FairEdge: A Fairness-Oriented Task Offloading Scheme for Iot Applications in Mobile Cloudlet Networks

SHUANG LAI [1], **XIAOCHEN FAN** [2], **(Student Member, IEEE), QIANWEN YE** [2,3],
ZHIYUAN TAN [4], **YUANFANG ZHANG** [2,5], **XIANGJIAN HE** [2], **(Senior Member, IEEE),**
AND PRIYADARSI NANDA [2]

[1]School of Humanities, Economics and Law, Northwestern Polytechnical University, Xi'an 710072, China
[2]School of Electrical and Data Engineering, University of Technology Sydney, Sydney, NSW 2007, Australia
[3]School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China
[4]School of Computing, Edinburgh Napier University, Edinburgh EH10 5DT, U.K.
[5]School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China

Corresponding author: Xiaochen Fan (xiaochen.fan@student.uts.edu.au)

ABSTRACT Mobile cloud computing has emerged as a promising paradigm to facilitate computation-intensive and delay-sensitive mobile applications. Computation offloading services at the edge mobile cloud environment are provided by small-scale cloud infrastructures such as cloudlets. While offloading tasks to in-proximity cloudlets enjoys benefits of lower latency and smaller energy consumption, new issues related to the cloudlets are rising. For instance, unbalanced task distribution and huge load gaps among heterogeneous mobile cloudlets are becoming more challenging, concerning the network dynamics and distributed task offloading. In this paper, we propose 'FairEdge', a Fairness-oriented computation offloading scheme to enable balanced task distribution for mobile Edge cloudlet networks. By integrating the balls-and-bins theory with fairness index, our solution promotes effective load balancing with limited information at low computation cost. The evaluation results from extensive simulations and experiments with real-world datasets show that, FairEdge outperforms conventional task offloading methods, and it can achieve a network fairness up to 0.85 and reduce the unbalanced task offload by 50%.

INDEX TERMS Mobile cloudlets, load balancing, edge computing, fair task offloading.

I. INTRODUCTION

In recent years, with the rapid development of mobile computing technologies and pervasive proliferation of mobile devices, mobile traffic data has been growing at an unprecedented rate. According to a latest white paper released by Cisco [1], the global mobile traffic data will increase seven-fold between 2017 and 2022, reaching 77 exabytes (1 exabyte = 10^{18} bytes) per month. Notably, of all IP traffic in 2022, over 50% will be Wi-Fi and smartphones will account for nearly 60% traffic offloading. While mobile applications are aggressively demanding in computation resources, mobile devices are still constrained by the limited capacities in the batteries, memory, and processors. As a consequence, the enlarging gap between resource-constrained mobile devices and computing-intensive applications has become a great challenge [2]. Nevertheless, cloud computing is the ultimate solution to deal with this challenge.

Generally, cloud computing allows mobile users to offload computation tasks,[1] *i.e.*, the executable application phases, on to cloud computing infrastructures (*i.e.*, IaaS, PaaS, and SaaS). In the scenario of mobile computing, by migrating computing-intensive tasks to the cloud, mobile devices can benefit from lower energy consumption and enjoy virtually unlimited computing capacity. This is exemplified by a wide range of cloud computing platforms, including Amazon Web Services, Microsoft Azure, and Google Cloud [3]. These cloud computing platforms provide computing services that can be remotely accessed by mobile users. However, existing

The associate editor coordinating the review of this manuscript and approving it for publication was Honghao Gao.

---

[1]In the remaining of this paper, we will use the terms "task offloading" and "computation offloading" interchangeably unless otherwise stated.

studies have shown the limitations of solely relying on offloading tasks to remote clouds. Since mobile users access remote clouds via a wide area network (WAN), they may experience long latencies caused by congested transmission over long distances between end devices and clouds [3].

Subsequently, the concept of mobile edge computing (MEC) has been proposed to provide mobile users with in-proximate computing resources, such as cloudlets [2]. A mobile cloudlet is a trusted, resource-limited cluster of computing servers, which is integrated with wireless local area networks (WLAN). By offloading tasks to a nearby mobile cloudlet, the demands of fast and interactive response can be sufficiently satisfied with the low-latency, one-hop, and high-bandwidth access. In comparison with remote cloud computing resources, the mobile cloudlets at edge networks can improve the task processing time significantly. As a result, mobile users of computation-intensive applications [4] (*e.g.*, virtual reality, image processing, and augmented gaming) can enjoy faster response and better Quality of Services (QoS), and enhanced Quality of Experience (QoE) [5].

However, considering the capacity of each cloudlet is limited, a mobile cloudlet would become overloaded if it travels in an area where too many mobile users offload computation-intensive tasks to it. In that case, above QoS and QoE for mobile users can be seriously impacted, making the communication cost and delay even higher than offloading tasks to a remote cloud. Therefore, it is of great importance to maintain load balancing among all mobile cloudlets at edge networks, so that each cloudlet's computing resource can be fully exploited, and mobile users can also have a quick response on their offloaded tasks.

Unfortunately, most existing solutions to improve the performance of edge networks have overlooked a fundamental issue, *i.e.*, the fairness of task offloading among mobile cloudlets. Indeed, it's difficult to achieve fairness in task offloading among mobile cloudlets, as the mobility of each cloudlet is random, and the network is intermittently connected. Moreover, as computation offloading behaviors of mobile users are uncontrollable, the task load of each cloudlet is highly dynamic, making it costly to probe the overall load information in the cloudlet network for comparison and decision making. Accordingly, two challenges need to be formally addressed.

- First, the load balancing should be achieved under the collaboration among mobile cloudlets. As the mobility-enhanced cloudlets opportunistically encounter each other, it is important for them to collaboratively offload tasks to each other for the benefit of overall load balancing in edge networks.
- Second, it should be low-cost and light-weight to achieve fairness in the mobile cloudlet network. Therefore, a universal fairness metric should be adopted to measure the fairness based on load information of each cloudlet. The fairness metric should be further taken into consideration when mobile cloudlets offload tasks to each other. Moreover, the fairness value of the cloudlet

network should be updated in each time interval, as load information of each cloudlet constantly changes.

In this paper, to deal with the above challenges, we propose FairEdge, a Fairness-oriented task offloading scheme for collaborative mobile cloudlets at Edge networks. FairEdge integrates the balls-and-bins theory [6] with fairness index [7] to achieve effective load balancing in mobile cloudlet networks. Particularly, under the FairEdge scheme, each cloudlet only needs to query load information from two random neighbors in each time interval. By comparing the task load and fairness indexes of these two neighbors, each cloudlet can make a practical decision on task offloading to preserve both load balancing and fairness. Ultimately, the fairness of the mobile cloudlet network will converge, and the fairness-oriented load balancing can be achieved.

The main contributions of this paper are summarized as follows.

- To the best of our knowledge, this work is the first to investigate the fairness issue in mobile cloudlet networks. By jointly considering the balancing property as well as the fairness index, we propose the FairEdge scheme based on *balls-into-bins* theory and Jain's fairness index. The task load information of each cloudlet is collected and compared in a low-cost manner, which resolves the difficulty in information collection from highly dynamic mobile edge networks.
- The Jain's fairness index is integrated as a part of the task offloading algorithm. By leveraging the task load information and fairness index of two targeted neighbors, the proposed FairEdge scheme enables a more reasonable computation offloading decision for each cloudlet. Fairness-oriented task offloading further contributes to the overall load balancing and fairness of the mobile cloudlet network.
- We evaluate the proposed FairEdge with simulations and experiments based on two real-world trace datasets. The evaluation results show that FairEdge can successfully achieve load balancing with guaranteed performance, with a near-optimal fairness index value of 0.85 and an improvement of 50% in balancing tasks among mobile cloudlets.

The rest of the paper is organized as follows. We introduce the related works and preliminaries on investigated issues in Section II and Section III, respectively. Then, we present the system model and task offloading problem in Section IV. We further propose the FairEdge algorithm with detailed descriptions in Section V. In Section VI, we present comprehensive simulation studies with real-world datasets. At last, we discuss the future trend of mobile edge computing in Section VII and conclude the paper in Section VIII.

II. RELATED WORK
The comprehensive reviews on mobile edge computing can be found in [8]. Particularly, Nayyer *et al.* [2] compared the mobile augmentation approaches for resource optimization

from the perspective of cloudlet-based networks. Generally, existing studies for cloudlet load balancing can be categorized into two groups, *i.e.*, optimal cloudlet placement and computation offloading optimization.

For computation offloading optimization, the objects of offloading algorithms include optimizing device energy [9], cloud workload [10], and application latency [11]. For instance, Sun and Ansari [12] proposed a latency-aware workload offloading strategy to balance tasks from mobile users to suitable cloudlets. Huang and Guo [13] and Huang and Song [14] investigated service provisioning problems under the cloudlet-based network and proposed an adaptive update scheme to maximize a weighted profit for network operators. Yang *et al.* [15] jointly considered security and sustainability issues of cloudlet networks and proposed a novel task offloading scheme to avoid DDoS attacks. Similarly, Fan *et al.* [16] proposed CTOM, a collaborative task offloading mechanism for mobile cloudlet networks.

For computation offloading in mobile edge computing scenarios, Du *et al.* [17] investigated the computation offloading problem in a mixed fog/cloud system by jointly optimizing the offloading decisions and allocation of computation resource, transmit power, and radio bandwidth. Zhang *et al.* [7] studied the fair task offloading for fog computing networks, where the task delay and corresponding energy consumption were formulated into the performance index with fairness scheduling metrics. Moreover, Zhu *et al.* [18] formulated the fair resource allocation problem in mobile edge computing as a Nash bargaining resource allocation game. Meskar and Liang [19] designed a multi-resource allocation mechanism by jointly considering dominant resource fairness and external resources fairness. Different from existing works, in this paper, we study the fairness in a mobile edge network, where mobility-enhanced cloudlets collaborate to offload computation tasks to each other. We aim to achieve the fairness and load balancing for the overall edge network of all mobile cloudlets.

III. PRELIMINARY

A. COMPUTATION OFFLOADING IN MOBILE EDGE NETWORKS

With the proliferation of mobile devices and advances in wireless communication technologies, mobile computing has experienced a major shift from centralized cloud computing to mobile edge computing [8]. In a typical mobile edge network, edge cloud servers (*e.g.*, cloudlets) are deployed at fixed locations or enhanced with mobility to provide computing services for nearby mobile users with proximate, high-speed and wireless access. Subsequently, mobile users can offload computation-intensive and latency-sensitive tasks to edge cloud servers for processing, thus saving both energy and computation resources on their own devices [2]. However, computation offloading in mobile edge networks also brings new challenges in how to efficiently utilize edge computing resources and enhance the overall performance

of cloudlets. In this work, we investigate how to improve resource sharing via cooperation and collaboration among mobile cloudlets based on the assumption that computation tasks can be offloaded from one cloudlet to others for more efficient processing. In particular, we adopt balls-into-bins theory for task distribution among mobile cloudlets, to optimize the offloading decisions in a distributed manner with low communication and computation cost.

B. TWO-CHOICE BALLS-INTO-BINS PROCESS FOR FAIRNESS-ORIENTED MOBILE EDGE CLOUDLETS

Balls-into-bins is a classic process to model task distribution among a group of uniform servers [6]. In this study, we adopt the load balancing theory of balls-into-bins process to assist the fairness-oriented task offloading for IoT applications in mobile cloudlet networks. The original goal of balls-into-bins processes is to allocate m balls into n bins, with each ball to be thrown into a uniformly and randomly selected bin at a probability of $1/n$. Based on this allocation process, the key criterion of load balancing in a balls-into-bins process is the maximum load, *i.e.*, the largest number of balls in any bin $\mathbb{M}$. Firstly, when $m = n$ and the task offloading is random, with high probability, the expectation of maximum load $\mathbb{M}$ is [20]:

$$\mathbf{E}(\mathbb{M}) = \Theta(\frac{\log n}{\log \log n}). \tag{1}$$

Meanwhile, if each ball has a chance to query the load information from d random selected bins and then makes allocation decisions based on load comparison of above d bins, the maximum load can be dramatically decreased. By comparison, if each ball is allocated to the least loaded of among d bins, the maximum load can be reduced to [20]:

$$\mathbf{E}(\mathbb{M}) = \frac{\log \log n}{\log d} + \Theta(1). \tag{2}$$

Similarly, a more general case is when $m \gg n$, if the task offloading is random, with high probability, the maximum load is [20]:

$$\mathbf{E}(\mathbb{M}) = \frac{m}{n} + \Theta(\sqrt{\frac{m \log n}{n}}). \tag{3}$$

If the task offloading is based on the load comparison of d random choices, with high probability, the maximum load is reduced to [20]:

$$\mathbf{E}(\mathbb{M}) = \frac{m}{n} + \frac{\log \log n}{\log d} + \Theta(1). \tag{4}$$

In this study, we use the balls-into-bins process to model load balancing with computation tasks (*e.g.*, balls) and mobile cloudlets (*e.g.*, bins), and we explicitly adopt the 'two-choice' paradigm for low-cost communication and computation. Accordingly, suppose that m user tasks are distributed into n mobile cloudlets, each task can be offloaded into the least loaded of $d = 2$ cloudlets independently and uniformly. When $m = n$, the maximum load of any cloudlet is with high probability at [21]:

$$\mathbf{E}(\mathbb{M}) = \Theta(\log \log n). \tag{5}$$

When $m \gg n$, the maximum load of any cloudlet is with high probability at [21]:

$$\mathbf{E}(\mathbb{M}) = \frac{m}{n} + \Theta(\log \log n). \tag{6}$$

Consider a mobile edge network where user tasks follow an arrival rate λ are allocated into n mobile cloudlets, the balls-into-bins process perfectly models the distributed computation offloading in a mobile edge cloudlet network. With the random choice in task offloading, the maximum load under an arbitrary round t is [22]:

$$\mathbf{E}(\mathbb{M}) = O(\frac{1}{1-\lambda} \cdot \log \frac{n}{1-\lambda}), \tag{7}$$

where $\lambda = \lambda(n) < 1$. As we mainly focus on the task redistribution among all mobile cloudlets, we further specify the number of random choices as 2 to enhance the adaptability and scalability of balls-into-bins theory. Accordingly, when selecting a target for computation task offloading, each mobile cloudlet can randomly and independently choose 2 nearby cloudlets within its inter-contact range as candidates. As a result, with a fixed arbitrary round t, the theoretical maximum load of any cloudlet becomes as [22]:

$$\mathbf{E}(\mathbb{M}) = O(\log \frac{n}{1-\lambda}), \tag{8}$$

where $\lambda = \lambda(n) \in [1/4, 1)$.

IV. SYSTEM MODEL AND PROBLEM DEFINITION
In this section, we consider a mobile edge network for cooperative task offloading. First, we model the mobile cloudlet and user task offloading. Then, we formulate the fairness-oriented load balancing problem for a mobile edge network.

A. EDGE CLOUDLET MODEL
In this study, we consider a mobile edge network in an urban area, which consists of: (1) a group of edge mobile cloudlets that are integrated with AP for data transmission and task processing, and (2) a number of mobile users that periodically send computation tasks to nearby cloudlets for task processing. First, we denote K edge mobile cloudlets by $\{1, 2, \ldots, k\}$, the location for each cloudlet as (x_i, y_i), and each mobile cloudlet is enhanced with random mobility to have opportunistic encounters with other cloudlets and mobile users. Moreover, we model each cloudlet i as an $M/M/n$ queue by referencing [23], *i.e.*, each cloudlet i has n_i servers with the service rate μ_i. Specifically, a cloudlet i stores the offloaded tasks as a FIFO queue, with the length of q_i^t at time t. In the edge cloudlet network, computation offloading by mobile users to each cloudlet i is modeled as a Poisson process with task arrival rate λ_i, as the number of tasks would constantly change at each time interval. During time interval t, the response time of a cloudlet i can be calculated as $\left\lceil \frac{q_i^t + \lambda_i}{\mu_i} \right\rceil$. Moreover, each cloudlet i also stores information on the number of tasks offloaded to another cloudlet j as $s_{j,i}$.

B. TASK TRANSMISSION MODEL
In this paper, we assume that each cloudlet can contact with other nearby cloudlets to exchange load information and redirect tasks. As some mobile cloudlets may be overloaded with user tasks, the tasks stored in them could experience long processing delays, which would degrade the service experience for the corresponding mobile users. Therefore, the task transmission model is formulated for mobile cloudlets to perform computation offloading for load balancing collaboratively. From the perspective of cloud service provides, it is also important to enhance the performance of mobile cloudlets to make the edge network more efficient and sustainable. The task transmission model is formulated to address the above issues with the following two considerations. First, only when the distance between two cloudlets is within an inter-contact range R, they can establish an intermittent connection. Second, based on [24], the connecting probability of two cloudlets i and j is computed as:

$$P_{i,j}(t_a, t_b) = e^{-\frac{1}{\alpha_{i,j}} \cdot t}, \quad t \geq 0, \tag{9}$$

where $\alpha_{i,j}$ is the pairwise connection rate of an exponential distribution as $f(t) = \frac{1}{\alpha_{i,j}} \cdot e^{-\frac{1}{\alpha_{i,j}} \cdot t}$. Then, based on the Jain's fairness index [7], we calculate the value of fairness index for each cloudlet i as:

$$f^t(i) = \frac{(\sum_{j=1}^{k} s_{j,i}^t)^2}{K \sum_{j=1}^{k} (s_{j,i}^t)^2}, \tag{10}$$

where K is the total number of edge cloudlets. Likewise, we further calculate the value of fairness index for the mobile edge network as:

$$F = \frac{(\sum_{i=1}^{k} q_i^t)^2}{K \sum_{i=1}^{k} (q_i^t)^2}, \tag{11}$$

where q_i^t is the number of loaded tasks in cloudlet i at time t.

C. PROBLEM DEFINITION
The fairness-oriented load balancing problem in a mobile cloudlet network is defined as follows. Given a set of K mobile edge cloudlet $\{1, 2, \ldots, k\}$, where each cloudlet i with service rate μ_i. Each cloudlet performs a random walk to collect random user tasks with arrival rate λ_i, which follows a Normal distribution. Meanwhile, for each mobile cloudlet i, it has a fairness index value $f^t(i)$ during time interval t. When two cloudlets encounter with each other, they collaboratively share load information and fairness values, then, they would perform fairness-oriented task offloading to enhance the load balancing of the mobile edge network.

1) LOAD BALANCING PROBLEM

The objective of fairness-oriented task offloading is to minimize the differences among task queues of all cloudlets so that the user tasks can be processed with the maximum utilization rate of edge cloudlet computing resources. Here, we formulate the optimization function with each cloudlet's task queue by:

$$\text{minimize } \max_{i=[1,k]} \{\| q_i - \bar{q} \|\}, \qquad (12)$$

where $\bar{q}$ is the averaged value of the task queues of all mobile cloudlets.

2) FAIRNESS OPTIMIZATION PROBLEM

Maximizing the fairness value for each mobile cloudlet can further enhance the efficiency and sustainability of a mobile edge network. Note that the lowest fairness value in a task offloading process among K mobile cloudlets is $\frac{1}{K}$. On the contrary, the highest fairness index value is 1, corresponding to the most balanced task offloading result that all mobile cloudlets hold the same number of user tasks for processing. The fairness optimization problem is as follows:

$$\text{maximize } \min_{i=[1,k]} \{f^t(i), f^t(i) \geq \frac{1}{K}\}, \qquad (13)$$

where $f^t(i)$ is the fairness value of cloudlet i during time interval t, based on Equation 10.

V. ALGORITHM DESIGN
A. OVERVIEW

To tackle the load balancing problem and fairness optimization problem in mobile cloudlet networks, we propose a heuristic algorithm called FairEdge. The major issue of achieving fairness-oriented computation offloading for mobile edge networks is the opportunistic encountering of mobile cloudlets. It would be costly in both computation and communication to control and regulate the task offloading process for all mobile cloudlets in a centralized manner. In contrast, a distributed task offloading scheme is more desirable, since each mobile cloudlet can collaboratively share its load information and fairness value nearby cloudlets. Moreover, with new incoming tasks at each cloudlet, the load information of the edge network constantly changes. To collect the above information and broadcast it to all mobile cloudlets could result in intensive overhead for the edge networks. Last but not least, for each mobile cloudlet, it only needs load information from nearby and contactable targets when making computation offloading decisions. To address above concerns, we are inspired by the 'balls-into-bins' theory and further adopt the 'two-choice' paradigm to design the FairEdge algorithm.

In general, we have three basic assumptions over the mobile edge network. First, each cloudlet i receives user tasks that follow a Poisson process of λ_i, meanwhile, these tasks are executable and offloadable to any other mobile cloudlet for processing. Second, we assume that the mobility trajectory of each cloudlet follows a random walk process within the edge network area. In each time interval, a mobile cloudlet is contactable to any other cloudlet within its communication range. Third, according to [25], the duration of time interval is long enough for each mobile cloudlet to perform a complete computation offloading.

With the above considerations, according to the models in Section IV, we devise an algorithm that enables each mobile cloudlet i to randomly select d target cloudlets within its communication range for computation offloading in each time interval. By probing and comparing load information from d nearby cloudlets, each mobile cloudlet i selects the least loaded one as the target for computation offloading. Then, the fairness index value of the target cloudlet will be computed based on Equation 10 and further compared with the fairness index value of the overall mobile cloudlet network. The computation offloading decision will be made based on the above comparison result. In the following, we formally present FairEdge, the fairness-oriented computation offloading algorithm for mobile cloudlet networks in Algorithm 1.

B. FairEdge ALGORITHM DESIGN

The FairEdge algorithm is proposed to achieve fairness-oriented task offloading in mobile cloudlet networks. To begin with, we define the input and output of FairEdge according to the edge cloudlet model and task transmission model. Next, the algorithm initializes the task queue q_i and record of task offloading S_i for each cloudlet i as well as the time interval t. Starting from the first time interval, FairEdge generates a random location for each cloudlet i and calculate each cloudlet's corresponding task load q_i at current time interval. At this stage, the fairness index value f of the mobile edge network is also calculated using the updated load information of all cloudlets. Next, each cloudlet i will send a probing message and add other cloudlets within communication range into its contact list c_i. To adopt balls-into-bins process for task offloading, FairEdge uses the d-choice policy to select d potential offloading targets in its contact list c_i randomly, and further chooses the least loaded one as the computation offloading target. By comparing the fairness index value f_{choice_i} of the chosen target with f, FairEdge will decide whether to allow cloudlet i to perform task offloading to $choice_i$ or not. If $f_{choice_i} \geq f$, the task offloading will be executed, and the attributes of cloudlet i and the target cloudlet will be updated.

The above process will iterate for each mobile cloudlet i and repeatedly execute for T time intervals. Finally, the FairEdge will output the ultimate task queue q_i and fairness index value f_i for each cloudlet i as well as the ultimate fairness index value of the mobile edge network. Note that the d-choice here is presented for general computation offloading with balls-into-bins theory. In practice, to reduce the communication cost and computation cost in the task offloading process, we apply the '2-choice' paradigm. Thereby, the FairEdge algorithm will only allow

---

Algorithm 1 FairEdge Algorithm

Input:

    Number of cloudlets k, time slots T, random choices d; servers, task arrival/service rates of cloudlet i: n_i, λ_i, μ_i; boundaries: a and b, contact range: r.

Output:

    Each cloudlet's: contact list c_i, task load q_i, fairness index f_i, overall fairness index: f, offload target: $choice_i$, offloading record: $S_i = s_{1,i}, \ldots, s_{k,i}$.

1: Initialize $q_i = 0$, $S_i = \emptyset$, $t = 0$;
2: **while** $t \leq T$ **do**
3:     Generate random location for each cloudlet i as: (x_i, y_i), where $0 < x_i < a$, $0 < y_i < b$;
4:     Calculate task load q_i for i with m, μ_i and λ_i;
5:     Calculate fairness index f with Equation 11;
6:     Select offloading targets for each cloudlet i:
7:     **while** $j \leq K$ **do**
8:         **if** $(x_i - x_j)^2 + (y_i - y_j)^2 < r^2$ **then**
9:             add j into c_i as $c_i^j = 1$;
10:         **end if**
11:     **end while**
12:     **if** $||c_i|| \geq d$ **then**
13:         **do**: randomly choose d cloudlets from c_i;
14:         $choice_i$ is the least load in d chosen cloudlets;
15:     **else if** $0 < ||c_{ij}|| < d$ **then**
16:         $choice_i$ is the least load in $||c_{ij}||$ cloudlets;
17:     **else if** $||c_{ij}|| = 0$ **then**
18:         skip task offloading for cloudlet i in this round;
19:     **end if**
20:     **if** $f_{choice_i} \geq f$ **then**
21:         cloudlet i performs task offloading to $choice_i$;
22:     **end if**
23:     $j = choice_i$;
24:     $s_{j,i} = s_{j,i} + 1$, $q_j = q_j + 1$;
25:     update f_i and f_j;
26:     $t = t + 1$;
27: **end while**
28: **return** task load q_i, offloading record S_i.

---

each cloudlet i randomly choose 2 contactable cloudlets for load comparison in each time interval.

At last, we briefly discuss the theoretical performance of FairEdge. First, as have been discussed in Section III-B, the mobile cloudlet network fits the case where user tasks follow an arrival rate λ into k cloudlets. With the random choice in task offloading, the maximum load under an arbitrary round t would be $\mathbf{E}(\mathbb{M}) = O(\frac{1}{1-\lambda} \cdot \log \frac{k}{1-\lambda})$, where $\lambda = \lambda(n) < 1$. For the 2-choice process, if $\lambda = \lambda(n) \in [1/4, 1)$, the maximum load of any cloudlet becomes as $\mathbf{E}(\mathbb{M}) = O(\log \frac{k}{1-\lambda})$. Second, by leveraging the '2-choice' paradigm for selecting the target for computation offloading, FairEdge only probes load information from two contactable neighboring cloudlets for comparison. According to [26] and [24], such a process would significantly reduce the complexity overhead to $O(1)$ compared with greedy offloading's $O(n)$ complexity.
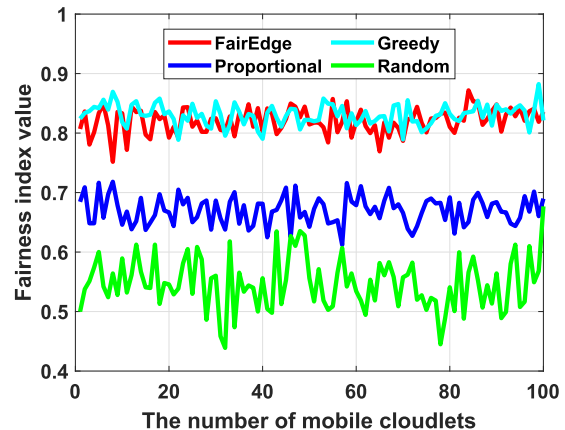

FIGURE 1. The fairness values of all mobile cloudlets in task offloading collaborations.

VI. EXPERIMENTAL STUDIES

In this section, we evaluate the performance of FairEdge with simulations and trace-driven evaluations. We first introduce the basic setups of simulation experiments and then present the evaluation results.

A. SIMULATION STUDY

1) SIMULATION SETUP

According to the mobile edge network model in Section IV and FairEdge Algorithms in Section V, we develop a simulation environment by referencing [16]. The fairness-oriented task offloading scheme is simulated in a 20 km^2 region, and we set the number of mobile cloudlets as 100, the total number of time slots as 600, and the contact range of mobile cloudlets as 20 meter. For each cloudlet i, we set the number of its server n_i by sampling the Poisson distribution with a mean of 2 as well as its service rate μ_i by sampling from the Normal distribution $\mathcal{N}(2, 1) > 0$. Meanwhile, mobile user's task arriving rate at cloudlet i is sampled from the Normal distribution $\mathcal{N}(4, 2) > 0$. We adopt three baseline methods for comparison, including random task offloading, proportional task offloading [26] and greedy task offloading [24]. We run the simulation codes on a Dell laptop with Intel Core i5 CPU, 8GB RAM. Each simulation is executed for 100 times and we report the final average results as follows.

2) EVALUATION OF FAIRNESS INDEX

We first evaluate the fairness of task offloading by calculating the fairness index of individual mobile cloudlet using Equation 10. The fairness index ranges from 0 to 1, with 0 as the most unfair case and 1 as the purely fair case. As shown in Figure 1, under the random offloading scheme and proportional offloading scheme, most of the fairness index values of mobile cloudlets are below 0.6 and 0.7, respectively. In comparison, the proposed FairEdge and greedy algorithm achieve an average value of fairness index over 0.8, showing that the task distribution is well-balanced across all mobile cloudlets. The greedy algorithm applies node traversal on
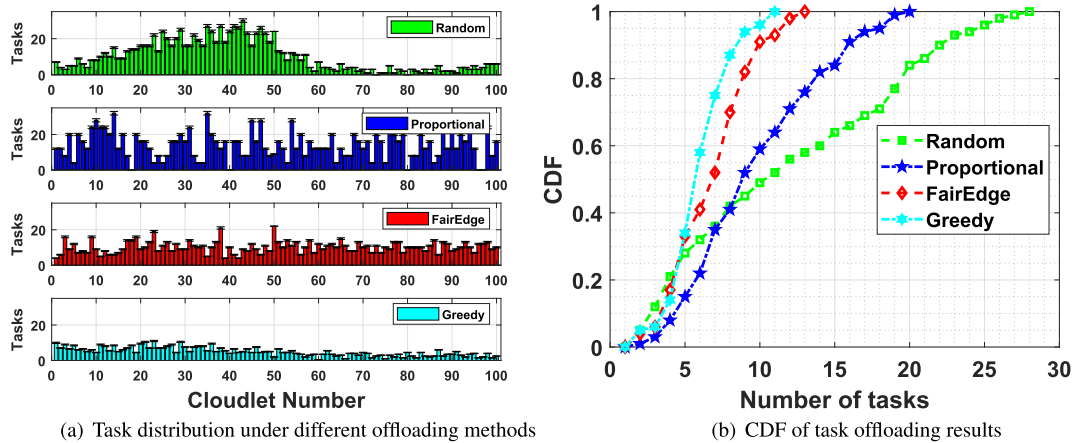
(a) Task distribution under different offloading methods

(b) CDF of task offloading results

FIGURE 2. Comparison of task offloading results in a simulation study.

each cloudlet when finding the least loaded and contactable neighbors for task offloading. Therefore, it ensures both the balance and fairness of task offloading at high communication costs. Meanwhile, the proposed FairEdge adopts balls-into-bins theory with the 'd-choice' scheme in task offloading. It achieves a close-to-greedy performance in fairness values while significantly reduces computation complexity to $o(1)$, as each cloudlet only needs to probe load information from two random neighbors and make a one-time comparison. Moreover, the fairness index values of FairEdge at some cloudlets are higher than those greedy offloading. In summary, the FairEdge can achieve network fairness up to 0.85 and reduce the unbalanced task offload by up to 50% in comparison with other baseline methods.

3) EVALUATION OF TASK DISTRIBUTION

Second, we evaluate the task distribution results under different task offloading algorithms. The bar plot of Figure 2 shows the final task distribution after all offloading time intervals. Obviously, under both random task offloading and proportional offloading schemes, there are huge gaps (up to 30) among different mobile cloudlets. In random task offloading, a group of mobile cloudlets (number 15 to 50) are processing much more tasks than other cloudlets (*e.g.*, number 60 to 90). Meanwhile, under proportional offloading, the overloaded mobile cloudlets are distributed more dispersedly. The above unbalance in task distribution would not only harm the fairness of mobile cloudlet network but also degrade the user experience, as cloudlets need more time to process all tasks. In comparison, the proposed FairEdge successfully enhances the balance in task offloading, as most cloudlets have nearly 10 tasks to process. The greedy method achieves the best performance in balancing task distribution at the cost of high communication and computation overheads, where most cloudlets are offloaded with less than 10 tasks and no cloudlet is idle.

To make a further comparison, we present the empirical cumulative distribution results of task offloading

in Figure 2(b). Here, the performance of FairEdge is very close to that of greedy offloading, where over 90% of mobile cloudlets are offloaded with less than 10 tasks. In contrast, the task offloading result by the proportional method shows that almost 20% of cloudlets are offloaded with more than 15 tasks. Besides, over 20% of mobile cloudlets have more than 20 tasks to process under random offloading. The above evaluation results validate the effectiveness of FairEdge, as it manages to balance the task distribution by using the fairness index and '2-choice' paradigm. In the following, we further evaluate the FairEdge in real-world scenarios by using mobility trace datasets for simulation.

B. EVALUATION ON REAL-WORLD TRACE DATASETS

To explore the feasibility of FairEdge in real-world scenarios, we conduct trace-driven studies of mobile computation offloading with two real-world mobility trace datasets. In brief, the two trace datasets contain Bluetooth encounter records of mobile nodes that can be used to emulate the communications among mobile cloudlets at edge networks. The reasons for using two different trace datasets for evaluations are: 1) to test the performance of FairEdge in different network scenarios, where cloudlets have different patterns of mobility; 2) to examine the scalability of FairEdge with mobile cloudlet networks in different scales. We present the details of each dataset and corresponding evaluation results of mobile computation offloading in the following.

1) MOBICLIQUE DATASET
a: BASIC SETUPS
We adopt a real-world mobility dataset called 'Mobi-Clique' [27] to emulate the random mobility of mobile cloudlets for task sharing and computation offloading. This dataset contains encountering records collected by a mobile network software called MobiClique. MobiClique leverages opportunistic contacts (*e.g.*, Bluetooth encounters) between smartphones to form a decentralized ad-hoc network for information sharing. The trace data of MobiClique
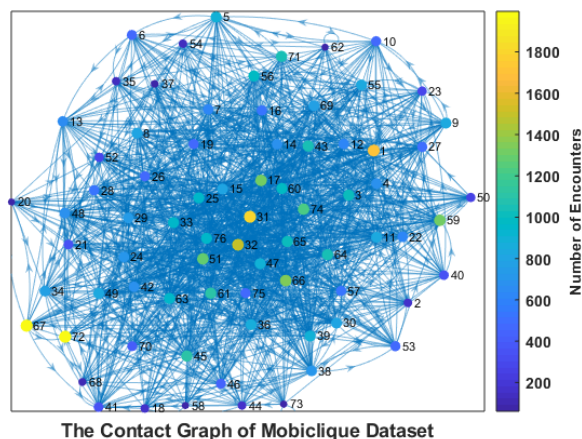
The Contact Graph of Mobiclique Dataset

FIGURE 3. Nodes and encountering records in the MobiClique [27] dataset.

was collected with 76 participants during the SIGCOMM 2009 conference in Barcelona, Spain. We process the Mobi-Clique dataset as an edge mobile cloudlet network and visualize it in Figure 3. Here, each vertex represents a mobile cloudlet, an edge between two vertices represents a contact, and the color of a vertex shows its active level in the network. In total, there are 76 mobile cloudlets and 69,186 contacts in the MobiClique dataset. Moreover, the timestamp of the first contact is 30 seconds, and the timestamp of the last contact is 320,684 seconds. Based on the above, we set the length of a time slot for computation offloading as 200 seconds, so that there are 1,604 time slots in total. For each cloudlet i, we set its number of servers, service rate, and task arriving rate the same as the previous simulation setup.

b: TASK OFFLOADING RESULTS
We conduct the mobile cloudlet task offloading with Mobi-Clique dataset for 100 times and take the average values of task distributions and standard deviations as the final task offloading results. The baseline methods include random task offloading, proportional task offloading, and greedy task offloading. As shown in Figure 4, under random offloading and proportional offloading, some particularly active mobile cloudlets are extremely overloaded (*e.g.*, offloaded with over 150 and even 200 tasks). While proportional offloading partially reduces the number of overloaded cloudlets, the overall task distribution is still highly imbalanced. In contrast, the proposed FairEdge scheme and greedy offloading scheme show remarkable performance in balancing the task distributions over the entire network, where the task load of each cloudlet is under 50. Also, by combining the '2-choice' paradigm from balls-into-bins theory with Jain's Fairness index, FairEdge further achieves a slightly lower task load on each cloudlet throughout the task offloading process.

The empirical cumulative distribution of task offloading results with the MobiClique dataset is presented in Figure 4(b). More than 80% of mobile cloudlets in FairEdge and greedy schemes are offloaded with less than 25 tasks. Meanwhile, more than 20% of mobile cloudlets in

random and proportional schemes have more than 30 tasks. The above evaluation results show the effectiveness of fairness-oriented task offloading scheme in a real-world scenario. FairEdge can effectively achieve balanced task offloading on a real-world mobility trace dataset, where different cloudlets have great disparities in active levels of mobility.

2) HAGGLE DATASET
a: BASIC SETUPS
We further evaluate the performance of FairEdge in a larger mobility trace dataset, *i.e.*, Haggle dataset [28]. The Haggle dataset is under the project of Koblenz Network Collection (KONECT) [29] for systematic study on diverse networks. In short, the Haggle dataset contains mobility and connectivity traces that were generated from iMote devices. The iMote devices are small portable devices to capture Bluetooth sightings (encounters) of their carriers. We process and visualize the contact graph of the Haggle dataset in Figure 5, where all 274 vertices are with 28,244 edges. Similar to the Mobi-Clique dataset, each vertex in the contact graph represents a mobile cloudlet, an edge between two vertices represents a contact and the color of each vertex shows its active level in the network. Different from the MobiClique dataset, the vertices in the Haggle contact graph are more distributed, where a small number of vertices form a 'contact center' (1-77) that links the rest edge vertices with sparse contacts. In the following evaluations, all basic setups are the same as the evaluations with the MobiClique dataset, except for the time interval. In the Haggle dataset, the beginning and ending timestamps are 20,733 seconds and 364,094 seconds, respectively. As the overall duration in the Haggle dataset is much longer than that of MobiClique, we set the length of a time slot as 3,600 seconds (*i.e.*, 1 hour) for computation offloading simulation.

b: TASK OFFLOADING RESULTS
In evaluations with the Haggle dataset, we also conduct task offloading with mobile cloudlets for 100 times. We take the average values of task distributions and standard deviations as the final results. As shown in Figure 6, a group of cloudlets with high contact level take the majority of tasks in the Haggle network. This is due to that the rest of each mobile cloudlet only has several contact opportunities for task offloading, and target cloudlets in these contacts are all in the group of the 'contact center' (cloudlets 1-77). As the cloudlets of number 80-274 have very few encounters with others, most of them are offloaded only a few tasks or even none. To make it clearer for performance comparison, we only provide the offloading results for cloudlets 1-77. The random task offloading shows the worst performance in balancing the task load in the contact center, as some cloudlets are overloaded with nearly 100 tasks. While proportional task offloading slightly improves the task distribution result, there still exist huge gaps (over 90 tasks) among cloudlets in the contact center. In contrast, FairEdge and greedy task offloading schemes
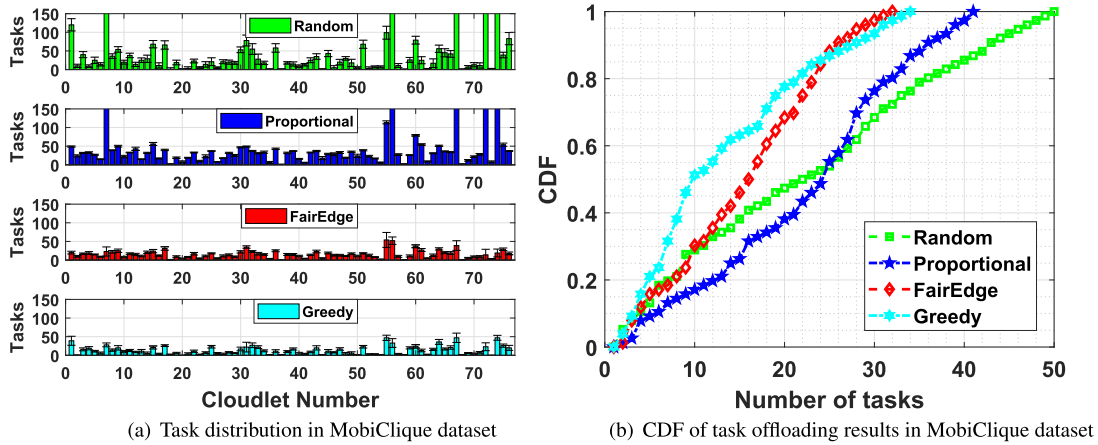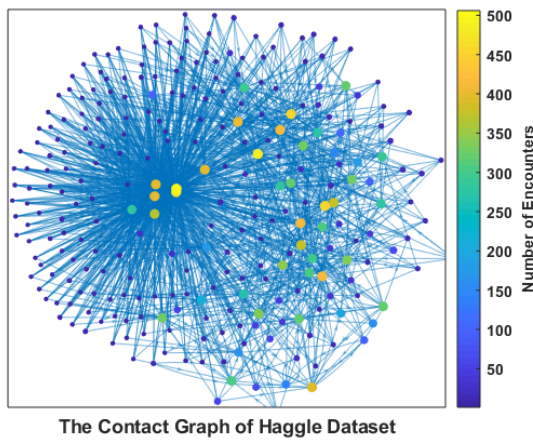
(a) Task distribution in MobiClique dataset

(b) CDF of task offloading results in MobiClique dataset

FIGURE 4. Comparison of task offloading results in trace-driven evaluations with the **MobiClique** dataset.

The Contact Graph of Haggle Dataset

FIGURE 5. Nodes and encountering records in the Haggle [28] dataset.

significantly enhance the balance in task distribution over all cloudlets in the contact center, where most of the cloudlets are offloaded with less than 50 tasks. Besides, for the cloudlets

with low contact level, FairEdge still preserves their fairness by re-balancing tasks from overloaded cloudlets to others. The empirical cumulative distribution of task offloading results with the Haggle dataset is presented in Figure 6(b). This CDF figure reveals that even there are huge gaps in contact level among different mobile cloudlets, FairEdge can still achieve balanced task offloading with the close performance to the greedy algorithm, showing the effectiveness of preserving fairness in mobile computation offloading.

VII. THE FUTURE DIRECTION

With the arrival of 5G, Mobile Edge Computing (MEC) is regarded as a promising technology for future communication systems, which brings the computing and storage resources to the proximity of mobile users [30]. While mobile users can save energy on their devices and reduce latency by offloading computation-intensive tasks to nearby edge computing servers, the task offloading problems in MEC remain as NP-hard [31]. Such problems are generally formulated with typical network settings and further solved with
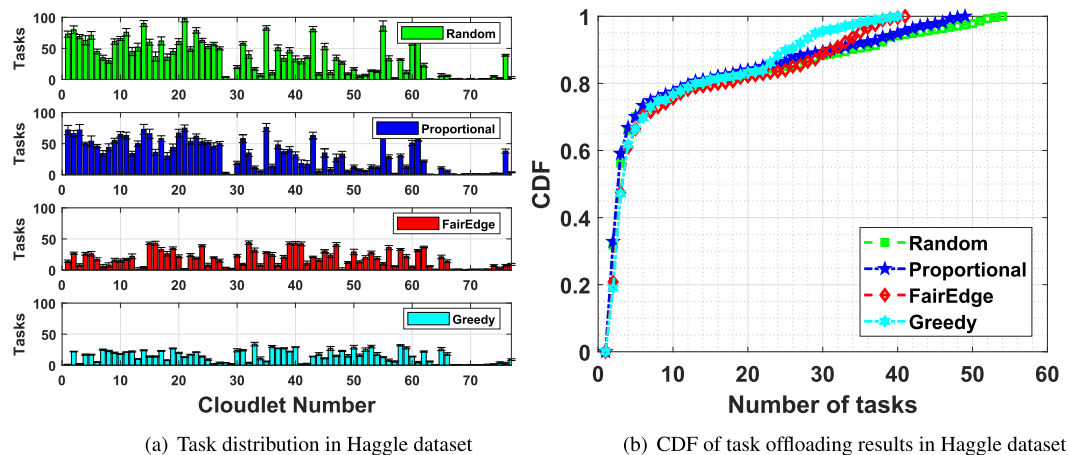

(a) Task distribution in Haggle dataset

(b) CDF of task offloading results in Haggle dataset

FIGURE 6. Comparison of task offloading results in trace-driven evaluations with the **Haggle** dataset.

task offloading strategies based on heuristic algorithms [32]. However, for the coming 5G networks, a typical mobile edge node is expected to have more than 2000 configurable parameters [33]. In that sense, with the explosive growth of mobile data, existing heuristic algorithms are not capable and scalable to tame the complexity in the highly dynamic computation offloading environment.

To address the above challenges, the future direction to optimize computation offloading in mobile edge computing networks is to apply deep learning. The Deep Reinforcement Learning (DRL) approaches can essentially eliminate the need for solving combinatorial optimization problems in MEC task offloading, thereby significantly reducing the computational complexity. For example, Huang *et al.* [34] proposed DROO, a Deep Reinforcement learning-based Online Offloading (DROO) framework that can learn binary offloading decisions from the past experiences and update offloading policy. Similarly, Li *et al.* [35] designed a DRL-based optimization framework with Q-learning for multi-use computation offloading and resource allocation in MEC. Moreover, Ning *et al.* [36] proposed a distributed DRL-based solution to minimize the offloading cost while satisfying the latency constraints of users in 5G-enabled vehicular networks. To this end, the above recent works have shown a significant new future direction of combining MEC with deep learning for emerging 5G networks.

VIII. CONCLUSION

In this paper, we have proposed FairEdge, a Fairness-oriented task offloading scheme to enable balanced task sharing and computation offloading for mobile Edge cloudlet networks. The FairEdge integrates balls-into-bins theory and Jain's fairness index for distributed task offloading among mobile cloudlets. We have developed the system model of computation offloading in edge mobile cloudlet networks, and formulated the load balancing problem together with fairness optimization problem. By adopting the 'two-choice' paradigm and using calculated fairness index values for cloudlets and the network, we have further proposed algorithm design of *FairEdge* and conducted extensive evaluation studies with simulations and experiments on real-world trace datasets. The experimental results have shown that FairEdge successfully achieved load balancing with guaranteed performance, with a near-optimal fairness index of up to 0.85 and an improvement of 50% over conventional baseline methods.

REFERENCES

[1] Cw Paper. (2019). *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022*. Accessed: Jan. 5, 2019. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.pdf

[2] M. Z. Nayyer, I. Raza, and S. A. Hussain, "A survey of cloudlet-based mobile augmentation approaches for resource optimization," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–28, Nov. 2018.

[3] A. J. Ferrer, J. M. Marquès, and J. Jorba, "Towards the decentralised cloud: Survey on approaches and challenges for mobile, ad hoc, and edge computing," *ACM Comput. Surv.*, vol. 51, no. 6, pp. 1–36, Jan. 2019.

[4] H. Gao, Y. Duan, L. Shao, and X. Sun, "Transformation-based processing of typed resources for multimedia sources in the IoT environment," *Wireless Netw.*, vol. 25, no. 1, pp. 1–17, 2019.

[5] H. Gao, Y. Xu, Y. Yin, W. Zhang, R. Li, and X. Wang, "Context-aware QoS prediction with neural collaborative filtering for Internet-of-Things services," *IEEE Internet Things J.*, early access, doi: 10.1109/JIOT.2019.2956827.

[6] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[7] G. Zhang, F. Shen, Y. Yang, H. Qian, and W. Yao, "Fair task offloading among fog nodes in fog computing networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[8] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.

[9] L. Gong, Y. Zhao, C. Xiang, Z. Li, C. Qian, and P. Yang, "Robust lightweight magnetic-based door event detection with smartphones," *IEEE Trans. Mobile Comput.*, vol. 18, no. 11, pp. 2631–2646, Nov. 2019.

[10] Y. Qu, S. Tang, C. Dong, P. Li, S. Guo, H. Dai, and F. Wu, "Posted pricing for chance constrained robust crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 19, no. 1, pp. 188–199, Jan. 2020.

[11] X. Fan, X. He, C. Xiang, D. Puthal, L. Gong, P. Nanda, and G. Fang, "Towards system implementation and data analysis for crowdsensing based outdoor RSS maps," *IEEE Access*, vol. 6, pp. 47535–47545, 2018.

[12] X. Sun and N. Ansari, "Latency aware workload offloading in the cloudlet network," *IEEE Commun. Lett.*, vol. 21, no. 7, pp. 1481–1484, Jul. 2017.

[13] H. Huang and S. Guo, "Service provisioning update scheme for mobile application users in a cloudlet network," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[14] H. Huang and G. Song, "Adaptive service provisioning for mobile edge cloud," *ZTE Commun.*, vol. 15, no. 2, pp. 1–9, 2017.

[15] N. Yang, X. Fan, D. Puthal, X. He, P. Nanda, and S. Guo, "A novel collaborative task offloading scheme for secure and sustainable mobile cloudlet networks," *IEEE Access*, vol. 6, pp. 44175–44189, 2018.

[16] X. Fan, X. He, D. Puthal, S. Chen, C. Xiang, P. Nanda, and X. Rao, "CTOM: Collaborative task offloading mechanism for mobile cloudlet networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[17] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee," *IEEE Trans. Commun.*, vol. 66, no. 4, pp. 1594–1608, Apr. 2018.

[18] Z. Zhu, J. Peng, X. Gu, H. Li, K. Liu, Z. Zhou, and W. Liu, "Fair resource allocation for system throughput maximization in mobile edge computing," *IEEE Access*, vol. 6, pp. 5332–5340, 2018.

[19] E. Meskar and B. Liang, "Fair multi-resource allocation with external resource for mobile edge computing," in *Proc. IEEE INFOCOM IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2018, pp. 184–189.

[20] P. Berenbrink, K. Khodamoradi, T. Sauerwald, and A. Stauffer, "Balls-into-bins with nearly optimal load distribution," in *Proc. 25th ACM Symp. Parallelism Algorithms Archit. (SPAA)*, 2013, pp. 326–335.

[21] M. Mitzenmacher, "The power of two choices in randomized load balancing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 12, no. 10, pp. 1094–1104, Oct. 2001.

[22] P. Berenbrink, T. Friedetzky, P. Kling, F. Mallmann-Trenn, L. Nagel, and C. Wastell, "Self-stabilizing balls & bins in batches: The power of leaky bins," in *Proc. ACM Symp. Princ. Distrib. Comput.* New York, NY, USA: ACM, 2016, pp. 83–92.

[23] M. Jia, W. Liang, Z. Xu, and M. Huang, "Cloudlet load balancing in wireless metropolitan area networks," in *Proc. IEEE INFOCOM 35th Annu. IEEE Int. Conf. Comput. Commun.*, Apr. 2016, pp. 1–9.

[24] Q. Li, P. Yang, X. Fan, S. Tang, C. Xiang, D. Guo, and F. Li, "Taming the big to small: Efficient selfish task allocation in mobile crowdsourcing systems," *Concurrency Computat., Pract. Exper.*, vol. 29, no. 14, Jul. 2017, Art. no. e4121.

[25] M. Jia, W. Liang, Z. Xu, M. Huang, and Y. Ma, "QoS-aware cloudlet load balancing in wireless metropolitan area networks," *IEEE Trans. Cloud Comput.*, early access, doi: 10.1109/TCC.2017.2786738.

[26] Q. Li, P. Yang, S. Tang, M. Zhang, and X. Fan, "Equilibrium is priceless: Selfish task allocation for mobile crowdsourcing network," *EURASIP J. Wireless Commun. Netw.*, vol. 2016, no. 1, p. 166, 2016.

[27] A.-K. Pietilainen and C. Diot, "Social pocket switched networks," in *Proc. IEEE INFOCOM Workshops*, Apr. 2009, pp. 1–2.

[28] J. Kunegis. (2017). *Haggle Network Dataset Konect*. Accessed: Dec. 1, 2019. [Online]. Available: http://konect.uni-koblenz.de/networks/contact

[29] K. Jerome, "Konect: The Koblenz network collection," in *Proc. 22nd Int. Conf. World Wide Web*. New York, NY, USA: ACM, 2013, pp. 1343–1350.

[30] L. Yang, H. Zhang, M. Li, J. Guo, and H. Ji, "Mobile edge computing empowered energy efficient task offloading in 5G," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6398–6409, Jul. 2018.

[31] J. Wang, J. Hu, G. Min, W. Zhan, Q. Ni, and N. Georgalas, "Computation offloading in multi-access edge computing using a deep sequential model based on reinforcement learning," *IEEE Commun. Mag.*, vol. 57, no. 5, pp. 64–69, May 2019.

[32] C. Xiang, P. Yang, C. Tian, L. Zhang, H. Lin, F. Xiao, M. Zhang, and Y. Liu, "CARM: Crowd-sensing accurate outdoor RSS maps with error-prone smartphone measurements," *IEEE Trans. Mobile Comput.*, vol. 15, no. 11, pp. 2669–2681, Nov. 2016.

[33] R. Li, Z. Zhao, X. Zhou, G. Ding, Y. Chen, Z. Wang, and H. Zhang, "Intelligent 5G: When cellular networks meet artificial intelligence," *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 175–183, Oct. 2017.

[34] L. Huang, S. Bi, and Y. J. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Trans. Mobile Comput.*, early access, doi: 10.1109/TMC.2019.2928811.

[35] J. Li, H. Gao, T. Lv, and Y. Lu, "Deep reinforcement learning based computation offloading and resource allocation for MEC," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2018, pp. 1–6.

[36] Z. Ning, Y. Li, P. Dong, X. Wang, M. S. Obaidat, X. Hu, L. Guo, Y. Guo, J. Huang, and B. Hu, "When deep reinforcement learning meets 5G vehicular networks: A distributed offloading framework for traffic big data," *IEEE Trans. Ind. Inf.*, early access, doi: 10.1109/TII.2019.2937079.

ZHIYUAN TAN received the Ph.D. degree in computer systems from the University of Technology Sydney (UTS), Australia, in 2014. He is currently a Lecturer and a Ph.D. supervisor with the School of Computing, Edinburgh Napier University (ENU), U. K. Prior to joining ENU, in 2016, he held different research positions with the University of Twente (Netherlands), UTS (Australia) and La Trobe University (Australia), respectively. His current research interests include cybersecurity, machine learning, pattern recognition, data analytics, virtualisation and cyber-physical system. He has received funding from the Commonwealth Scientific and Industrial Research Organisation (CSIRO), UTS and ENU Development Trust, and the Scottish Informatics and Computer Science Alliance (SICSA) for his research. He also has been a co-investigator of other network security research projects funded by CSIRO, Minster of Education (Oman), and ITEA2-/CATRENE. He has published over 50 quality scholarly articles. His most recent research achievements have been published in highly-cited the IEEE Transactions, Elsevier and Springer journals as well as premier international conferences. He has an H-Index of 14 and his research articles have received over 1000 citations, since 2014. Moreover, over the past years he has received several international research awards, including two best paper awards and the National Research Award, in 2017, from the Research Council of the Sultanate of Oman. Besides, he has chaired various major international workshops and conferences, as well as Guest-Edited Special Issues for international journals including Concurrency and Computation: Practice and Experience (Wiley), Computers and Electrical Engineering (Elsevier), IEEE Access, *ACM Transactions on Multimedia Computing, Communications, and Applications*. He is also a Section Editor of *Ad Hoc & Sensor Wireless Networks Journal*.

SHUANG LAI is currently an Associate Professor with the School of Humanities, Economics and Law, Northwestern Polytechnical University, Xi'an, China. His major research interests include network and information security, electronic government, electronic commerce, and emergency management.

YUANFANG ZHANG is currently pursuing the Ph.D. degree with Northwestern Polytechnical University, China, and the dual Ph.D. degrees with the School of Electrical and Data Engineering, University of Technology Sydney (UTS), Australia. His research interests include low-light scenario enhancement, saliency detection, object detection/recognition, and reinforcement learning.

XIAOCHEN FAN (Student Member, IEEE) received the B.S. degree in computer science from the Beijing Institute of Technology, China, in 2013. He is currently pursuing the Ph.D. degree with the School of Electrical and Data Engineering, University of Technology Sydney (UTS), Australia. His research interests include mobile/pervasive computing, deep learning, and the IoT.

XIANGJIAN HE (Senior Member, IEEE) received the Ph.D. degree in computing sciences from the University of Technology Sydney, Australia, in 1999. Since 1999, he has been with the University of Technology Sydney. He is currently a Full Professor and the Director of the Computer Vision and Pattern Recognition Laboratory.

QIANWEN YE received the B.S. degree in electronic information engineering from the Hubei University of Technology, Wuhan, China, in 2013. She is currently pursuing the dual Ph.D. degrees with the Faculty of Engineering and Information Technology, University of Technology Sydney, and the School of Information and Communication engineering, Beijing University of Posts and Communications. Her current research interests include the IoT, cooperation location, IFI, and channel models.

PRIYADARSI NANDA received the Ph.D. degree from the University of Technology Sydney, Australia. He is currently a Senior Lecturer with the School of Computing and Communications, University of Technology Sydney. He has more than 26 years of experience in cybersecurity, the Internet of Things security, networks quality of service, assisted health care using sensor networks, and wireless sensor networks. He has published more than 70 refereed research articles.