same base station are orthogonal to each other, and there is only interference between accessing macro base stations. Therefore, the entire task-sharing computation process includes transmission and computation delays, and energy consumption includes transmission and execution energy consumption.

(1) Task Sharing Delay

We denote the data rate of user i choosing to access server h as

$$s_{i \rightarrow h}^M = H_{i \rightarrow h} \left(1 + \frac{P_i^h Q_i^h}{X_i^h + z_0} \right). \quad (5)$$

In this equation, H denotes the actual data transfer rate at which user i sends an uplink computation request to server h . P_i^h denotes the power between user i and server h . Q_i^h denotes the gain between terminal i and server h . X_i^h denotes the interference that exists between other users accessing the server and user i . z_0 denotes the background noise power.

The transmission latency for user i to share the computational task directly to the MEC server is

$$T_{i \rightarrow h}^P = \frac{A_i}{S_{i \rightarrow h}^M}. \quad (6)$$

The calculated delay in performing the completed tasks is

$$T_{i(z)}^P = \frac{B_i}{R_i^h}. \quad (7)$$

In this equation, R_i^h denotes the computational capacity of user i located at server h . Since there is no further form of delay in the entire computational task-sharing process, the total delay for user i to choose to share the computational task directly to the server is

$$T_i^P = \frac{A_i}{S_{i \rightarrow h}^M} + \frac{B_i}{R_i^h}. \quad (8)$$

Substituting the decision mechanism $d_{i,j}$ into the delay of the server shown in equation (6) yields

$$T_{i \rightarrow h}^P = \frac{A_i}{H_{i \rightarrow h} \left(1 + \left(P_i^h Q_i^h / \sum_{l=1, l \neq i}^M d_{l,2} P_l^h Q_l^h + z_0 \right) \right)}. \quad (9)$$

In this equation, l denotes the user accessing the server, P_l^h denotes the transmission power of l determined by server h based on some power control algorithm, and Q_i^h denotes the channel gain between user i and server h .

(2) Task-Sharing Energy Consumption

The energy consumption of user i directly sharing the computing task to the MEC server consists of both transmission energy $E_{i \rightarrow b}^P$ and execution energy $E_{i(z)}^P$, as shown in equations (10) and (11).

$$E_{i \rightarrow b}^P = \frac{P_i^h A_i}{T_{i \rightarrow h}^P}, \quad (10)$$

$$E_{i(z)}^P = B_i L_i^h. \quad (11)$$

In this equation, L_i^h represents the energy consumption of user i over a single CPU cycle of the macro base station.

Since there is no further form of energy consumption in the entire computational sharing process, the total energy consumption E_i^P that user i chooses to share the computational task directly to the server can be expressed as follows:

$$E_i^P = \frac{P_i^h A_i}{T_{i \rightarrow h}^P} + B_i L_i^h. \quad (12)$$

(3) Task-sharing Calculation

The MEC task-sharing computational model for joint consideration of latency and energy consumption is

$$\min \sum_{i=1}^M \left\{ \partial_i^t \left[d_{i,1} \frac{B_i}{R_i^L} + d_{i,2} \left(\frac{A_i}{H_{i \rightarrow h} (1 + (P_i^h Q_i^h / \sum_{l=1, l \neq i}^M d_{l,2} P_l^h Q_l^h + z_0))} + \frac{B_i}{R_i^h} \right) \right] \right. \\ \left. + \partial_i^e \left[d_{i,1} B_i L_i^L + d_{i,2} \left(\frac{P_i^h A_i}{H_{i \rightarrow h} (1 + (P_i^h Q_i^h / \sum_{l=1, l \neq i}^M d_{l,2} P_l^h Q_l^h + z_0))} + B_i L_i^h \right) \right] \right\}, \quad (13)$$

$$\text{s.t. } T_{i(z)}^L \leq T_i^{\max}, \quad (14)$$

$$T_i^N \leq T_i^{\max}, \quad (15)$$

$$\partial_i^T + \partial_i^E = 1, \dots, \partial_i^T, \quad \partial_i^E \in [0, 1], \quad (16)$$

$$d_{i,1} + d_{i,2} = 1, \dots, d_{i,1}, \quad d_{i,2} \in [0, 1]. \quad (17)$$

In this equation, the time delay $T_{i(z)}^L$ of the local execution method and the time delay T_i^N of the direct sharing of the computational task to the MEC server are given in the constraints, both of which are less than the maximum delay demand T_i^{\max} . The time demand factor ∂_i^T and the energy demand factor ∂_i^E in the trade-off mechanism take values in the range of $[0, 1]$ and their sum is 1. The decision mechanism $d_{i,1} = 1$, $d_{i,2} = 0$ means that user i chooses the local computational model to compute, $d_{i,1} = 0$, $d_{i,2} = 1$ means that user i chooses the task-sharing computational model to compute to the server.

3.2.2. Resource Allocation Model. Considering the problem of irrational resource allocation due to malicious competition caused by irregular behavior, this paper attempts to introduce a bargaining game model based on reputation value to achieve an overall optimal allocation of computational resources [33].

Assume that the MEC server computes resources to meet all demands, a MEC server receives compute share tasks from M users in one work cycle, and the MEC server has the compute capacity to perform K CPU cycles per second. We assume that E_i represents the computational resources allocated by the MEC server for the corresponding user i . Each user competes for the computational resources on the MEC server through a bargaining game process and gives greater priority to the one with the higher reputation value, then the resource allocation model [34] is

$$\max \prod (E_i - E_{i(\min)})^{\theta_i} \\ \text{s.t. } \sum_{i=1}^M E_i \leq E. \quad (18)$$

In this equation, $E_{i(\min)}$ represents the minimum computational resource allocated to user i by the MEC server, and θ_i represents the user's authoritative decision factor for the MEC server's computational resource, which is related to the user's current reputation value G_i , as shown in equation (19).

$$\theta_i = \frac{G_i}{\sum_{i=1}^M G_i}. \quad (19)$$

We adopt the Verifiable Caching Interaction Digest schema (VCID) proposed in Ref. [35] to get reputation value G_i . This mechanism is suitable for new applications generated by intelligent terminals in the Internet of Things environment, and has good convergence and scalability. It assesses credibility based on hosts' behavior. The calculation process is

$$G_i = \alpha T(i, O) + (1 - \alpha) T(O). \quad (20)$$

In this equation, $T(i, O)$ represents the direct credibility, its value depends on previous interaction experience. $T(O)$ represents the indirect credibility, and its value is based on the reputation of the authority O ; i and O can be gotten from the credit institutions. α is the factor used to regulate specific gravity. When G_i is less than a certain threshold, i will be rejected for authorization.

The resource allocation model can also be modeled as a benefit function for different reputation values, where the higher the reputation value, the larger the computational resources allocated. In order to maximize the benefits, this paper adopts the Lagrange multiplier method to solve the task-sharing resource allocation game model based on the reputation value in the MEC environment. Taking the logarithm of the resource allocation and introducing a Lagrange factor of τ , the Lagrange function is constructed, as shown in equation (20).

$$T = \left(\sum_{i=1}^M \theta_i \ln(E_i - E_{i(\min)}) \right) + \tau \left(\sum_{i=1}^M E_i - E \right). \quad (21)$$

4. Model Solution

The computational task-sharing decision model developed in Section 3 includes a task-sharing decision model and a resource allocation model, corresponding to the task-sharing decision problem and the resource allocation problem, respectively. The task-sharing decision problem is

non-deterministic polynomials (NP) in terms of mathematical modeling and needs to apply to multi-competition models when considering the search for the optimal mobile edge computing task-sharing scheme. Therefore, the PSO algorithm with high search accuracy and fast convergence speed is considered to solve the computational task-sharing model proposed in this paper for joint consideration of delay and energy consumption to achieve low delay and low energy consumption for computational task-sharing decisions [35]. At the same time, the amount of resources available to each server in the resource allocation problem is bounded, and the definition domain is within the computing resources of the base server. According to the theory of large values of functions with bounded definition domains, if large values of resource allocation functions exist, it can only be obtained at the point of maximum value within the definition domain or the boundary point of the definition domain, and the Lagrange multiplier method is an effective way to determine whether the function has a point of maximum value or not [36]. Therefore, to magnify the benefit function for smart mobile terminals with different reputation values, this paper uses the Lagrange multiplier method [37] to solve the task-sharing resource game theory allocation model to achieve a rational and reliable optimal resource allocation.

4.1. Task Sharing Decision Solving based on Improved PSO Algorithm

4.1.1. PSO Algorithm. The PSO algorithm is a population-based intelligent stochastic optimization algorithm based on the collaborative search for food by a flock of birds. The PSO algorithm randomly generates M particles to form a population U in an N -dimensional search space, $U = \{L_1, L_2, \dots, L_M\}$, each particle represents a potential solution, the position of the k th particle is denoted as a vector $L_k = \{l_{k1}, l_{k2}, \dots, l_{kN}\}$, the velocity is denoted as a vector $V_k = \{v_{k1}, v_{k2}, \dots, v_{kN}\}$, the k th particle search to the most available position during the search process is denoted as $L_{best_k} = \{l_{best_{k1}}, l_{best_{k2}}, \dots, l_{best_{kN}}\}$, and the optimal position searched by the population of particles during the global search process is denoted as $G_{best} = \{g_{best_1}, g_{best_2}, \dots, g_{best_N}\}$, the k th particle's position and velocity updates are calculated as

$$v_{kd}(t+1) = \omega v_{kd}(t) + x_1 \text{rand}() [l_{best_{kd}}(t) - l_{kd}(t)] + x_2 \text{rand}() [g_{best_d}(t) - l_{kd}(t)], \quad (22)$$

$$l_{kd}(t+1) = l_{kd}(t) + v_{kd}(t+1), \quad (23)$$

$$1 \leq k \leq M, 1 \leq d \leq N. \quad (24)$$

In this equation, x_1 and x_2 are acceleration factors, indicating the extent to which particles are affected by individual values and social cognition, respectively. rand indicates the random number in $[0, 1]$. ω indicates the inertia weight, is nonnegative, and used to adjust the search scope of the solution space. t indicates the number of iterations.

4.1.2. Improved PSO Algorithm. In this paper, the proposed MEC task-sharing model solves the multi-participant NP problem, where the system consumption generated by n end-users choosing different task-sharing methods is different, but considering that the consumption values are not too different, the total value of system consumption generated by different combinations of task sharing methods does not change much and is more stable. The PSO algorithm with fast convergence is used to solve the problem. It is prone to cause the consumption of the system in high-dimensional states with multiple users involved to fall into a local optimum, so the algorithm is improved in terms of economic and exploratory capabilities, respectively. In terms of economy, in the optimization process, the inertia weight in the model is adaptively and dynamically adjusted to improve the local and global optimization capabilities of the algorithm to obtain better solution quality. In terms of exploration ability, from the perspective of mutation, the acceleration factor in the particle swarm is optimized to improve the exploration ability of the algorithm in the solution space.

(1) Inertia Weighting Improvement. The MEC offloading decision-making model in this paper is oriented to the multi-task decision-making process of multiple smart mobile terminals. The PSO algorithm tends to fall into the precocious convergence of locally optimal solutions when solving high dimensional functions of system consumption under multitasking. Using a single adjustment method where the inertial weights are kept constant or linearly decreasing, it is difficult to guarantee that every dimension of the particle tends to be optimal at the same time, making the probability of simultaneously searching for an optimal solution in every dimension of the system consumption function under multiple participation very small. Therefore, in this paper, an adaptive nonlinear dynamic approach is used to find optimality, and a cosine function is introduced based on previous studies, as shown in equation (24).

$$\omega = \frac{\omega_{\max} + \omega_{\min}}{2} \cos\left(\frac{\pi t}{T_{\max}}\right) + \frac{\omega_{\max} - \omega_{\min}}{2}. \quad (25)$$

In this equation, ω_{\max} indicates the maximum value of the inertia weight, which is generally taken as 0.7, ω_{\min} indicates the very small value of the weight, which is generally taken as 0.1, and T_{\max} indicates the maximum number of iterations. Then, the value of (t/T_{\max}) is between 0 and 1. And, because the cosine function is monotonically decreasing over the interval $[0, \pi]$, ω is going to increase as t increases.

The weight of inertia controls the influence of the historical position of particles on the current search state, and maintains the balance between global search and local search, which can effectively achieve adaptive nonlinear adjustment of its value and improve the efficiency and intelligence of the algorithm. In the early stage of the algorithm, to increase the global search, the inertia weights should have a larger value; in the late algorithm, they should maintain a reasonable convergence rate to increase the local

search capability, so the inertia weights should be kept smaller.

(2) *Improvement of the Acceleration Factor.* Since particle position is influenced by both individual and population extremes, the cognitive part and social part could greatly affect the direction and velocity of particle convergence. The acceleration factor x_1 in the PSO algorithm usually takes a value of 0.43 and x_2 usually takes a value of 0.4. But, in this paper's MEC task-sharing decision-oriented model, when the number of iterations varies, there is a bias about the leading position between the cognitive part and social part. When the number of iterations is small, the cognitive part represented by x_1 plays a dominant role, and the social part represented by x_2 plays a secondary role. When the number of iterations is large, the accumulation of social knowledge continues to increase, just the opposite. This paper faces dynamic multi-local computation problems and server computation selection problems in the MEC task-sharing decision. To further improve the PSO algorithm's ability to explore the solution space, we improved the acceleration factor from a variation perspective. The dynamic acceleration factor is used instead of the static acceleration factor as shown in equations (25) and (26).

$$x_1 = \frac{\alpha}{t}, \quad (26)$$

$$x_2 = \beta t^2. \quad (27)$$

In this equation, t indicates the number of iterations. α indicates the individual cognitive impact factor, which takes values ranging from 149 to 280 in this scenario. β indicates the social cognitive impact factor, which takes values ranging from 0.00013 to 0.000205 in this scenario.

4.1.3. Task Sharing Decision Method based on Improved PSO Algorithm. The mapping of the solution space of the improved PSO algorithm to the task-sharing decision problem is shown in Table 2.

The task-sharing decision-solving process based on the improved PSO algorithm is shown in Algorithm 1.

Algorithm 1 The task sharing decision-solving process based on the improved PSO algorithm.

Step 1. Initialize: Spatial dimension $N = 50$ and population size $M = 100$ were determined; inertial weight extremes ω_{\max} and ω_{\min} were set to 0.7 and 0.1, respectively; and initial position and initial velocity were randomly generated within the search space.

Step 2. Find the adaptation value. Calculate the total consumption of the system based on the adaptation function shown in equation (13).

Step 3. Find the individual extreme value L_{best_k} and the population extreme value G_{best} . Find the minimal value L_{best_k} of system consumption resulting from choosing different offloading methods and the optimal value G_{best} of system consumption for all different combinations of offloading methods.

TABLE 2: The table describes the solution space of the improved PSO algorithm with respect to the task sharing decision problem.

Improved PSO algorithm	Task-sharing decision-making issues
Spatial dimension	Quantity
Stocks	Pooling of different task-sharing decisions
Particle location	Different task-sharing decisions
Fitness value	Total system consumption

Step 4. Update the particle position and velocity. Update the velocity and position according to equations (21) to (26). Each particle shares the same distance from the current system eigenvalue. Particle k selects the nearest consumption target to the system; the other particles will vectorize their positions relative to particle k and their preferred positions away from the small system energy target.

Step 5. Update the individual polar value L_{best_k} and the group polar value G_{best} . Compare the system consumption value of each particle to L_{best_k} . Replace L_{best_k} with the current position if the current consumption value is smaller than L_{best_k} . Then, compare the minimum system consumption position in L_{best_k} with G_{best} . Replace G_{best} if the system consumption value in L_{best_k} is smaller than G_{best} .

Step 6. Determine the termination condition. The termination condition is judged by the number of iterations of the algorithm. If the termination condition is satisfied, exit the loop and return the superior search result G_{best} . If the termination condition is not satisfied, repeat steps 2 to 5 until the termination condition is satisfied.

Step 7. Outputs the optimal solution, which is the small value consumed by the system.

Since the resource allocation problem is NP-hard, we use a heuristic algorithm to solve it. It can be seen from the pseudo-code of the algorithm that the running time depends on the number of mobile devices participating in the allocation and the number of iterations set. The time complexity is $O(n^2)$, but since n here is generally not a large number, the running time is acceptable.

4.2. Resource Allocation Solution based on Lagrange Multiplier Method. The resource allocation model is aimed at the benefit function of smart mobile terminals with different reputation values, and the amount of resources available to each smart mobile terminal is not greater than the computing resources of the MEC server. To maximize the benefits of MEC resources, the maximum value point of the benefit function is only obtained within the bounded range or boundary point of the computing resource. We know that the Lagrangian multiplier method is an effective method to determine whether the benefit function has the maximum value of computing resources. So, this paper adopts the Lagrangian multiplier method to solve the

Input: acceleration factors x_1, x_2 , inertia weight ω , iterations t_0 , task-sharing decision model mentioned above
Output: the minimum of total system consumption of task-sharing decision model

Begin

- (1) $N = 50, M = 100$ //spatial dimension, population size
- (2) $\omega_{\min, \max}, t = 0$
- (3) $fitness(x) \leftarrow$ Equation (13)//take total system consumption as the fitness function
- (4) **For** each particle k
- (5) Initialize velocity V_k and position L_k for particle k randomly
- (6) evaluate particle k and set $L_{best_k} = L_k$
- (7) **End for**
- (8) **While** $t \leq t_0$
- (9) **For** $k = 1$ to 100
- (10) Update the velocity and position of particle i according to equations (21), (22), (25), and (26)
- (11) Evaluate particle i
- (12) if $fitness(L_k) < fitness(L_{best_k})$
- (13) $L_{best_k} \leftarrow L_k$
- (14) if $fitness(L_{best_k}) < fitness(G_{best})$
- (15) $G_{best} \leftarrow L_{best_k}$
- (16) **End for**
- (17) **End while**
- (18) print G_{best}

Stop

ALGORITHM 1: Task-sharing decision model using improved PSO algorithm.

abovementioned unloading resource game allocation model based on reputation value. Take the logarithm of the resource allocation model and introduce the Lagrangian factor τ to construct the Lagrangian function, as shown in equation (20).

Using the one-time bias derivative of T for τ as 0, the one-time bias derivative can determine the locally optimal feasible solution, yielding equation (27).

$$\tau = \frac{-1}{E - \sum_{i=1}^M E_{i(\min)}}. \quad (28)$$

The quadratic partial derivative of t is then performed to obtain equation (28).

$$\frac{\partial^2 T}{\partial \tau^2} = \frac{1}{\tau^2}. \quad (29)$$

Since the quadratic partial derivative of τ is greater than zero, it can be determined that there is a most-valued solution to the benefit function.

Substitute equation (27) into equation (20) to obtain equation (29).

$$E_i = E_{i(\min)} + \theta_i \left(E - \sum_{i=1}^M E_{i(\min)} \right). \quad (30)$$

Ultimately, the MEC server allocates computational resources according to the results of equation (29), performs different computational task-sharing tasks, and reports the results of the computation back to the end-user.

The process of solving the resource allocation based on the Lagrange multiplier method is shown in Algorithm 2.

Algorithm 2. The process of solving the resource allocation based on the Lagrange multiplier method.

Step 1. Taking the logarithm of the target function for users with different reputation values and introducing the Lagrange factor τ , then construct the Lagrange function T

Step 2. Perform a first partial derivation of the constructed benefit function T about the computational resource requirements E_i of each MEC server to obtain a feasible solution for the local superiority of the computational resources. Then, perform a second partial derivation, and if the second partial derivation is not equal to 0 then prove that the benefit function has a valued solution.

Step 3. Make the value of the bias derivative of the benefit function T with respect to the computational resource requirement E_i for each MEC server equal to 0, and solve for the stationary point $E_i = E_{i(\min)}$.

Step 4. Obtain a bias derivative of the benefit function T with respect to the Lagrange factor τ once.

$$\frac{\partial T}{\partial \tau} = -\frac{1}{\tau} + \sum_{i=1}^M E_{i(\min)}. \quad (31)$$

Step 5. From the formula (28), it can be seen that the benefit function exists as the most valuable solution. Then, make the value of the benefit function T on the Lagrange factor τ of the first partial derivative equal to 0, the solution to the value of τ as shown in formula (27).

$$E_i = E_{i(\min)} + \frac{G_i}{\sum_{i=1}^M G_i} \left(E - \sum_{i=1}^M E_{i(\min)} \right). \quad (32)$$

Algorithm 2 can be described in Algorithm 2. It can be seen that the time complexity of Algorithm 2 is $O(n)$.

Input: the reputation value G_i , the minimum computational resource $E_{i(\min)}$ of every device i , the total computational resource of the MEC server E

Output: the deserved computational resource E_i of every device i

- (1) **Begin**
- (2) **For** $i = 1$ to M
- (3) $\theta_i = (G_i / \sum_{i=1}^M G_i)$ // Calculate the authoritative decision factors
- (4) **End for**
- (5) $T = (\sum_{i=1}^M \theta_i \ln(E_i - E_{i(\min)})) + \tau (\sum_{i=1}^M E_i - E)$ // Construct Lagrangian function
- (6) **For** each device i requesting for resource
- (7) Calculate the partial derivative of T with respect to E_i
- (8) **If** $(\partial^2 T / \partial \tau^2) > 0$
- (9) **For** $i = 1$ to M
- (10) $E_i = E_{i(\min)} + (G_i / \sum_{i=1}^M G_i) (E - \sum_{i=1}^M E_{i(\min)})$
- (11) **End for**
- (12) **End if**
- (13) Print $E_i (i = 1 \text{ to } M)$
- Stop

ALGORITHM 2: Resource allocation based on the Lagrange multiplier method.

5. Experimental Simulation

In this paper, we use ONE, an open-source simulation tool, to evaluate the performance of DERV. We compare the packet transfer ratio, average end-to-end delay [38], routing overhead [39], and energy consumption with the FCNS, SCR, MILECR, and epidemic algorithm [40] to verify the advantages of DERV in terms of system overhead. In addition, we considered the effects of three different schemes on the overall energy consumption of the model [41], namely, low energy task sharing, randomly assigned task-sharing [42], and task sharing with joint delay and energy consumption.

5.1. Parameter Settings. To follow the characteristics of massive data transfer in a short period in a 5G environment, this paper uses real datasets downloaded from CRAWDDAD to drive node activities, and datasets from Infocom 5, Infocom 6, Cambridge, and Intel are selected for simulation [43]. The specific simulation environment parameter settings are shown in Table 3.

The number of nodes set up in this experiment is 50, the node computing power is randomly assigned to 500–1000 MHz/s, and the computing power of the server is assigned to 5 GHz/s. The specific physical model parameter settings are shown in Table 4.

5.2. Metrics. Metrics are used to measure the performance of algorithms in opportunistic complex social networks. Four are selected in this paper, they are packet transfer ratio, average end-to-end delay, routing overhead, and overall energy consumption. Here is a brief introduction of what they mean.

- (1) Packet transfer ratio: the ratio of the number of packets received at the destination to the number of packets in recent years, with the deep integration ts sent from the source.

- (2) Average end-to-end delay: the average delay in data transmission between two nodes.
- (3) Routing overhead: the total size of routing packets sent for maintenance and also for route discovery.
- (4) Energy consumption: ALL the energy used in data transmission between two nodes.

5.3. Analysis of Results. As the simulation time increases, the clustered bar graphs in Figures 3(a)–3(d) show the packet transfer ratios of the DERV, FCNS, SCR, MILECR, and epidemic algorithms. When the simulation time is short, the performance advantage of the DERV algorithm is not as pronounced as the other four algorithms, but as the simulation time increases, the success rate of the DERV algorithm is significantly higher than that of the other four algorithms. This is because our solution fully considers the problem of nodes maliciously competing for computational resources, and introduces the measure of reputation, where nodes with a higher reputation are allocated more computational resources, thus achieving an overall optimal allocation of computational resources.

Figures 4(a)–4(d) show the comparison results of the average end-to-end delay for the DERV, FCNS, SCR, MILECR, and epidemic algorithms as the simulation time increases. Among the comparison schemes, the DERV strategy has the lowest average end-to-end latency and this advantage becomes more pronounced with the increase in the simulation time. This is because the DERV algorithm introduces delay and energy consumption trade-off factors, enables full sharing of multitasking computational resources across multiple smart mobile devices, and optimizes the routing strategy for data transfer more efficiently than the other schemes. In contrast, regarding the epidemic algorithm, a large number of copies of information are generated during data transfer, which will lead to an increase in forwarding delay. In addition, SCR and MILECR algorithms use the strategy of neighbor node cooperative transmission,

TABLE 3: The parameter settings for a specific simulation environment.

Dataset	Infocom5	Infocom6	Cambridge	Intel
Device	iMote	iMote	iMote	iMote
Duration (day)	3.5	4	11.5	4
Number of experimental devices	41	98	52	9
Number of internal contacts iMote	22459	170601	10873	1364
Number of nodes	41	98	52	9
Buffer size(M)	5	5	5	5
TTL	60 min	60 min	2 days	0.5 day

TABLE 4: Specific physical model parameter settings.

Parameters	Value
Node's computing power (GHz)	[0.5,1.0]
Node's reputation value	[0,10]
Time requirement factor	[0,1]
Acer station's computing power (GHz)	10
Power (mW)	100
Gain between nodes and station	10^{-6}
Background noise power (dBm)	-100
The actual bandwidth of the uplink calculation request (kHz)	15
Calculating task data maximum (kB)	5000
Individual CPU power consumption of the station (W)	5
Total CPU required for the task (mc)	1000

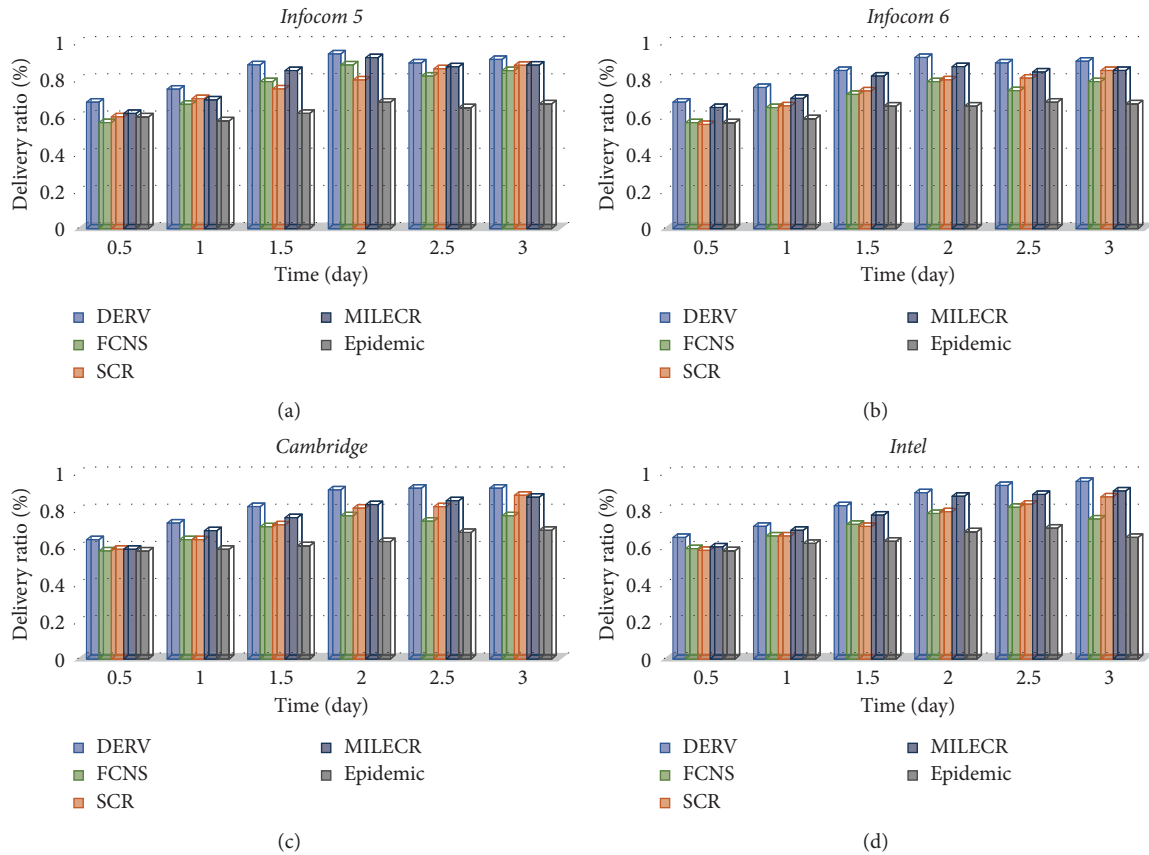


FIGURE 3: (a)–(d) The packet transfer ratios of the DERV, FCNS, SCR, MILECR, and epidemic algorithms on datasets Infocom5, Infocom6, Cambridge, Intel.

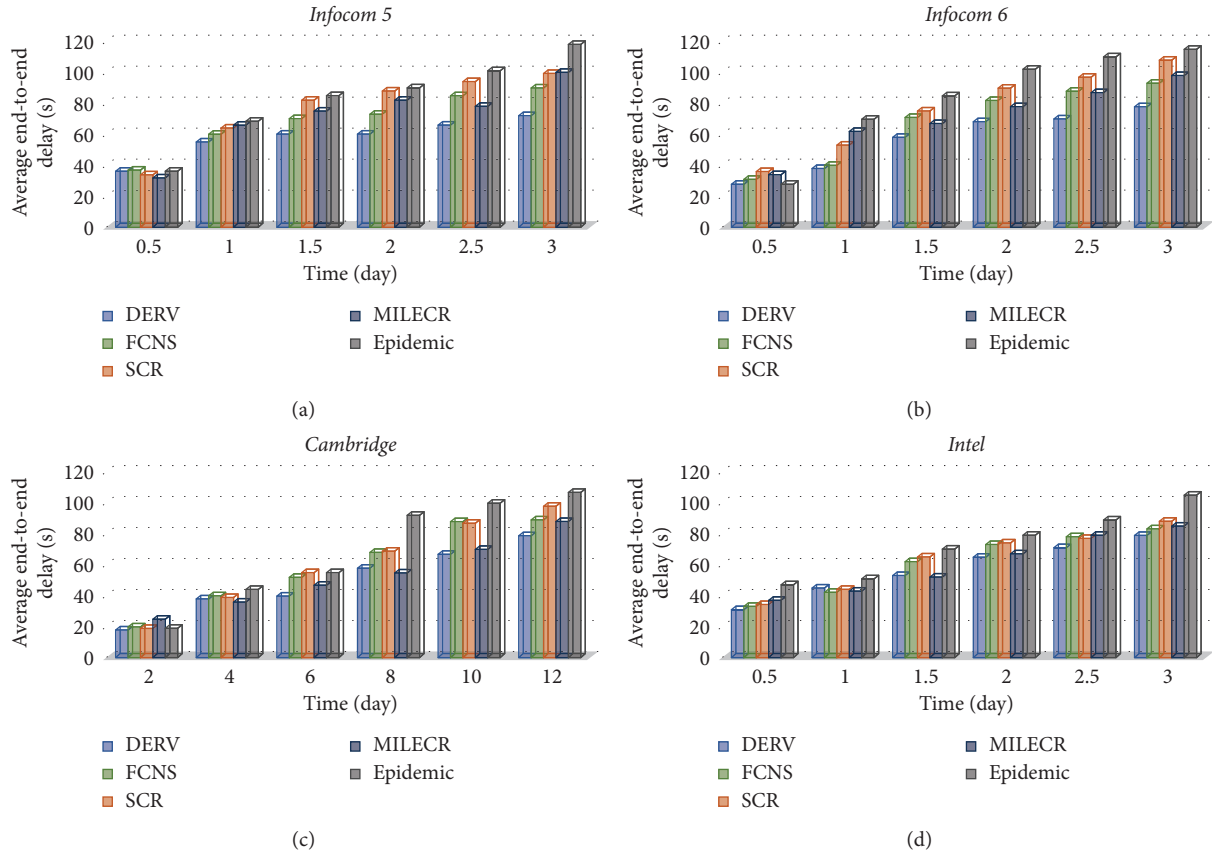


FIGURE 4: (a–d) The average end-to-end delay for the DERV, FCNS, SCR, MILECR, and epidemic algorithms on datasets Infocom5, Infocom6, Cambridge, Intel.

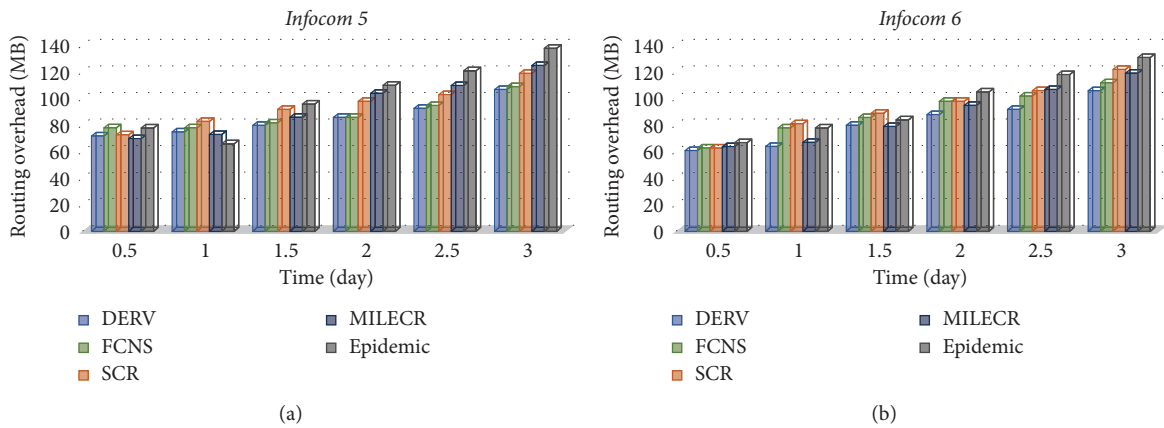


FIGURE 5: Continued.

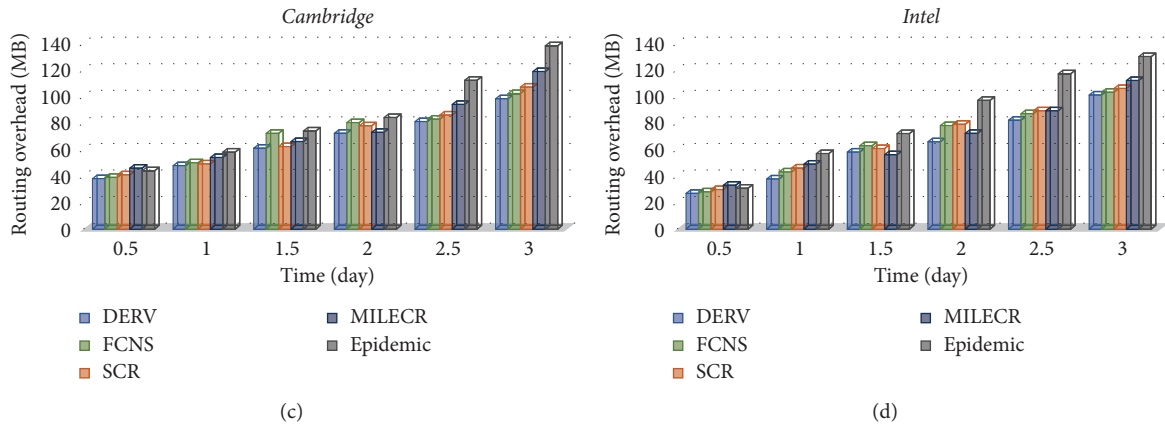


FIGURE 5: (a, b) The routing overhead of the DERV, FCNS, SCR, MILECR, and epidemic algorithms on datasets Infocom5, Infocom6, Cambridge, Intel.

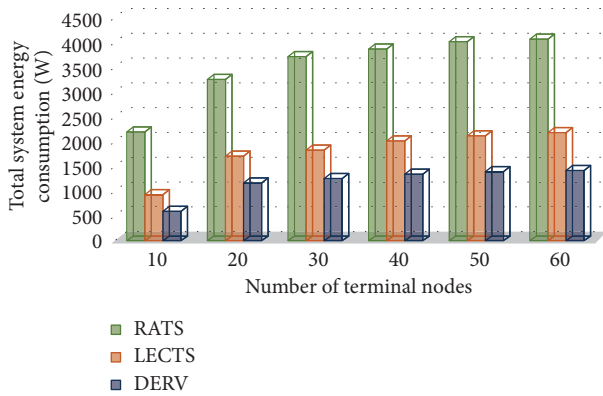


FIGURE 6: The overall energy consumption of three different schemes.

which reduces the impact of node cache on message transmission, but still has high latency when the experiment is longer. FCNS algorithm is more affected by node cache and has poorer performance when the node cache space is smaller. In summary, the DERV algorithm is the best method to improve the average end-to-end delay performance in high-speed communication scenarios compared to other algorithms.

Figures 5(a) and 5(b) show the comparison results of the routing overhead of the DERV, FCNS, SCR, MILECR, and epidemic algorithms when the simulation time increases. DERV can better predict the next-hop node compared to other models. By sending messages to nodes that satisfy the reputation value in the communication domain, the routing cost of sending messages to other noncooperating nodes can be effectively reduced. In addition, nodes do not need to use a computational model for continuous computation and decision-making during message transmission. This can reduce the time and routing resource costs. In the case of the epidemic algorithm, a large number of redundant message copies require time and computational resources, and the routing overhead is significantly higher than that of other algorithms. For SCR and MILECR algorithms, the

cooperative mechanism facilitates the rational allocation of computational resources, so the cost of these two algorithms is at an intermediate level. The FCNS algorithm takes into account the mobile similarity of nodes, but does not fully consider the transmission preferences of nodes, so its performance is poorer than the DERV algorithm. Compared to the results, DERV outperforms the other four models in terms of routing overhead.

Figure 6 shows a comparison of the overall energy consumption of three different schemes, namely, low energy task sharing (LETS), randomly assigned task sharing (RATS), and task sharing with joint time delay and energy consumption in this paper. As shown in the figure, as the number of end nodes increases, the total energy consumption of the system under all three schemes increases, while the total energy consumption of the scheme proposed in this paper is always at the lowest level.

6. Conclusion

This paper proposes a task-sharing model (DERV) based on reputation value, which solves the problems of high latency, high energy consumption, and low reliability faced by computing sharing in emerging mobile applications in the big data environment. DERV divides the computing sharing task in the opportunistic complex IoT environment into two processes: shared decision-making and resource allocation. The experimental results show that the computing task-sharing model proposed in this paper can meet the service requirements of low latency, low energy consumption, and high reliability in emerging intelligent applications, and can effectively realize the overall optimized configuration of computing shared resources. Among them, the main innovations and contributions of this article are as follows:

- (1) This article focuses on providing low-latency, low-energy consumption, and high-reliability service quality guarantees for time-delay and energy-sensitive computing-intensive smart mobile terminals in a big data environment. We propose a network

model composed of users, MEC servers, and trusted institutions.

- (2) To realize the multitasking computational task-sharing scheme for multiple smart mobile terminals, we propose a task-sharing decision model that takes into account both delay and energy consumption. The model uses an improved PSO algorithm to achieve an optimal computing task-sharing scheme with low latency and low energy consumption.
- (3) Aiming at the problem of unreasonable resource allocation caused by malicious competition for resources due to irregular behavior in the IoT system, this paper proposes a bargaining game model based on reputation value, which mainly uses the Lagrangian multiplier method to realize the calculation Reliable allocation of resources.

In the future, with the increase in the computing power of mobile devices in opportunistic complex social networks, the DERV model proposed in this paper can be applied to the transmission environment of 5G and big data networks. We will collect larger real datasets in social scenarios and explore ways to improve information transmission performance.

Data Availability

The data used to support the findings of this study are currently under embargo while the research findings are commercialized. Requests for data, 12 months after publication of this article, will be considered by the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by The National Natural Science Foundation of China (61762051 and 82160955).

References

- [1] S. N. Moosavi and V. Pourahmadi, "Opportunistic multiple access (OMA) for crowdsensing networks with sparse activation," *Trans Emerging Tel Tech*, vol. 30, Article ID e3559, 2019.
- [2] J. Wu, Z. Chen, and M. Zhao, "An efficient data packet iteration and transmission algorithm in opportunistic social networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 3141–3153, 2020.
- [3] J. Wu, Z. Chen, and M. Zhao, "Community recombination and duplication node traverse algorithm in opportunistic social networks," *Peer-to-Peer Networking and Applications*, vol. 13, pp. 940–947, 2020.
- [4] J. Lorincz, A. Capone, and J. Wu, "Greener, energy-efficient and sustainable networks: state-of-the-art and new trends," *Sensors*, vol. 19, no. 22, Article ID 4864, 2019.
- [5] G. Yu and J. Wu, "Content caching based on mobility prediction and joint user Prefetch in Mobile edge networks," *Peer-to-Peer Networking and Applications*, vol. 13, pp. 1839–1852, 2020.
- [6] J. LUO, W. U. Jia, and Y. WU, "Advanced data delivery strategy base on multi-perceived community with IoT in social complex networks," *Complexity*, vol. 2020, Article ID 3576542, 20 pages, 2020.
- [7] Y. A. N. G. Weiyu, W. U. Jia, and J. LUO, "Effective date transmission and control base on social communication in social opportunistic complex networks," *Complexity*, vol. 2020, Article ID 3721579, 13 pages, 2020.
- [8] M. S. Abdalzaher and O. Muta, "A game-theoretic approach for enhancing security and data trustworthiness in IoT applications," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 11250–11261, 2020.
- [9] W. U. Jia, C. H. A. N. G. Liu, and Y. U. Genghua, "Effective data decision-making and transmission system based on mobile health for chronic diseases management in the elderly," *IEEE Systems Journal*, 2020.
- [10] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016.
- [11] H. Zhang, J. Guo, L. Yang, X. Li, and H. Ji, "Computation offloading considering fronthaul and backhaul in small-cell networks integrated with MEC," in *Proceedings of the 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPs)*, pp. 115–120, Atlanta, GA, USA, May 2017.
- [12] Y. Lu, L. Chang, and J. Luo, "Routing algorithm based on user adaptive data transmission scheme in opportunistic social networks," *Electronics*, vol. 10, Article ID 1138, 2021.
- [13] J. Li, H. Gao, T. Lv, and Y. Lu, "Deep reinforcement learning based computation offloading and resource allocation for MEC," in *Proceedings of the 2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, Barcelona, Spain, April 2018.
- [14] Y. Nan, W. Li, W. Bao et al., "Adaptive energy-aware computation offloading for cloud of things systems," *IEEE Access*, vol. 5, pp. 23947–23957, 2017.
- [15] Y. L. Teng, W. Liu, W. P. Ouyang, L. I. Kun, and M. Song, "Queue-aware joint optimization of offloading and transmission in wireless mobile edge computing systems," *Journal of Beijing University of Posts and Telecommunications*, vol. 42, 2019.
- [16] H.-H. Chang, W.-Y. Chiu, H. Sun, and C.-M. Chen, "User-centric multiobjective approach to privacy preservation and energy cost minimization in smart home," *IEEE Systems Journal*, vol. 13, no. 1, pp. 1030–1041, 2019.
- [17] D. Waters, A. Donnellan, and J. Fox, "An adaptable internet of things network infrastructure implemented for a smart building system," in *Proceedings of the 2021 32nd Irish Signals and Systems Conference (ISSC)*, pp. 1–7, Athlone, Ireland, June 2021.
- [18] K. Mamo, J. I. Nieto, R. Buenrostro, and M. Z. Ali, "Spectrum based power management for congested iot networks," *Sensors*, vol. 21, Article ID 14, 2021.
- [19] K. Zhang, Y. Zhu, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Deep learning empowered task offloading for mobile edge computing in urban informatics," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7635–7647, 2019.
- [20] B. Huang, Y. Li, Z. Li et al., "Security and cost-aware computation offloading via deep reinforcement learning in mobile edge computing," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 3816237, 20 pages, 2019.

- [21] Y. Dong, L. Chang, J. Luo, and W. Jia, "A routing query algorithm based on time-varying relationship group in opportunistic social networks," *Electronics*, vol. 10, Article ID 1595, 2021.
- [22] L. I. Xiaoli and W. U. Jia, "Node-oriented secure data transmission algorithm based on IoT system in social networks," *IEEE Communications Letters*, vol. 24, 2020.
- [23] Z. Fang, L. Chang, J. Luo, and W. Jia, "A data transmission algorithm based on triangle link structure prediction in opportunistic social networks," *Electronics*, vol. 10, Article ID 1128, 2021.
- [24] Q. Han, H. Wen, J. Wu, and M. Ren, "Rumor spreading and security monitoring in complex networks," in *Computational Social Networks 2015*, M. Thai, N. Nguyen, and H. Shen, Eds., Springer, Berlin, Germany, Springer, 2015 Lecture Notes in Computer Science.
- [25] Y. Deng, F. Gou, and J. Wu, "Hybrid data transmission scheme based on source node centrality and community reconstruction in opportunistic social network," *Peer-to-Peer Networking and Applications*, vol. 14, pp. 1–13, 2021.
- [26] J. Wu, S. Yin, Y. Xiao, and G. Yu, "Effective data selection and management method based on dynamic regulation in opportunistic social networks," *Electronics*, vol. 9, Article ID 1271, 2020.
- [27] Y. Xiao and J. Wu, "Data transmission and management based on node communication in opportunistic social networks," *Symmetry*, vol. 12, no. 8, Article ID 1288, 2020.
- [28] Y. I. N. Sheng, W. U. Jia, and Y. U. Genghua, "Low energy consumption routing algorithm based on message importance in opportunistic social networks," *Peer-to-Peer Networking and Applications*, vol. 14, pp. 1–14, 2020.
- [29] W. U. Jia, Q. U. Jingge, and Y. U. Genghua, "Behavior prediction based on interest characteristic and user communication in opportunistic social networks," *Peer-to-Peer Networking and Applications*, vol. 14, no. 2, pp. 1006–1018, 2021.
- [30] J. Wu, W. Zou, and H. Long, "Effective path prediction and data transmission in opportunistic social networks," *IET Communications*, vol. 15, pp. 2202–2211, 2021.
- [31] Y. A. N. G. Weiyu, J. LUO, and W. U. Jia, "Application of information transmission control strategy based on incremental community division in IoT platform," *IEEE Sensors Journal*, vol. 21, no. 19, pp. 21968–21978, 2021.
- [32] F. Gou and J. Wu, "Triad link prediction method based on the evolutionary analysis with IoT in opportunistic social networks," *Computer Communications*, vol. 181C, pp. 143–155, 2022.
- [33] W. Lin, X. Zhang, L. Qi, W. Li, and S. Nepal, "Location-aware service recommendations with privacy-preservation in the internet of things," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 99, pp. 1–9, 2020.
- [34] S. Li, K. K. R. Choo, Q. Sun, W. J. Buchanan, and J. Cao, "IoT forensics: amazon echo as a use case," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6487–6497, 2019.
- [35] X. Xu, X. Zhang, M. Khan, W. Dou, S. Xue, and S. Yu, "A balanced virtual machine scheduling method for energy-performance trade-offs in cyber-physical cloud systems," *Future Generation Computer Systems*, vol. 105, pp. 789–799, 2020.
- [36] Y. Yuan, X. Zhang, and J. Tang, "Guest editorial special issue on privacy and security in computational intelligence," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, pp. 590–592, 2020.
- [37] L. Qi, Y. Chen, Y. Yuan, S. Fu, X. Zhang, and X. Xu, "A QoS-aware virtual machine scheduling method for energy conservation in cloud-based cyber-physical systems," *World Wide Web*, vol. 23, 2020.
- [38] X. Xu, X. Zhang, H. Gao, Y. Xue, and W. Dou, "BeCome: blockchain-enabled computation offloading for IoT in mobile edge computing," *IEEE Transactions on Industrial Informatics*, vol. 16, 2019.
- [39] W. Tang, C. Wu, L. Qi, X. Zhang, X. Xu, and W. Dou, "A WiFi-aware method for mobile data offloading with deadline constraints," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 7, pp. 1–15, Article ID e5318, 2021.
- [40] S. Li, S. Zhao, P. Yang, P. Andriotis, L. Xu, and Q. Sun, "Distributed consensus algorithm for events detection in cyber-physical systems," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2299–2308, 2019.
- [41] Y. Yuan and W. Banzhaf, "Making better use of repair templates in automated program repair: a multi-objective approach," *Evolution in Action: Past, Present and Future*, Springer, Berlin, Germany, 2020.
- [42] X. Zhao, W. Dou, X. Yin, H. Wang, and L. Qi, "Edge computing-enabled deep learning for real-time video optimization in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 99, p. 1, 2020.
- [43] W. Dou, W. Tang, B. Liu, X. Xu, and Q. Ni, "Blockchain-based mobility-aware offloading mechanism for fog computing services," *Computer Communications*, vol. 164, no. 6, pp. 261–273, 2020.