IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2019.DOI

FairEdge: A Fairness-oriented Task Offloading Scheme for IoT Applications in Mobile Cloudlet Networks

SHUANG LAI[1], XIAOCHEN FAN[2], YUANFANG ZHANG[2,3], ZHIYUAN TAN[4], XIANGJIAN HE[2], 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, Ultimo, NSW 2007, Australia
[3]School of Computer Science, Northwestern Polytechnical University, Xi'an, 710072, China
[4]School of Computing, Edinburgh Napier University, Edinburgh, EH11 4DY, U.K.

Corresponding authors: Xiaochen Fan (email: 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 challenging with respect to 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, 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 year, with the rapid development of mobile computing technologies and pervasive proliferation of mobile devices, mobile traffic data has been growing at an unprecedent rate. According to a latest white paper released by Cisco [1], the global mobile traffic data will increase sevenfold 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 [2], [3], mobile devices are still constrained by the limited capacities in the batteries, memory and processers. As a consequence, the enlarging gap between resource-constrained mobile devices and computing-intensive applications has become a great challenge.

It is believed that 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 the virtually unlimited computing capacity. This is exemplified by a wide range of cloud computing platforms, including Amazon Web Services, Microsoft Azure and Google Cloud [4]. These cloud computing platforms provide computing services that can be remotely accessed by mobile users. However, existing studies have shown the limitations of solely relying on offloading tasks to remote clouds. Since mobile users access remote clouds via wide area network (WAN), they may experience long latencies caused by congested transmission over long distance between end devices and clouds [4], [5].

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

Subsequently, the concept of mobile edge computing (MEC) has been proposed to provide mobile users with in-proximate computing resources, such as cloudlets [5], [6]. A mobile cloudlet is a trusted, resource limited cluster of computing servers that are integrated with local are wireless networks. By offloading tasks to a nearby mobile cloudlet, the demands of fast and interactive response by mobile users can be sufficiently satisfied with 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, for mobile users of computation-intensive applications [7], [8] (*e.g.*, virtual reality, image processing and augmented gaming), they can enjoy faster response and better Quality of Services [9], [10] (QoS) as well as Quality of Experience [11] (QoE).

However, considering the capacity of each cloudlet is limited, a mobile cloudlet would become overloaded if 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 quick response on their offloaded tasks.

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

- First, 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.
- Second, the fairness of the mobile cloudlet network should be low-cost and light-weight to achieve. Therefore, an universal metric should be adopted to measure the fairness based on load information of each cloudlet. The fairness values should be further taken into consideration when mobile cloudlets offload tasks to each other. Moreover, the fairness metric of the cloudlet network should be updated in each time interval, as load information of all cloudlets are constantly changing.

In this paper, to deal with the aforementioned challenges, we propose FairEdge, a fairness-oriented task offloading scheme for collaborative mobile cloudlets at edge networks. FairEdge integrates the balls-and-bins theory [12] with fairness index [13] to achieve effective load balancing in mobile cloudlet networks. Particularly, under FairEdge scheme, each cloudlet only needs to query load information from two random neighboring peers 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 fairness issue in mobile cloudlet networks. By fully considering the balancing property as well as the fairness index, we propose FairEdge scheme based on balls-into-bins theory and Jain's fairness index. The task load information of 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 part of the task offloading algorithm. By jointly considering the task load information and fairness index of two targeted neighbors, the proposed FairEdge scheme enables a more reasonable offloading decision for each cloudlet, and this further contribute to the overall load balancing and fairness of the mobile cloudlet networks.
- We evaluate the proposed FairEdge with simulations based on real-world datasets. The evaluation results show that FairEdge can successful achieve load balancing with guaranteed performance, with a near-optimal fairness index of 0.85 and an improvement of 50% in balancing tasks among mobile cloudlets.

The rest of paper is organized as follows. We introduce the related works and preliminaries on investigated issues in Section II and Section III, respectively. Then, we introduce system model and the problem in Section IV. We further present our 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 [14], [15]. Particularly, Nayyer *et al.* [6] compared the mobile augmentation approaches [16], [17] 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 common objects of offloading algorithms include optimizing device energy [18], [19], bandwidth utilization [20], [21], network connectivity [22], cloud workload [23] and application latency [24]. For instance, Sun *et al.* [25] proposed a latency-aware workload offloading strategy to balance tasks from mobile users to suitable cloudlets. Huang *et al.* [26], [27] investigated the service provisioning problem under the cloudlet-based network and proposed an adaptive update scheme with the objective to maximize a weighted profit for network operators. Yang *et al.* [28] jointly considered security and sustainability issues of cloudlet networks and proposed a novel task offloading scheme to avoid DDoS attacks. Similarly, Fan *et al.* [29] proposed CTOM, a collaborative task offloading mechanism for mobile cloudlet networks. Chen *et al.* [30] further investigated the task offloading problem in ultra-dense network and formulated the offloading problem as a mixed integer non-linear problem.

For computation offloading in mobile edge computing scenarios, Du *et al.* [31] 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, while guaranteed fairness for each individual mobile user. Zhang *et al.* [13] 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.* [32] formulated the fair resource allocation problem in mobile edge computing as a Nash bargaining resource allocation game and further introduced the time-sharing variable to solve the problem. Meskar *et al.* [33] 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, with the aim to achieve the fairness and load balancing for the overall edge network.

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 [14]. In a typical mobile edge network, edge cloud servers (*e.g.*, cloudlets) are deployed at fixed location or enhanced with mobility to be accessed by 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 [15]. However, computation offloading in mobile edge networks also brings new challenges in how to efficiently utilize edge computing resource to 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 [12]. 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 $\mathbb{M}$. First, when $m = n$ and the task offloading is random, with high probability the expectation of maximum load $\mathbb{M}$ is [34]:

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

Meanwhile, if each ball has a chance to query the load information from d random selected bins and then makes allocation decision based on load comparison of above d bins, the maximum load can be dramatically deceased. By comparison, if each ball is allocated to the least loaded of among d bins, the maximum load is with high probability as [34]:

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

Similarly, a more general case is when $m \gg n$, if the task offloading is random, with high probability the maximum load is [34]:

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

If the task offloading is based on the load comparison of d random choices, with high probability the maximum load is reduced to [34]:

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

In this study, we use 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, as 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 [35]:

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

When $m \gg n$, the maximum load of any cloudlet is with high probability at [35]:

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

Consider a mobile edge network where user tasks follow an arrival rate λ are allocated into n mobile cloudlets, a 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 [36]:

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

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 cloudlets becomes as [36]:

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

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 $\{1, 2, ..., k\}$, the location for each cloudlet as (x_i, y_i), and each mobile cloudlet is enhanced with random mobility to have opportunistic encounter with other cloudlets and mobile users. Moreover, we model each cloudlet i as an $M/M/n$ queue by referencing [37], *i.e.*, each cloudlet i has n_i servers with the service rate μ_i. Specifically, a cloudlet i stores the offloaded tasks as an 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. At 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 of 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 connect 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 delay, which could degrade the service experience for the corresponding mobile users. Therefore, the task transmission model is formulated for mobile cloudlets to collaboratively perform computation offloading for load balancing. From the perspective of cloud service provides, it is also important to enhance performance of mobile cloudlets to make the edge network more efficient and sustainable. The task transmission model is formulated for to address above issues with following two considerations. First, only when the distance between two cloudlets are within an inter-contact range R, they can establish a intermittent connection. Second, according to [38], the connecting probability of two cloudlets i and j is computed as:

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

where $\alpha_{i,j}$ is the pairwise connection rate of an exponential distribution as $f(t) = \frac{1}{\alpha_{i,j}} \cdot e^{-\frac{1}{\alpha_{i,j}} \cdot t}$. Second, based on the Jain's fairness index [13], 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 task 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 $\{1, 2, ..., k\}$, 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 $f_t(i)$ at 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.

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:

$$\text{minimize} \max_{i=[1,k]} \{\parallel q_i - \bar{q} \parallel\}, \tag{12}$$

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

Fairness Optimization Problem: Maximizing the value for each mobile cloudlet can further enhance the efficiency and sustainability of a mobile edge network. Note that the maximum 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 cloudlet holds the same number of user tasks for processing. The fairness optimization problem is as follows:

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

where $f(i)$ is the fairness value of cloudlet i at 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 collaborative mobile cloudlets is determined by the opportunistic encounters of mobile cloudlets. Apparently, it would be costly in both computation and communication to control and regulate 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 task offloading in each time interval, the load information of the whole network constantly change. To collect above information and broadcast it to all mobile cloudlet 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 by/to any other mobile cloudlet. Second, we assume that the mobility trajectory of each cloudlet follows a random walk process within the edge network area. At each time interval, a mobile cloudlet is contactable to any other cloudlet within its communication range. Third, according to [39], the duration of time interval is long enough for each mobile cloudlet to perform a complete computation offloading.

With above considerations in mind, 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 over all mobile cloudlet network. The computation offloading decision will be made with the above comparison result. Based on the above explanation, we formally present FairEdge, the fairness-oriented computation offloading algorithm for mobile cloudlet networks as in Algorithm 1.

B. FAIREDGE ALGORITHM DESIGN

The FairEdge algorithms 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 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 to that performs random walk and calculate its 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 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 d-choice policy to randomly select d potential targets in its contact list c_i 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 successful and attributes related to 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 mobile edge networks. 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 task offloading process, we apply the '2-choice' paradigm. Thereby, the FairEdge algorithm will only allow each cloudlet i randomly choose 2 contactable cloudlets for load comparison in each time interval.

At last, we brief 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 cloudlets becomes as $\mathbf{E}(\mathbb{M}) = O(\log \frac{k}{1-\lambda})$. Second, by leveraging '2-choice' paradigm for selecting computation offloading target, FairEdge only probes load information from two contactable neighboring cloudlets for comparison. According to [38] and [40], such process would significantly reduce the complexity overhead to $O(1)$ compared with greedy offloading's $O(n)$ complexity.

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: $S_i = s_{1,i}, ..., s_{k,i}$.

```
Initialize q_i=0, S_i=∅, t=0;
while t ≤ T do
    Generate random location for each cloudlet i as: (x_i, y_i), where 0 < x_i < a, 0 < y < b;
    Calculate task load q_i for i with m, μ_i and λ_i;
    Calculate fairness index f with Equation 11;
    Select offloading target for each cloudlet i:
    while j ≤ K do
        if (x_i − x_j)^2 + (y_i − y_j)^2 < r^2 then
            add j into c_i as c_i^j = 1;
        end if
        if ||c_i|| ≥ d then
            do: randomly choose d cloudlets from c_i;
            choice_i is the least load in d chosen cloudlets;
        else if 0 < ||c_ij|| < d then
            choice_i is the least load in ||c_ij|| cloudlets;
        else if ||c_ij|| = 0 then
            skip task offloading for cloudlet i in this round;
        end if
        if f_{choice_i} ≥ f then
            cloudlet i performs task offloading to choice_i;
        end if
        s_{j,i} = s_{j,i} + 1, q_j = q_j + 1;
        update f_i and f_j;
        j = j + 1 (j ≠ i);
    end while
    t = t + 1;
end while
return task load q_i, offloading record S_i.
```

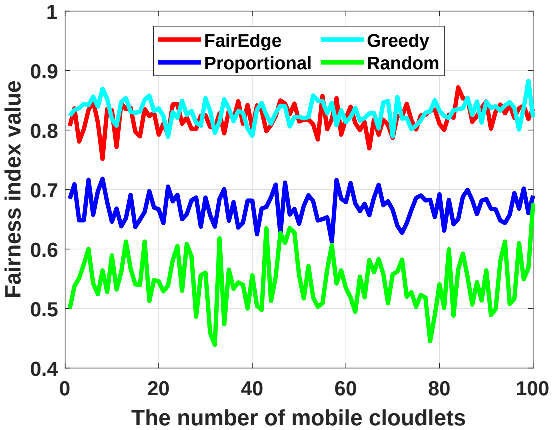


FIGURE 1: The overall fairness values of each mobile cloudlet in task offloading collaborations.

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.

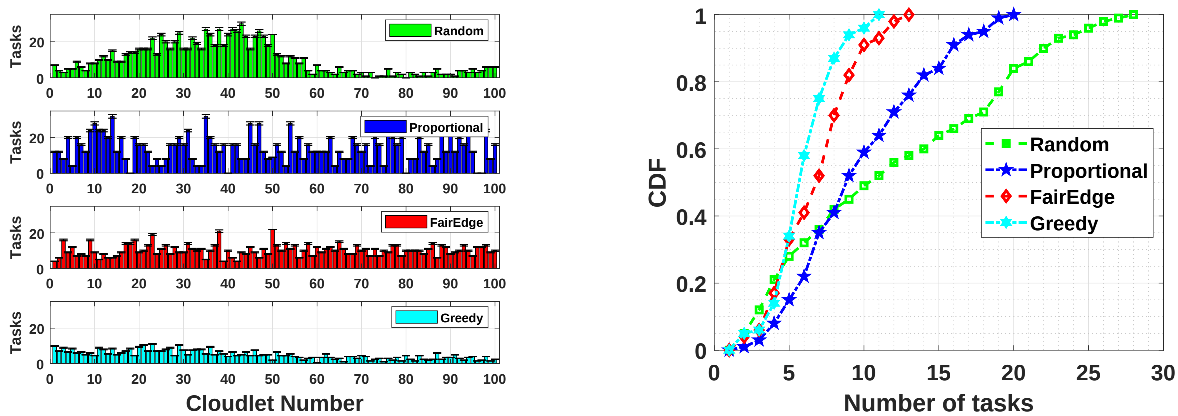
A. SIMULATION STUDY

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 [29]. The fairness-oriented task offloading scheme is simulated in a 20 km^2 region, 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 m. 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 [40] and greedy task offloading [41]. We run the simulation codes on a Dell laptop with Intel Core i5 CPU, 8GB RAM. Each simulation is executed for 20 times and we report the final average results as follows.

2) Evaluation on 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, most of fairness index values of mobile cloudlets under random offloading scheme and proportional offloading scheme are below 0.6 and 0.7, respectively. In comparison, the proposed FairEdge and greedy algorithm achieve an average value of fairness index of more than 0.8, showing that the task distribution is well-balanced across all mobile cloudlets. The greedy algorithm applies node traversal on each cloudlet



(a) Task distribution under different offloading methods

(b) CDF of task offloading results

FIGURE 2: Comparison of task offloading results in mobile cloudlets

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 cost. Meanwhile, the proposed FairEdge adopts balls-into-bins theory and 'd-choice' scheme in task offloading. It achieves a close-greedy performance in fairness values while reducing computation complexity to $o(1)$, as each cloudlet only needs to probe load information from two random neighbours and make 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 a network fairness up to 0.85 and reduce the unbalanced task offload by 50% in comparison with other baseline methods.

3) Evaluation on Task Distribution

Second, we evaluate the task distribution results under different task offloading algorithms. The bar plot of Figure 2(a) shows the final task distribution after all offloading time intervals. Obviously, under both random task offloading and proportional offloading schemes, there are 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 others 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 the cloudlets require 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.

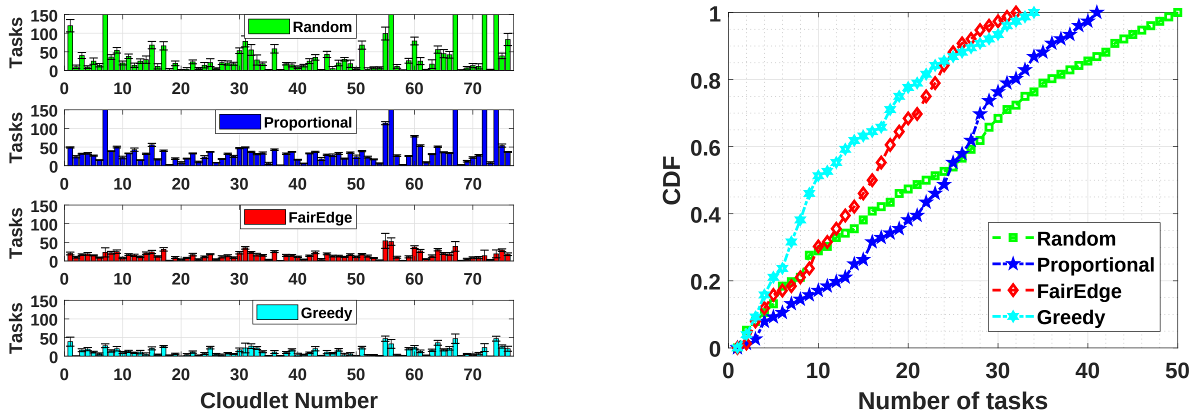
To make a further comparison, we present the empirical cumulative distribution results of task offloading in Figure 2(b). Here, the performance of FairEdge is very close to that of greedy offloading, where over 90% mobile cloudlets are offloaded with less than 10 tasks. In contrast, the task offloading result by proportional method shows that almost 20% cloudlets are offloading with more than 15 tasks. In addition, over 20% mobile cloudlets have more than 20 tasks to process under random offloading. Above evaluation results validate the effectiveness of FairEdge, as it manages to balance the task distribution by using fairness index and '2-choice' paradigm in task offloading. 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 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 of using two different trace datasets for evaluations are: 1) to test the performance of FairEdge in different network scenarios, where cloudlets have different pattern of mobility; 2) to examine the scalability of FairEdge with mobile cloudlet networks of different scales. We present the details of each dataset and corresponding evaluation results of mobile computation offloading in the following.

1) MobiClique Dataset

Basic Setups. We adopt a real-world mobility dataset called 'MobiClique' [42] to emulate a random mobility of mobile cloudlet 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 [43]. The trace data of MobiClique



(a) Task distribution in MobiClique dataset

(b) CDF of task offloading results in MobiClique dataset

FIGURE 4: Comparison of task offloading results in MobiClique dataset

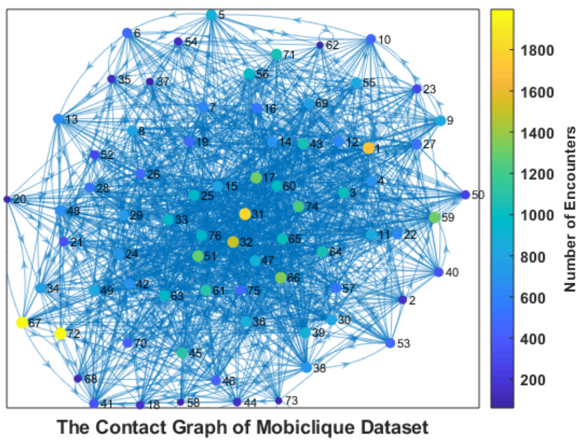


FIGURE 3: Nodes and their encountering records in MobiClique [42] dataset.

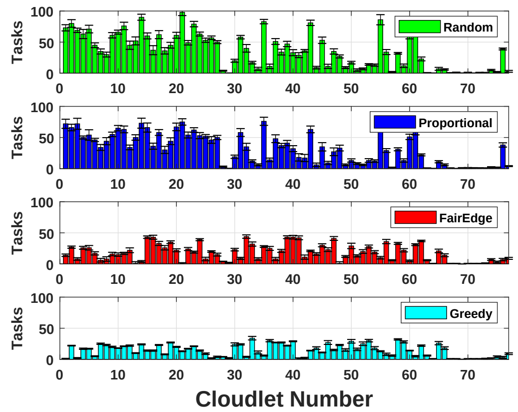
was collected with 76 participants during SIGCOMM 2009 conference in Barcelona, Spain. We process the MobiClique dataset as an edge mobile cloudlet network and visualize it in Figure 3. Here, each vertex represents a mobile cloudlet, an edge between two vertices represents a contact, and the color of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

Task Offloading Results. We conduct the mobile cloudlet task offloading with MobiClique 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(a), 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 individual cloudlet is under 50. In addition, by combining '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.

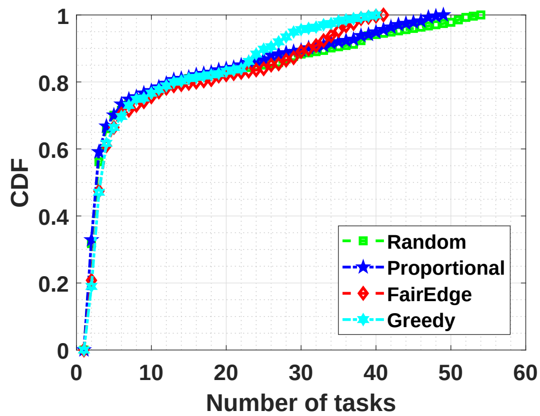
The empirical cumulative distribution of task offloading results with MobiClique dataset is presented in Figure 4(b). It is clear that more than 80% of mobile cloudlets in FairEdge and greedy schemes are offloaded with less than 25 tasks. Meanwhile, more than 20% of mobile cloudlets in random and proportional 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 real-world mobility trace dataset, where great disparity exists in the active level of different cloudlets.

2) Haggle dataset

Basic Setups. We further evaluate the performance of FairEdge in a larger mobility trace dataset, *i.e.*, Haggle dataset [44]. The Haggle dataset is under the project of Koblenz Network Collection (KONECT) [45] 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 Haggle dataset in Figure 5, where all 274 vertices are with



(a) Task distribution in Haggle dataset



(b) CDF of task offloading results in Haggle dataset

FIGURE 6: Comparison of task offloading results in Haggle dataset

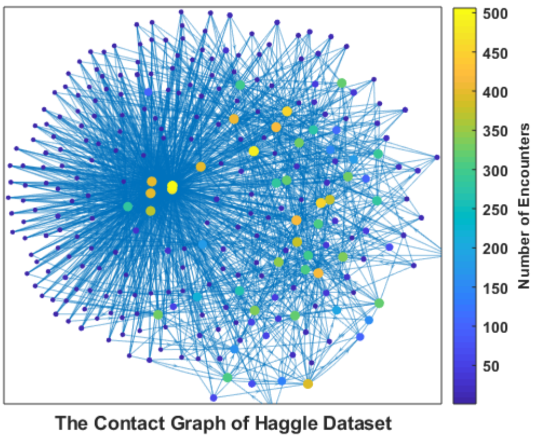


FIGURE 5: Nodes and their encountering records in Haggle [44] dataset.

28,244 edges. Similar to MobiClique 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 MobiClique dataset, the vertices in the Haggle contact graph are more distributed, where a small number of vertices form a 'contact center' (1-66) that links the rest edge vertices with sparse contacts. In the following evaluations, all basic setups are the same as evaluations with MobiClique dataset, except for the time interval. In Haggle dataset, the beginning and ending time stamps are 20,733 seconds and 364,094 seconds, respectively. As the overall duration in Haggle dataset is much longer than that of MobiClique, we set the length of time slot as 3,600 seconds (*i.e.*, 1 hour) for computation offloading simulation.

Task Offloading Results. In evaluations with 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), 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 1-66). As the cloudlet of number 80-274 have very few encounters with others, most of them are offloaded only several 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 several cloudlets are overloaded with nearly 100 tasks. While proportional task offloading slightly improves the task distribution result, still there exist huge gaps (over 90 tasks) among cloudlet in the contact center. In contrast, FairEdge and greedy task offloading schemes 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 preserve their fairness by re-balancing tasks from several overloaded cloudlets to others. The empirical cumulative distribution of task offloading results with 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.

VII. THE FUTURE DIRECTION

With the arrival of 5G, Mobile Edge Computing (MEC) is regarded as a promising technology for future communication systems, by bringing the computing and storage resources to the proximity to mobile users [46]. 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 [47]. Such problems are generally formulated with typical network settings and further solved with task offloading strategies based on heuristic algorithms [48]. However, for the coming 5G networks, a typical mobile edge node is expected to have more than 2000 configurable parameters [49]. In that sense, with the explosive growth of mobile data, existing heuristic algorithms are not capable and scalable to tame the complexity in highly dynamic computation offloading environment.

To address such 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 of solving combinatorial optimization problems in MEC task offloading, thereby significantly reducing the computational complexity [24]. For example, Huang *et al.* [50] 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.* [51] designed a DRL-based optimization framework with Q-learning for multi-use computation offloading and resource allocation in MEC. Moreover, Zhao *et al.* [52] leveraged a multi-LSTM based prediction model to assist task offloading strategy and improve the performance of edge computing systems. Ning *et al.* [53] 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 show 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 more 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 by mobile cloudlets. By adopting the 'two-choice' paradigm and comparing fairness indexes between the source cloudlet and the target cloudlet, We have developed the system model of computation offloading in edge mobile cloudlet networks and formally formulated the load balancing problem and fairness optimization problem. We have further developed the algorithm design of FairEdge and conducted intensive evaluations studies with both simulation and real-world mobility trace datasets. The experimental results show that, FairEdge successfully achieves load balancing with guaranteed performance, with a near-optimal fairness index of up to 0.85 and improvement of 50% in comparisons with conventional baseline methods.

ACKNOWLEDGMENT

This work was supported by the China Scholarship Council. Shuang Lai and Xiaochen Fan contributed equally to this work.

REFERENCES

[1] C. W. Paper, "Cisco visual networking index: Global mobile data traffic forecast update, 2017-2022," https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.pdf, 2019.
[2] J. Yu, Z. Kuang, B. Zhang, W. Zhang, D. Lin, and J. Fan, "Leveraging content sensitiveness and user trustworthiness to recommend fine-grained privacy settings for social image sharing," IEEE transactions on information forensics and security, vol. 13, no. 5, pp. 1317–1332, 2018.
[3] J. Yu, M. Tan, H. Zhang, D. Tao, and Y. Rui, "Hierarchical deep click feature prediction for fine-grained image recognition," IEEE transactions on pattern analysis and machine intelligence, 2019.
[4] 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 Computing Surveys (CSUR), vol. 51, no. 6, p. 111, 2019.
[5] K. Bilal, O. Khalid, A. Erbad, and S. U. Khan, "Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers," Computer Networks, vol. 130, pp. 94–120, 2018.
[6] M. Z. Nayyer, I. Raza, and S. A. Hussain, "A survey of cloudlet-based mobile augmentation approaches for resource optimization," ACM Computing Surveys (CSUR), vol. 51, no. 5, p. 107, 2018.
[7] Y. Yin, J. Xia, Y. Li, W. Xu, L. Yu et al., "Group-wise itinerary planning in temporary mobile social network," IEEE Access, vol. 7, pp. 83 682–83 693, 2019.
[8] H. Gao, Y. Duan, L. Shao, and X. Sun, "Transformation-based processing of typed resources for multimedia sources in the iot environment," Wireless Networks, pp. 1–17, 2019.
[9] Y. Yin, L. Chen, Y. Xu, J. Wan, H. Zhang, and Z. Mai, "QoS prediction for service recommendation with deep feature learning in edge computing environment," Mobile Networks and Applications, pp. 1–11, 2019.
[10] 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 of Things Journal, pp. 1–1, 2019.
[11] H. Gao, W. Huang, and X. Yang, "Applying probabilistic model checking to path planning in an intelligent transportation system using mobility trajectories and their statistical data," Intelligent Automation and Soft Computing, vol. 25, no. 3, pp. 547–559, 2019.
[12] M. Mitzenmacher and E. Upfal, Probability and computing: randomization and probabilistic techniques in algorithms and data analysis. Cambridge university press, 2017.
[13] G. Zhang, F. Shen, Y. Yang, H. Qian, and W. Yao, "Fair task offloading among fog nodes in fog computing networks," in 2018 IEEE International Conference on Communications (ICC). IEEE, 2018, pp. 1–6.
[14] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," IEEE Communications Surveys & Tutorials, vol. 19, no. 4, pp. 2322–2358, 2017.
[15] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," IEEE Communications Surveys & Tutorials, vol. 19, no. 3, pp. 1628–1656, 2017.
[16] J. Yu, C. Hong, Y. Rui, and D. Tao, "Multitask autoencoder model for recovering human poses," IEEE Transactions on Industrial Electronics, vol. 65, no. 6, pp. 5060–5068, 2017.
[17] J. Yu, C. Zhu, J. Zhang, Q. Huang, and D. Tao, "Spatial pyramid-enhanced netvlad with weighted triplet loss for place recognition," IEEE transactions on neural networks and learning systems, 2019.
[18] L. Gong, Y. Zhao, X. Chaocan, Z. Li, C. Qian, and P. Yang, "Robust lightweight magnetic-based door event detection with smartphones," IEEE Transactions on Mobile Computing, 2018.
[19] C. Chen, Y. Ding, X. Xie, S. Zhang, Z. Wang, and L. Feng, "Trajcompressor: An online map-matching-based trajectory compression framework leveraging vehicle heading direction and change," IEEE Transactions on Intelligent Transportation Systems, 2019.
[20] C. Chen, S. Jiao, S. Zhang, W. Liu, L. Feng, and Y. Wang, "Tripimputor: Real-time imputing taxi trip purpose leveraging multi-sourced urban data," IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 10, pp. 3292–3304, 2018.
[21] D. Liu, Z. Cao, Y. He, X. Ji, M. Hou, and H. Jiang, "Exploiting concurrency for opportunistic forwarding in duty-cycled iot networks," ACM Transactions on Sensor Networks (TOSN), vol. 15, no. 3, p. 31, 2019.
[22] D. Liu, Z. Cao, M. Liu, M. Hou, and H. Jinag, "Contention-detectable mechanism for receiver-initiated mac," ACM Transactions on Embedded Computing Systems (TECS), vol. 18, no. 4, p. 31, 2019.

[23] Y. Qu, S. Tang, C. Dong, P. Li, S. Guo, H. Dai, and F. Wu, "Posted pricing for chance constrained robust crowdsensing," IEEE Transactions on Mobile Computing, 2018.
[24] J. Wang, J. Pan, F. Esposito, P. Calyam, Z. Yang, and P. Mohapatra, "Edge cloud offloading algorithms: Issues, methods, and perspectives," ACM Computing Surveys (CSUR), vol. 52, no. 1, p. 2, 2019.
[25] X. Sun and N. Ansari, "Latency aware workload offloading in the cloudlet network," IEEE Communications Letters, vol. 21, no. 7, pp. 1481–1484, 2017.
[26] H. Huang and S. Guo, "Service provisioning update scheme for mobile application users in a cloudlet network," in 2017 IEEE International Conference on Communications (ICC). IEEE, 2017, pp. 1–6.
[27] H. Huang and G. Song, "Adaptive service provisioning for mobile edge cloud," ZTE Communications, vol. 15, no. 2, pp. 1–9, 2017.
[28] 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. 44 175–44 189, 2018.
[29] 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 2018 IEEE International Conference on Communications (ICC). IEEE, 2018, pp. 1–6.
[30] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," IEEE Journal on Selected Areas in Communications, vol. 36, no. 3, pp. 587–597, 2018.
[31] 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 Transactions on Communications, vol. 66, no. 4, pp. 1594–1608, 2018.
[32] 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.
[33] E. Meskar and B. Liang, "Fair multi-resource allocation with external resource for mobile edge computing," in IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2018, pp. 184–189.
[34] P. Berenbrink, K. Khodamoradi, T. Sauerwald, and A. Stauffer, "Balls-into-bins with nearly optimal load distribution," in Proceedings of the twenty-fifth annual ACM symposium on Parallelism in algorithms and architectures. ACM, 2013, pp. 326–335.
[35] M. Mitzenmacher, "The power of two choices in randomized load balancing," IEEE Transactions on Parallel and Distributed Systems, vol. 12, no. 10, pp. 1094–1104, 2001.
[36] 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 Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing. ACM, 2016, pp. 83–92.
[37] M. Jia, W. Liang, Z. Xu, and M. Huang, "Cloudlet load balancing in wireless metropolitan area networks," in IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications. IEEE, 2016, pp. 1–9.
[38] 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 and Computation: Practice and Experience, vol. 29, no. 14, p. e4121, 2017.
[39] M. Jia, W. Liang, Z. Xu, M. Huang, and Y. Ma, "Qos-aware cloudlet load balancing in wireless metropolitan area networks," IEEE Transactions on Cloud Computing, 2018.
[40] Q. Li, P. Yang, S. Tang, M. Zhang, and X. Fan, "Equilibrium is priceless: selfish task allocation for mobile crowdsourcing network," EURASIP Journal on Wireless Communications and Networking, vol. 2016, no. 1, p. 166, 2016.
[41] Q. Li, P. Yang, S. Tang, C. Xiang, and F. Li, "Many is better than all: Efficient selfish load balancing in mobile crowdsourcing systems," in 2015 Third International Conference on Advanced Cloud and Big Data. IEEE, 2015, pp. 1–6.
[42] A.-K. Pietilainen and C. Diot, "Social pocket switched networks," in IEEE INFOCOM Workshops 2009. IEEE, 2009, pp. 1–2.
[43] J. Lee, K. Lopatin, R. Hussain, and W. Nawaz, "Evolution of friendship: A case study of mobiclique," in Proceedings of the Computing Frontiers Conference. ACM, 2017, pp. 267–270.
[44] J. Kunegis, "Haggle network dataset konect," 2017. [Online]. Available: http://konect.uni-koblenz.de/networks/contact
[45] K. Jerome, "Konect: the koblenz network collection," in Proceedings of the 22nd International Conference on World Wide Web. ACM, 2013, pp. 1343–1350.
[46] L. Yang, H. Zhang, M. Li, J. Guo, and H. Ji, "Mobile edge computing empowered energy efficient task offloading in 5G," IEEE Transactions on Vehicular Technology, vol. 67, no. 7, pp. 6398–6409, 2018.
[47] 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 Communications Magazine, vol. 57, no. 5, pp. 64–69, 2019.
[48] 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 Transactions on Mobile Computing, vol. 15, no. 11, pp. 2669–2681, 2015.
[49] 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 communications, vol. 24, no. 5, pp. 175–183, 2017.
[50] L. Huang, S. Bi, and Y. J. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," IEEE Transactions on Mobile Computing, 2019.
[51] J. Li, H. Gao, T. Lv, and Y. Lu, "Deep reinforcement learning based computation offloading and resource allocation for mec," in 2018 IEEE Wireless Communications and Networking Conference (WCNC). IEEE, 2018, pp. 1–6.
[52] X. Zhao, K. Yang, Q. Chen, D. Peng, H. Jiang, X. Xu, and X. Shuang, "Deep learning based mobile data offloading in mobile edge computing systems," Future Generation Computer Systems, vol. 99, pp. 346–355, 2019.
[53] Z. Ning, P. Dong, X. Wang, M. S. Obaidat, X. Hu, L. Guo, Y. Guo, J. Huang, B. Hu, and Y. Li, "When deep reinforcement learning meets 5g vehicular networks: A distributed offloading framework for traffic big data," IEEE Transactions on Industrial Informatics, 2019.