IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 July 2023, accepted 1 September 2023, date of publication 13 September 2023, date of current version 28 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3314889

SURVEY

Bandit Learning-Based Distributed Computation in Fog Computing Networks: A Survey

HOA TRAN-DANG [1], **(Member, IEEE), KI-HYUP KWON** [1], **(Member, IEEE), AND DONG-SEONG KIM** [1,2], **(Senior Member, IEEE)**

[1]Department of Electronic Engineering, Kumoh National Institute of Technology, Gumi-si 39177, South Korea
[2]Industrial Academic Cooperation Foundation, Kumoh National Institute of Technology, Gumi-si 39177, South Korea

Corresponding author: Dong-Seong Kim (dskim@kumoh.ac.kr)

This work was supported in part by the Ministry of Science and ICT (MSIT), South Korea, under the Grand Information Technology Research Center Support Program supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP) under Grant IITP-2023-2020-0-01612; in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2022R1I1A1A01071058; and in part by the Korea Research Fellowship Program through the National Research Foundation of Korea (NRF) funded by MSIT under Grant RS-2023-00249687.

ABSTRACT Fog computing is a decentralized computing infrastructure that extends the capabilities of cloud computing closer to the edge of the network. In a fog computing network (FCN), computing resources, such as processing power, storage, and networking, are distributed at various points in the network, including edge devices, fog nodes (FNs) such as access points, gateways, and local servers. This architecture allows resource-limited end devices to offload part of their computational tasks to nearby FNs to achieve the reduced response delay services and energy efficiency. However, the high dynamics and complicated heterogeneity of fog computing environment in many application scenarios result in the uncertainty of network information that is raised as a critical challenge to design efficient computation offloading strategies. Meanwhile, existing solutions such as centralized optimization, matching and game theory-based decentralized offloading are inadequate to be adopted in this context because they require the perfect knowledge of system parameters. Considering as a promising approach to deal with the information uncertainty issues, bandit learning has used recently to develop distributed computation offloading (DCO) algorithms for the FCNs. In this paper, we aim at reviewing such of these state-of-the-art DCO solutions and elaborate their advantages and limitations. Additionally, we identify open research challenges and provide future directions for research in this area. Our survey shows that bandit learning is a promising approach for efficient computation offloading in fog computing, and we expect that future research will continue to explore its potential for improving the performance and energy efficiency of fog computing-enabled systems.

INDEX TERMS Fog computing network, distributed computation offloading, multi-armed bandit (MAB) learning, reinforcement learning, non-stationary bandit, contextual bandit, non-contextual bandit, resource allocation.

I. INTRODUCTION

A. CONTEXT AND MOTIVATION

The rise of advanced technologies such as the Internet of Things (IoT), wireless communication, and Artificial Intelligence (AI) has led to the emergence of computation-intensive mobile applications that require ultra low latency, such as real-time face recognition and augmented reality [1]. However, end devices with resource and computation limitations are unable to meet these requirements, creating a contradiction that requires new classes of alternative solutions.

So far, cloud computing still has been an essential solution to this problem because it provides powerful

The associate editor coordinating the review of this manuscript and approving it for publication was Rodrigo S. Couto.

resources to support the task computation efficiently through different on-demand services (i.e., PaaS, IaaS, SaaS) [2], [3]. However, cloud computing-based solutions are inadequate to satisfy the expected quality of service (QoS) and quality of experience (QoE) requirements for certain types of latency-sensitive applications because of the long physical distance between the end devices and the remote cloud servers, scarce spectrum resources, and intermittent network connectivity.

Fog computing is one such technology emerged to address this issue by moving communication, computing, control, and storage capabilities from the cloud to the edge of the network to support geographically distributed, latency sensitive, and QoS/QoE-aware IoT services and applications [4]. Originally, *fog* referred to as **F**rom c**O**re to ed**G**e computing was defined by Cisco to extend the cloud computing to the edge of the network [5]. Technically, fog computing is a highly virtualized platform integrated and deployed in the edge network devices (called fog nodes (FNs)) such as gateways, switches, and hubs providing computing, storage, and networking services between end devices and the cloud computing [6]. With this architecture, the resource-limited end devices can offload part or whole computation tasks to the nearby FNs to experience the low latency services [7] as well as achieve the energy efficiency [8]. In addition, fog computing networks (FCNs) formed by networking FNs can further provide high-performance computing services through resource sharing mechanisms and collaborative services architecture [9]. However, deciding when and where to offload computation tasks in fog computing environments is a challenging problem, as it involves considering multiple factors such as the availability and reliability of FNs, the energy consumption of mobile devices, and the latency requirements of applications [10].

There are a large number of centralized optimization techniques and algorithms proposed in the literature to provide optimal solutions to the aforementioned resource allocation problems [11]. However, these solutions require a centralized control to gather the global system information, thus incurring a significant overhead and computation complexity of algorithms. This complexity is further amplified by the rapidly increase of density and heterogeneity of FCNs [12] when dealing with combination integer programming problems [13].

The aforementioned limitations of optimization have lead to a second class of game theory based offloading solutions that can avoid the cost-intensive centralized resource management as well as substantially reduce the complexity of algorithms [14], [15]. In this type of algorithms, task offloading is modeled as a non-cooperative game between task nodes (TNs that are FNs having computation needs) and helper nodes (HNs that are FNs with spare computing resources) POMT [16] and POST [17]. To obtain the TN-to-HN pairing in distributed manner, each TN constructs its own best response function taking account into the performance objective (i.e., delay reduction) and the behaviors of other players using only local information. When the players (TNs) are mutually best responding, the system reach a Nash equilibrium (NE). Despite their potentials in terms of efficiency and low computation complexity, these approaches pose several limitations. First, classical game theoretical algorithms such as best response require some information regarding actions of other players [18]. Correspondingly, many assumptions are introduced in the game theory based algorithms to simplify the system models that, in some case, are impractical. Second, most game-theoretic solutions, for example, NE, investigate one-sided stability notions in which equilibrium deviations are evaluated unilaterally per player [19]. In addition, the stability must be achieved by both sides of players (i.e., resource providers and resource requesters) in the context of fog computing environment.

Ultimately, managing resource allocation effectively in such a complex fog computing environment leads to a fundamental shift from the traditional centralized mechanism toward distributed approaches. Recently, matching theory has emerged as a promising technique for solving offloading problems in the fog computing because it can alleviate the shortcomings of game theory and optimization-based approaches [20], [21], [22], [23]. Basically, matching theory provides mathematically tractable solutions for the combination problem of matching players in two distinct sets, depending on the individual information and preference of each player. Following the deferred acceptance (DA) procedure, the matching game between the two sides of players achieves the stability in a distributed manner [24]. Alternatively, the matching-based approaches have potential advantages over the optimization and game theory based solutions owing to the distributed and low computational complexity algorithm.

However, most of aforementioned solutions assume the information regarding the resource states of fog computing nodes are known *a priori*, which is not realistic in many practical applications. For example, the TNs are likely to be uncertain about the computing resources (i.e., CPU frequency, queuing delay) of HNs at time of offloading decision since it may be varying over time. Therefore, to efficiently offload the tasks in an online manner, the TNs must interact iteratively with the HNs to learn their unknown computing resource status. To address this problem, multi-armed bandit (MAB) learning has emerged as a promising technique for making optimal offloading decisions in fog computing environments. Bandit learning algorithms use a trial-and-error approach to learn the optimal offloading policy by iteratively exploring and exploiting the available options [25]. This approach can handle the dynamic and uncertain nature of fog computing environments and provide efficient and effective offloading decisions [26]. Bandit learning algorithms are a class of reinforcement learning (RL) algorithms that are designed to handle decision-making problems in which a decision-maker must repeatedly choose among multiple options with uncertain outcomes. In distributed computation scenarios, bandit learning can be used to allocate computing resources among multiple tasks and devices in an efficient manner. Bandit learning algorithms can dynamically learn from the results of previous decisions and adjust their resource allocation strategies accordingly, in order to optimize the overall performance of the system. Bandit learning has been used to develop efficient decision making strategies in a wide range of applications, including online advertising [27], recommender systems [28], job market matching [29], spectrum scheduling in wireless networks [30], task and job scheduling in autonomous systems [31]

B. CONTRIBUTIONS

The key contributions of paper are summarized as follows:

- This paper provides a fundamental concept and architecture of FCNs as well as states the generic computation offloading problem for the FCNs, that is inadequate to be solved by optimization based solutions with prior knowledge of network parameters assumed.
- This paper then aims at reviewing the state-of-the-art algorithms for DCO using bandit learning to deal with the information uncertainty of networks. The review also focuses on elaborating the advantages and limitations of proposed solutions.
- Additionally, we identify the open research challenges and future research directions, which have not been concerned in the existing literature.

C. OUTLINE OF PAPER

The rest of this paper is organized as follows. In Section II, we introduce the model of FCNs, typical computation offloading models, and generic optimization problem of computation offloading. IN Section III, we briefly introduce the concept of bandit learning and typical algorithms to solve the MAB problems. Section IV focuses on reviewing and elaborating the distributed computation offloading algorithms based on the bandit learning. Section V discus the remaining challenges regarding the development of bandit learning based offloading algorithms in different scenarios. Section VI explores the open research issues. Finally, Section VII summarizes and concludes the paper.

D. NOTATIONS

For the sake of readability, Table 1 summarizes the list of abbreviations adopted in this paper.

TABLE 1. List of Abbreviations.

Symbols	Definitions
IoT	Internet of Thing
CPS	Cyber-Physical System
FN	Fog Node
FCN	Fog Computing Network
VFC	Vehicular Fog Computing
MEC	Mobile Edge Computing
UDN	Ultra-Dense Network
WSN	Wireless Sensor Network
QoS	Qualit of Service
QoE	Quality of Experience
TN	Task Node
HN	Helper Node
UCB	Upper Bound Confidence
TS	Thompson Sampling
MAB	Multi-Armed Bandit
CMAB	Combinatorial MAB
BCO	Bandit Convex Optimization

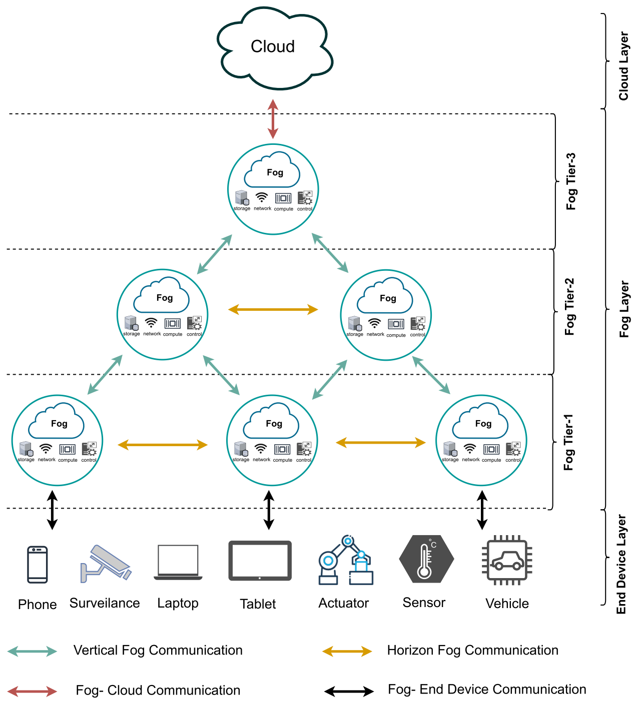


FIGURE 1. An illustrative N-tier model of FCNs integrated in IoT systems and CPS.

II. SYSTEM ARCHITECTURE AND PROBLEM FORMULATION

A. FOG COMPUTING NETWORK

A generic architecture of fog-based IoT systems and CPS can be viewed as a three layer structure as shown in Fig. 1. The lowest level of the hierarchy termed as end device layer includes all network-connected physical devices such as mobile phones, tablets, sensors, actuators, and vehicles. Their primary function is to detect various events and send the unprocessed data they sense to the upper layer in the hierarchy. The second layer placed between the edge and cloud computing layers, commonly referred to as the fog computing layer, consists of smart devices such as routers, gateways, switches, and access points, that have the ability to process, calculate, and store received data temporarily. Typically, there are multiple tier (N-tiers) of fog nodes (FNs) structured in the fog computing layer of IoT systems and CPSs to execute the applications [32]. FNs can connect together through wired and wireless links to form a mesh network, hereafter termed as fog computing network (FCN) that enables load balancing, fault tolerance, resilience, data sharing, and reduces cloud communication. Architecturally, this requires that FNs must have the capability to communicate horizontally (peer-to-peer or east-to-west) [33] and vertically (north-to-south) within the fog hierarchy. In addition, FNs must be able to find, trust, and utilize services provided by other nodes to maintain reliability, availability, and serviceability (RAS) [34]. These fog computing devices are connected to the cloud infrastructure and are tasked with periodically transmitting data to the cloud.

At the topmost tier of this architecture lies the cloud computing layer, which is comprised of numerous powerful servers and data centers with the capacity to process and store vast amount of data.

B. COMPUTATION OFFLOADING MODELS

There are many models introduced in the literature to perform the computational offloading operations in FCNs. Depending on the application scenarios, the models are established appropriately to support the systems to achieve a single objective or multiple objectives simultaneously such as minimization of total energy consumption, minimization of offloading delay, and maximization of resource utilization, and fairness and balance of workload. Fundamentally, an offloading model takes into account multiple factors including the system architecture, the task properties to derive efficient algorithms, that determine offloading locations, times to offload, and how a task is offloaded (how data of task is handled). In the following paragraphs, we summarize and discuss these relevant aspects to highlight the key features of popular offloading models in the literature.

Regarding the offloading locations, there are two major classes of offloading models including intra-layer and inter-layer offloading. The former refers to models that the offloading operations take place in the same layer, whereas the later involve multiple layers (e.g., between IoT and fog layer, between fog and cloud). Concretely, the computational offloading processes can take place only within a stratum of IoT-Fog-Cloud systems where the computing devices in the same tier (e.g., the IoT, fog, and cloud tier) can share their available resources to handle the tasks cooperatively. Recently, the advance of technologies can equip with modern IoT devices more features regarding powerful resource, computing capability to process tasks locally. In combination with the emergence of device-to-device (D2D) communication technologies, the computational offloading between IoT devices is pervasive in the future fog computing-enabled systems. In the same sense, the tasks can be offloaded within the fog layer and cloud layer, mainly to balance the workload as well as improve the resource utilization [35]. However, the heterogeneity of FN types exposes a challenge of communication between them. It requires unified middleware and protocols to enable fog-to-fog communication and collaboration such as FRAMES developed in [33] to jointly offloading the tasks. Otherwise, FNs can communicate via a centralized agent such as FSP or brokers in their fog domains.

In most of application scenarios, the offloading processes involve multiple layers. For example, as per [7], a task generated by an IoT device can be processed by itself locally or offload to a FN or the cloud finally. As demonstrated by simulation analysis, the offloading locations for tasks should be flexible with respect to the task type to get the benefit of offloading operations. Concretely, the heavy tasks should be offloaded to the cloud tier, while the medium tasks are processed by FNs. In addition, the light tasks can be computed locally by IoT devices if they have sufficient resource or offloaded to FNs, otherwise. As the tasks can be splittable, one part of task can be processed by IoT node and the other by the fog or cloud. Finally, there exist several application scenarios, in which the upper layers require the lower layers to execute the task. Theses uncommon offloading models include cloud offloading to fog/IoT and end user devices, fog offloading to the IoT and end user devices for specific purposes of applications [10].

The determination of times to offload tasks is an important aspect in the offloading models. Generally, offloading is needed when FNs are unable to process the tasks locally, or processing them may not satisfy the QoS requirements. Although the modern IoT devices and end user equipment can process some types of tasks locally, the majority of tasks (e.g., complex and heavy tasks, and sporadic tasks emergency cases) generated in the IoT layer are offloaded to the upper layers. However, the task offloading incurs additional cost such as communication delay and energy consumption. Therefore, the offloading model requires an inclusion of mechanism to monitor the system performance, traffic flow rates, network conditions that can support to make the offloading decisions appropriately. For example, the FOGPLAN framework in [36] can provide the dynamic offloading strategies to adapt to the dynamic change of QoS requirements. By observing and analyzing the task processing queue of FNs constantly, tasks currently resided in the processing queues of these FNs must be offloaded to HNs if the predicted processing delays are no longer to meet the deadlines of tasks. The network reliability is also concerned in the fog networks since it directly impacts on the communication delay of offloading processes [37].

The offloading models also specify how the input data of tasks is offloaded and processed. Generally, a full offloading method is applied for a task when its whole data is indivisible and processed by a single HN. Conversely, as a divisibility of task is enabled, a partial offloading scheme can be used to offload a fractional part of task to HNs while the other part of task is processed locally by TN. In the most of studies, a task is assumed to be decomposed into two subtasks, thus there needs only one HN to offload the subtask. As the subtasks are totally independent, the task division is an effective technique employed in the offloading models to cope with the heterogeneity of computing device resources, and simultaneously improve the performance of computing operations. For example, according to the FEMTO model in the work [38], each task is divided into two subtasks with different data sizes, which are then processed by the IoT node and offloaded to the fog entity respectively. This method contributes to minimizing the energy consumption of task offloading while achieving the workload fair among the fog nodes and satisfying the deadline constraints of tasks. Similarly, the partial offloading is utilized in the task offloading models for the heterogeneous fog networks to reduce the task execution delay [17]. Dividing a task into multiple (more than two) subtasks is also considered in [39] to exploit the parallel computation of subtasks at different FNs. As analyzed in [17], compared to the full offloading model, the partial offloading offers more advantages in terms of delay reduction, energy saving, resource utilization, and workload balancing. The independence of subtasks enabling the parallel processing of subtasks is obviously a key to achieve these advantage. However, in practice, some or all subtasks of a tasks can exist a data dependency relation. For example, the output of a subtask can be an input data for another subtask. Thus, completing the task requires a subtask scheduling plan to with respect to the subtak processing order. This in turn can impact the performance of partial offloading models. For instance, as evaluated and analyzed in [39], a number of subtasks for a task can be optimized depending on the system context. In addition, not all tasks should be divided because more subtasks can probably lead to a coupling resource problem. An offloading framework in FRATO is then introduced based on many factors such as the FN resource status (e.g., queue status, computing capability), task request rates, and task properties (e.g., divisibility) to offer a dynamic offloading policy.

As illustrated in Fig. 2, FRATO dynamically applies the partial offloading and full offloading modes for the tasks based on the queue status of FNs. In this way, FRATO is able to significantly reduce the offloading delay as well as improving the resource utilization, especially in cases of high rate of task requests.

A similar investigation is presented in [33] that considers three models of task processing, in which the subtasks can be executed in sequential, parallel, and mixed processing order.

C. OPTIMAL COMPUTATION OFFLOADING ISSUES

Denote $\mathbb{C} = \{\mathbb{C}_i, \mathbb{C}_j, \mathbb{C}_k, \ldots\}$ as the set of objective functions, established by individual computing nodes (i.e., end devices, FNs, or clouds) and by the system for the computational offloading performance at a given time. Some of typical objective functions concerned in the literature include total consumption energy [40], average task execution delay [41], total payment cost of resource usage [42], and fairness and workload balancing index [43]. Moreover, there also present objective functions of individual resources to indicate the inherent selfishness and rational of computing nodes. These kinds of objective functions are referred to as utility ones, which correspond to the benefits and revenues of available resource provision. Summarily, the generic optimization

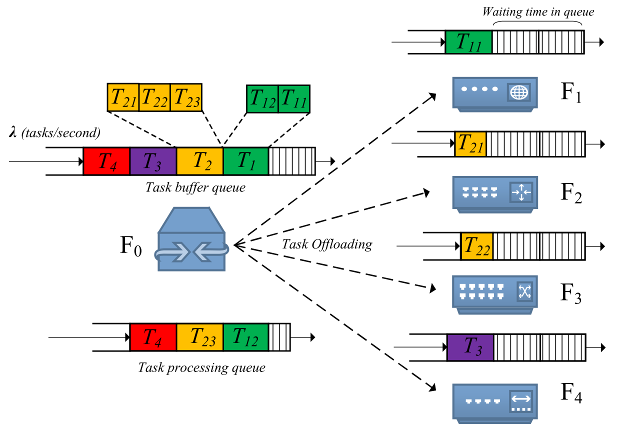


FIGURE 2. A dynamic computational offloading model is proposed in [39] that integrates partial and full offloading to balance the workload in the fog layer. The full offloading plan is used for task T_3, while the subtask T_{11} of T_1, subtask T_{21} and T_{22} of T_2 are offloaded partially by F_1, F_2 and F_3, respectively. T_4 is processed locally by F_0.

problem in FCNs can be represented in the following form:

$$\begin{aligned} P: &\quad \min(\mathbb{C}_i)\ \&/\ \max(\mathbb{C}_j)\ \&/\ \max(\mathbb{C}_k)\ \&/\ \ldots \\ \text{s. t.} &\quad \text{Constraints.} \end{aligned} \tag{1}$$

Depending on the application scenarios, the problem **P** can be in form of single or multi-objective model. Regardless the ultimate objectives of problems, the constraints involve the resource competition, resource limitations, and task scheduling. Concretely, a FN can receive multiple requests for task offloading. However, only a certain number of requests are accepted to be processed owing to the limitation of resource such as limited buffer capacity, low residual energy. Furthermore, scheduling the tasks in HNs is considered to respect to the QoS requirements. From the global point of view, the problem becomes a combinatorial problem, which is proven to be NP-hard due to the natural presence of coupling resource problems [44]. Therefore, achieving the globally optimized solution is infeasible, especially in the large-scale systems. In addition, there is an extensive cost of overhead to collect the global information. These issues urge the need to design the distributed algorithms to support the computational offloading processes efficiently.

III. BACKGROUNDS OF BANDIT LEARNING

A. BANDIT LEARNING CONCEPT

Bandit learning is a type of reinforcement learning that involves making decisions based on incomplete or uncertain information [45]. It is a trial-and-error approach where an agent learns to make decisions by exploring different options and evaluating their outcomes. The name "bandit" comes from the idea of a gambler at a slot machine (known as a "one-armed bandit") who tries different options and learns which one pays off the most [46].

In bandit learning, the agent selects an action based on the information available at that time, and observes the outcome of that action. The agent then updates its knowledge about the environment based on the observed outcome and selects the next action accordingly. The goal of bandit learning is to maximize the cumulative reward obtained by the agent over a series of decisions.

Bandit learning has been used to develop efficient decision making strategies in a wide range of applications, including online advertising, recommender systems, task and job scheduling in autonomous systems [30], [31]. In the context of fog computing, bandit learning algorithms can be used to make optimal offloading decisions by considering the available options and their potential outcomes. By iteratively exploring and exploiting the available options, bandit learning algorithms can provide efficient and effective offloading decisions in dynamic and uncertain fog computing environments [47].

B. BANDIT LEARNING ALGORITHMS

Bandit learning algorithms can be broadly categorized into two types: contextual and non-contextual.

Non-contextual bandit learning algorithms, also known as multi-armed bandit algorithms, involve selecting an action based solely on the current state of the environment, without considering any additional context or information. These algorithms explore different actions and evaluate their outcomes to gradually learn which actions provide the best rewards. Examples of non-contextual bandit learning algorithms include Epsilon-Greedy, Upper Confidence Bound (UCB), and Thompson Sampling (TS) [48].

Contextual bandit learning algorithms, on the other hand, consider additional contextual information when selecting an action [49], [50]. Contextual information includes features such as user preferences, device capabilities, and network conditions. These algorithms use this contextual information to make more informed decisions and improve the accuracy of their predictions.

Both non-contextual and contextual bandit learning algorithms have their advantages and limitations. Non-contextual bandit learning algorithms are simple and efficient, making them well-suited for applications where computational resources are limited. Contextual bandit learning algorithms, on the other hand, can provide more accurate and personalized recommendations by considering additional contextual information. The choice of algorithm depends on the specific application and the available resources of target systems.

IV. BANDIT LEARNING BASED DCO ALGORITHMS IN FCNs

Generally, the DCO in the FCNs can be modeled as a multi-armed bandit (MAB) or a multi-player MAB problem, in which TNs and HNs serve as players and arms respectively. This section emphasizes on summarizing and elaborating the key bandit learning-based algorithms developed in the literature for DCO in the FCNs. This review is divided into two parts according to the two main types of bandit learning (i.e., non-contextual and contextual) used to design the algorithms.

A. NON-CONTEXTUAL BANDIT LEARNING BASED DISTRIBUTED COMPUTATION

The principle of bandit learning techniques can be used in a family of bandit convex optimization (BCO) algorithms to solve convex optimization problems in which the objective functions and constraints are time varying [51], [52]. These similar features are prevalent in the computation offloading optimization problems of fog-based IoT systems where the function of accumulated network delay (need to be minimized) and long-term workload balancing constraints are variant over time. Based on this investigation, the studies [53] proposes a method for managing the task offloading problems in the large-scale and dynamic IoT systems using BCO. Concretely, a family of online bandit saddle-point (BanSaP) schemes are developed, which adaptively adjust the online operations based on (possibly multiple) bandit feedback of the loss functions, and the changing environment. The authors demonstrate the effectiveness of the proposed method through simulations, showing that it can simultaneously yields sublinear dynamic regret and fit in cases that the best dynamic solutions vary slowly over time. In particular, numerical experiments in the fog computing offloading tasks corroborate that the proposed BanSap approach offers competitive performance relative to existing approached based on gradient feedback.

The main focus in [54] is on offloading computing tasks in the context of IoT, where decision-making processes need to be able to adapt to changing user preferences and the unpredictable availability of resources. To address the challenges of such human-in-the-loop systems where loss functions are difficult to model, the authors developed a family of bandit online saddle-point (BanSP) schemes. These schemes adjust online operations based on bandit feedback of the loss functions and the changing environment. The paper evaluates the performance of BanSP by measuring dynamic regret, which is a generalization of static regret, and fit, which captures the cumulative amount of constraint violations. The authors prove that BanSP can simultaneously achieve sub-linear dynamic regret and fit, provided that the best dynamic solutions change slowly over time. Numerical tests on fog computing tasks demonstrate that BanSP performs well even with limited information.

Concerning the execution of hard real-time tasks within fixed deadlines in the IoT systems, the paper [55] introduces a two-tiered framework to offload the tasks using the fog and cloud computing instead of sensor nodes (SNs). To facilitate the task processing, a directed acyclic task graph (DATG) is formed by breaking down high-level tasks into smaller subtasks. The tasks are initially offloaded to a nearby FN using a greedy selection to avoid the combinatorial optimizations at the SNs, thus saving time and energy. As IoT environments are dynamic, adaptive solutions are necessary. An online learning scheme called ϵ-greedy nonstationary

MAB-based scheme (D2CIT) is proposed for task allocation among FNs. D2CIT enables the selection of a set of FNs for subtask distribution, parallel execution with minimum latency, energy, and resource usage. Simulation results show that D2CIT reduces latency by 17% and offers a speedup of 59% compared to existing online learning-based task offloading solutions in fog environments due to the induced parallelism.

The paper [56] develops an online task offloading strategy that minimizes the long-term cost that takes into account factors such as latency, energy consumption, and switching cost. To achieve this goal, a stochastic programming problem is formulated, with the expectation that the system parameters may abruptly change at unknown times. Additionally, the fact that queried nodes can only provide feedback on processing results after task completion is considered. To address these challenges, an effective bandit learning algorithm called BLOT is proposed to solve the non-stationary stochastic programming problem under a bandit model. The research also demonstrates the asymptotic optimality of BLOT in a non-stationary fog-enabled network and presents numerical experiments to justify the superior performance of proposed algorithm compared to the baseline approaches.

The paper [57] proposes a learning-based approach for task offloading in fog networks with the goal of reducing latency for delay-sensitive applications. The approach integrates Combinatorial MAB (CMAB) which is a generalization of the classical multi-armed bandit (MAB) problem to find the best set of arms to pull together, rather than finding the best single arm to pull [58]. Initially, the algorithm being suggested acquires knowledge of the shared computing resources of fog nodes, with minimal computational expenses. Next, the objective is to reduce the time taken for task offloading by simultaneously refining the task allocation decision and spectrum scheduling. Ultimately, simulation outcomes reveal that the proposed approach surpasses the conventional UCB algorithm with regards to delay performance and maintains extremely low offloading delays in a dynamically evolving system.

The authors in [59] also tackle the task offloading issues but consider the case of vehicular fog computing (VFC). The VFC environment with diverse modes of mobility introduces unpredictability with regards to the availability of resources and their demand, which create unavoidable obstacles in making optimal decisions for offloading. Moreover, these uncertainties pose additional challenges for task offloading in the face of an oblivious adversary attack and the risks associated with data privacy. The authors then have developed a novel algorithm for adversarial online learning with bandit feedback that leverages the principles of the adversarial MAB theory. This algorithm is designed to facilitate efficient and simple decision making for offloading by optimizing the selection of FNs, with the goal of minimizing costs associated with offloading services such as delay and energy usage. Fundamentally, the proposed approach involves implicitly adjusting the exploration bonus during selection, and incorporating assessment rules that account for the volatile nature of resource supply and demand. Theoretically, the input-size dependent selection rule allows for the selection of an appropriate FN without the need to explore sub-optimal actions. Additionally, the appropriate score patching rule facilitates quick adaptation to changing circumstances, reducing variance and bias, and ultimately achieving a better balance between exploitation and exploration. Simulation results demonstrate the effectiveness and robustness of the proposed algorithm.

The authors in [60] consider the task offloading in the fog computing network with time-varying stochastic time of arrival tasks and channel conditions. Due to the unavailability of global knowledge of all fog nodes in practice, the problem is modeled as a combinatorial multi-armed bandit (CMAB) problem, without prior information about channel conditions and stochastic task arrival characteristics. To address this problem, the paper proposes the WFCUCB algorithm, which extends the classical CMAB problem to include one *i.i.d.* variable and one non-stationary random variable. The paper's numerical results demonstrate that the WFCUCB algorithm is capable of fast learning and achieves superior performance compared to other possible strategies.

Due to the highly dynamic environment of the vehicular network, it is challenging to ensure that task offloading delay is minimized. To address this issue, task replication is introduced into the VEC system as proposed in the study [61], where multiple replicas of a task are offloaded simultaneously to several vehicles, and the task is considered completed once the first response among the replicas is received. The impact of the number of task replicas on the offloading delay is examined, and the optimal number of task replicas is determined through a closed-form approximation. Using these findings, a learning-based task replication algorithm (LTRA) is developed using CMAB theory. The LTRA algorithm is designed to operate in a distributed manner and can adapt automatically to the VEC system's dynamics. The proposed algorithm's delay performance is evaluated using a realistic traffic scenario. The results show that, under the simulation settings, the optimized LTRA algorithm with a specific number of task replicas can decrease the average offloading delay by more than 30% compared to the benchmark without task replication while also improving the task completion ratio from 97% to 99.6%.

The authors [62] also study the task offloading in the VFC systems with uneven workload distribution and the reliability of the communication between the FNs. In the work, they utilized the concept of CMAB to facilitate the selection of task offloading destinations in a distributed manner, without overburdening system resources. This is achieved by replicating tasks across multiple destination nodes and selecting the optimal number of replicating nodes to ensure reliability and minimize delay in a vehicular resource-sharing environment. This approach also reduces overall system residence time and enhances task delivery ratio by reducing task failures. Additionally, redundant tasks are eliminated from node queues after receiving the first response from candidate nodes in the surrounding area. They compared their solution with other baseline approaches based on several performance metrics, such as task residence time, end-to-end delays, delivery rate, and utilization ratio. The simulation results demonstrate that the proposed learning-based task offloading solution effectively utilizes resources and ensures its effectiveness over other approaches compared to the algorithms presented in [61].

In the work [63], the authors present a new online algorithm that formulates vehicular task offloading as a mortal multi-armed bandit problem, enabling distributed decision making on node selection. The algorithm leverages contextual information of edge nodes and transforms the exploration space from infinite to finite. Theoretical analysis shows that the proposed algorithm has sublinear learning regret, and simulation results confirm its efficacy.

B. CONTEXTUAL BANDIT LEARNING BASED DISTRIBUTED COMPUTATION

The study presented in [64] addresses the problem of data offloading in heterogeneous and dynamic fog computing-enabled wireless sensor networks. The authors model the data offloading problem as a contextual MAB problem that uses the heterogeneity of sensor nodes (SNs) as contextual information. The proposed algorithm for dynamic node movement in urban environments has been enhanced to ensure stable performance of the collaborative system despite the complexities and changes of the urban environment. By analyzing and simulating human movement data in such settings, the proposed approach can effectively minimize offloading delay and increase the success rate of offloading.

The work [65] studies the optimal computation offloading problem in the mobile edge computing (MEC) integrated ultra-dense networks (UDNs). proposes a new algorithm for task offloading in ultra-dense networks. The typical approaches for distributing tasks among multiple users involve a central node that makes decisions regarding which server to use and how to allocate resources. However, as the number of users increases, this method becomes excessively complicated, requiring significant communication overhead and intricate global optimization procedures. This paper introduces a new approach for distributing tasks among users in a UDN by allowing them to make local task offloading decisions independently. The goal is to minimize the average delay of tasks among all users, which is achieved by formulating an optimization problem. To accomplish this, a novel algorithm called Calibrated Contextual Bandit Learning (CCBL) is developed. This algorithm enables users to learn the computational delay functions of micro base stations and predict the task offloading decisions of other users in a decentralized manner. The convergence of the algorithm is verified using the approachability theory. Additionally, a user-oriented version of the algorithm is proposed to decrease computational complexity and increase the convergence rate. Simulation results demonstrate that the proposed algorithm outperforms existing decentralized algorithms and approaches the performance of centralized methods.

V. CHALLENGES OF BANDIT LEARNING APPLICATIONS FOR DISTRIBUTED COMPUTATION IN FOG COMPUTING

The overviewed studies in the literature show the effectiveness of using the principle of bandit learning to design the distributed computation algorithms in the fog computing. However, there are still existing many challenges needed to be tackled in variety of fog computing scenarios.

A. UNCERTAINTY

Inherently, fog computing environments are highly dynamic and uncertain, which makes it challenging to predict the resource availability and reliability of FNs. This uncertainty can affect the accuracy of bandit learning algorithms and make it difficult to make optimal offloading decisions. In addition, uncertainty is caused from multiple sources such as mobility of FNs, communication reliability between FNs, operational reliability of FNs. However, there are no works presented in the literature capturing the all dimensions of uncertainty to design the efficient algorithms.

B. RESOURCE CONSTRAINTS

Fog nodes have limited computational and storage resources, which can affect the performance and efficiency of bandit learning algorithms. Therefore, it is important to design bandit learning algorithms that can handle these constraints while still providing accurate and efficient offloading decisions.

C. COMMUNICATION OVERHEAD

Offloading involves exchanging data and information between mobile devices and fog nodes, which can result in significant communication overhead. This overhead can affect the performance and efficiency of offloading, especially in large-scale fog computing environments.

D. HETEROGENEITY

Fog computing environments are highly diverse and heterogeneous, with different types of devices, sensors, and fog nodes. This heterogeneity can affect the accuracy and efficiency of bandit learning algorithms, as different devices and nodes may have different capabilities and characteristics.

E. SECURITY AND PRIVACY

Offloading involves sharing sensitive data and information across different devices and fog nodes, which can raise security and privacy concerns. Therefore, it is important to design bandit learning algorithms that can handle these concerns and ensure secure and private offloading.

Addressing these challenges requires developing new and innovative bandit learning algorithms that can handle the dynamic and heterogeneous nature of fog computing environments while still providing accurate and efficient offloading decisions. Additionally, interdisciplinary research is needed to address the security, privacy, and communication challenges associated with offloading. Overall, overcoming

TABLE 2. Summarization of bandit learing-based DCO algorithms in the FCNs.

Study	Key Features of Algorithms	Results	Pros(+) & Cons(-)
[53], [54]	• Online bandit saddle-point (BanSaP) • Adaptively adjusting online offloading	• Sublinear dynamic regret and fit • Low network cost • Reduced latency	(+) Competitive performance relative to existing approach (+) Lower computation complexity (-) Best performance if dynamic environment vary slowly (-) Small network size
[55]	• Parallel computation for DATG tasks • ϵ-greedy to select FNs for offloading	• Latency reduction 15% • Improved speedup 59% • Energy efficiency • Resource usage reduction	(+) Low computation complexity (+) Large network size (-) Network environment is static and unchangable (-) Resource conflict issues dot not be solved
[56]	• Stochastic modeling of network • Dicounted UCB for online learning	• Long-term cost reduction • Latency reduction • Energy efficiency	(+) Considering dynamic of network environment (+) Regret bounded (-) Small network size (1 TN and 9 HNs)
[57]	• CMAB • Stochastic modeling of network • Joinly optimization of task allocation and spectrum scheduling	• Offloading latency minimization	(+) Considering high dynamic change of environments (+) Better performance that UCB (-) Small network size
[59]	• Adversarial online learning with bandit feedback	• Offloading delay minimization • Offloading energy minimization	(+) Considering dynamic of environment (+) Scalable & low complexity (-) No considering system-level regret-optimal aspect
[60]	• CMAB with UCB • Vayrant of tasks and channel condition • Learning fast	• Low latency compared to random, fixed offloading • Near optimal performance	(+) Considering dynamic of network environment (+) Low complexity of algorithm (+) Large network size (-) Resource conflicts do not be solved
[62]	• CMAB with discounted UCB • Replicating tasks across multiple HNs	• Low residence time • Enhance task delivery ratio	(+) Consider information uncertainty of system (+) Low complexity (+) Fast convergence (-) More overhead due to task replication (-) Small network size
[63]	• Mortal MAB • Incorporating contextual information into candidate arm • Transform the infinite exploration space to a finite one	• Sublinear learning regret • Low offloading latency	(+) Considering the change of arms (HNs) (+) Low complexity (-) Small network size (-) No considering the resource conflict
[64]	• Contextual MAB • Linear contextual bandit data offloading algorithm (LCBOD) based on the LinUCB	• Minimize offloading delay • Increase the success rate of offloading	(+) Considering high dynamics of networks (+) Low complexity (+) Large network size (-) Reward distribution of arms (HNs) known (-) More overhead due to more control information exchanged
[65]	• Context MAB • Calibrated learning	• Minimize long-term offloading delay • Increase the success rate of offloading	(+) Considering high dynamics of networks (+) Low complexity (+) Large network size (+) Fast converge rate (-) More overhead due to more control information exchanged

these challenges is essential for realizing the potential benefits of distributed computation offloading in fog computing using bandit learning.

VI. OPEN RESEARCH ISSUES

A. THOMPSON SAMPLING FOR BANDIT LEARNING

Generally, the empirical results demonstrates that the TS-based bandit learning algorithm is better than ϵ-greedy and UCB algorithm in a long run [66]. However, there is no existing work in the literature investigating the power of TS technique in the bandit learning algorithm in the context of fog computing.

B. BANDIT LEARNING WITH TWO-SIDED MATCHING

Matching theory has been investigated to be design the efficient distributed algorithm for task computation in the fog networks [21], [22], [23]. In addition, the matching theory can be used to handle the resource conflict problems, where multiple players attempt to pull an arm in the multi-player MAB problems (see Fig. 3). The primary simulation results as presented in [67] show that the combination of matching theory and TS-based bandit learning can improve the fog-based system performance in terms of response delay compared to the ϵ-greedy and UCB algorithms.

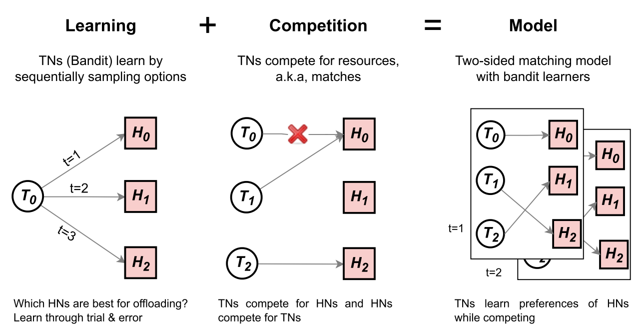


FIGURE 3. Bandit learning based two-sided matching.

An emerging line of research is dedicated to the problem of one-to-one matching markets with bandits, where the preference of one side is unknown and thus we need to match while learning the preference through multiple rounds of interaction [29]. However, in many real-world applications of fog computing scenarios, a FN can serve multiple tasks requested from other FNs, which motivates the study of the many-to-one matching problem. For example, the TS-based bandit learning can be used in many-to-one matching problem in the matching market [68], [69].

C. ROBUSTNESS AND SCALABILITY

Bandit learning algorithms need to be robust and scalable to handle the dynamic and heterogeneous nature of fog computing environments. This requires developing new algorithms that can handle large-scale and complex decision-making problems while maintaining high accuracy.

D. SECURITY AND PRIVACY

Computation offloading involves sharing sensitive data and information across different devices and fog nodes. Therefore, it is essential to develop bandit learning algorithms that can handle security and privacy concerns, such as protecting sensitive data and ensuring secure communication between devices and nodes.

E. HETEROGENEITY AND DIVERSITY

Fog computing environments are highly diverse and heterogeneous, with different types of devices, sensors, and fog nodes. Therefore, bandit learning algorithms need to be designed to handle this diversity and heterogeneity to make optimal offloading decisions.

F. TRADE-OFF BETWEEN ACCURACY AND RESOURCE CONSUMPTION

Bandit learning algorithms require significant computational resources to make accurate decisions. Therefore, there is a trade-off between accuracy and resource consumption that needs to be addressed.

G. REAL-TIME DECISION-MAKING

In fog computing environments, decisions need to be made quickly to ensure timely and efficient offloading. Therefore, bandit learning algorithms need to be designed to handle real-time decision-making to minimize delays and maximize the efficiency of offloading.

VII. CONCLUSION

In conclusion, this survey paper has provided a comprehensive overview of the use of bandit learning methods to develop the distributed computation offloading algorithms for the fog computing networks. We have reviewed the state-of-the-art techniques for computation offloading decision-making and highlighted the advantages and limitations of using bandit learning in this context. Our survey has shown that bandit learning algorithms can effectively address the dynamic and heterogeneous nature of fog computing environments, leading to improved performance and energy efficiency of mobile devices. However, we have also identified several open research challenges, such as the need for more robust and scalable bandit learning algorithms and the integration of security and privacy considerations in offloading decisions. Overall, the potential benefits of bandit learning for distributed computation offloading in fog computing are significant, and we believe that future research in this area will continue to advance the state-of-the-art and enable the development of more efficient and effective fog computing systems.

REFERENCES

[1] K. H. Abdulkareem, M. A. Mohammed, S. S. Gunasekaran, M. N. Al-Mhiqani, A. A. Mutlag, S. A. Mostafa, N. S. Ali, and D. A. Ibrahim, "A review of fog computing and machine learning: Concepts, applications, challenges, and open issues," *IEEE Access*, vol. 7, pp. 153123–153140, 2019.

[2] B. P. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *Proc. 5th Int. Joint Conf. INC, IMS, IDC*, Aug. 2009, pp. 44–51.

[3] A. Botta, W. de Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and Internet of Things: A survey," *Future Gener. Comput. Syst.*, vol. 56, pp. 684–700, Mar. 2016.

[4] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[5] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. 1st, Ed., MCC Workshop Mobile Cloud Comput.* New York, NY, USA: ACM Press, Aug. 2012, pp. 13–16.

[6] M. Aazam and E.-N. Huh, "Fog computing: The cloud-IoTIoE middleware paradigm," *IEEE Potentials*, vol. 35, no. 3, pp. 40–44, May 2016.

[7] A. Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue, "On reducing IoT service delay via fog offloading," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 998–1010, Apr. 2018.

[8] S. Sarkar and S. Misra, "Theoretical modelling of fog computing: A green computing paradigm to support IoT applications," *IET Netw.*, vol. 5, no. 2, pp. 23–29, Mar. 2016.

[9] S. Sarkar, S. Chatterjee, and S. Misra, "Assessment of the suitability of fog computing in the context of Internet of Things," *IEEE Trans. Cloud Comput.*, vol. 6, no. 1, pp. 46–59, Jan. 2018.

[10] M. Aazam, S. Zeadally, and K. A. Harras, "Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities," *Future Gener. Comput. Syst.*, vol. 87, pp. 278–289, Oct. 2018.

[11] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 283–294, Feb. 2018.

[12] G. Lee, W. Saad, and M. Bennis, "An online optimization framework for distributed fog network formation with minimal latency," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2244–2258, Apr. 2019.

[13] K. Guo, M. Sheng, T. Q. S. Quek, and Z. Qiu, "Task offloading and scheduling in fog RAN: A parallel communication and computation perspective," *IEEE Wireless Commun. Lett.*, vol. 9, no. 2, pp. 215–218, Feb. 2020.

[14] D. Monderer and L. S. Shapley, "Potential games," *Games Econ. Behav.*, vol. 14, no. 1, pp. 124–143, May 1996.

[15] Q. D. Lã, Y. H. Chew, and B.-H. Soong, *Potential Game Theory*. Cham, Switzerland: Springer, 2016.

[16] Y. Yang, Z. Liu, X. Yang, K. Wang, X. Hong, and X. Ge, "POMT: Paired offloading of multiple tasks in heterogeneous fog networks," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8658–8669, Oct. 2019.

[17] Z. Liu, Y. Yang, K. Wang, Z. Shao, and J. Zhang, "POST: Parallel offloading of splittable tasks in heterogeneous fog networks," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3170–3183, Apr. 2020.

[18] S. Durand and B. Gaujal, "Complexity and optimality of the best response algorithm in random potential games," in *Proc. Symp. Algorithmic Game Theory (SAGT)*, Sep. 2016, pp. 40–51.

[19] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.

[20] Z. Liu, X. Yang, Y. Yang, K. Wang, and G. Mao, "DATS: Dispersive stable task scheduling in heterogeneous fog networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3423–3436, Apr. 2019.

[21] H. Tran-Dang and D.-S. Kim, "A review on matching-based models for distributed computation offloading in fog-enabled IoT systems," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2023, pp. 758–763.

[22] H. Tran-Dang and D.-S. Kim, "DISCO: Distributed computation offloading framework for fog computing networks," *J. Commun. Netw.*, vol. 25, no. 1, pp. 121–131, Feb. 2023.

[23] H. Tran-Dang and D.-S. Kim, "A survey on matching theory for distributed computation offloading in IoT-Fog-Cloud systems: Perspectives and open issues," *IEEE Access*, vol. 10, pp. 118353–118369, 2022.

[24] Y. Gu, W. Saad, M. Bennis, M. Debbah, and Z. Han, "Matching theory for future wireless networks: Fundamentals and applications," *IEEE Commun. Mag.*, vol. 53, no. 5, pp. 52–59, May 2015.

[25] T. Lattimore and C. Szepesvári, *Bandit Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2020.

[26] H. Tran-Dang, S. Bhardwaj, T. Rahim, A. Musaddiq, and D.-S. Kim, "Reinforcement learning based resource management for fog computing environment: Literature review, challenges, and open issues," *J. Commun. Netw.*, vol. 24, no. 1, pp. 83–98, Feb. 2022.

[27] T. Geng, X. Lin, H. S. Nair, J. Hao, B. Xiang, and S. Fan, "Comparison lift: Bandit-based experimentation system for online advertising," in *Proc. 35th AAAI Conf. Artif. Intell., 33rd Conf. Innov. Appl. Artif. Intell., 11th Symp. Educ. Adv. Artif. Intell. (EAAI)*. Palo Alto, CA, USA: AAAI Press, Feb. 2021, pp. 15117–15126. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/17775

[28] C. Yan, J. Xian, Y. Wan, and P. Wang, "Modeling implicit feedback based on bandit learning for recommendation," *Neurocomputing*, vol. 447, pp. 244–256, Aug. 2021.

[29] L. T. Liu, F. Ruan, H. Mania, and M. I. Jordan, "Bandit learning in decentralized matching markets," *J. Mach. Learn. Res.*, vol. 22, no. 1, pp. 9612–9645, Jan. 2021.

[30] F. Li, D. Yu, H. Yang, J. Yu, H. Karl, and X. Cheng, "Multi-armed-bandit-based spectrum scheduling algorithms in wireless networks: A survey," *IEEE Wireless Commun.*, vol. 27, no. 1, pp. 24–30, Feb. 2020.

[31] D. Bouneffouf, I. Rish, and C. Aggarwal, "Survey on applications of multi-armed and contextual bandits," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2020, pp. 1–8.

[32] H. Gupta, A. V. Dastjerdi, S. K. Ghosh, and R. Buyya, "IFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, edge and fog computing environments," *Softw., Pract. Exp.*, vol. 47, no. 9, pp. 1275–1296, Sep. 2017.

[33] M. Al-Khafajiy, T. Baker, H. Al-Libawy, Z. Maamar, M. Aloqaily, and Y. Jararweh, "Improving fog computing performance via Fog-2-Fog collaboration," *Future Gener. Comput. Syst.*, vol. 100, pp. 266–280, Nov. 2019.

[34] *IEEE Standard for Adoption of OpenFog Reference Architecture for Fog Computing*, IEEE Standard 1934-2018, 2018, pp. 1–176.

[35] J. Contreras-Castillo, S. Zeadally, and J. A. Guerrero-Ibañez, "Internet of Vehicles: Architecture, protocols, and security," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3701–3709, Oct. 2018.

[36] A. Yousefpour, A. Patil, G. Ishigaki, I. Kim, X. Wang, H. C. Cankaya, Q. Zhang, W. Xie, and J. P. Jue, "FOGPLAN: A lightweight QoS-aware dynamic fog service provisioning framework," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5080–5096, Jun. 2019.

[37] J. Yao and N. Ansari, "Fog resource provisioning in reliability-aware IoT networks," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8262–8269, Oct. 2019.

[38] G. Zhang, F. Shen, Z. Liu, Y. Yang, K. Wang, and M.-T. Zhou, "FEMTO: Fair and energy-minimized task offloading for fog-enabled IoT networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4388–4400, Jun. 2019.

[39] H. Tran-Dang and D.-S. Kim, "FRATO: Fog resource based adaptive task offloading for delay-minimizing IoT service provisioning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 10, pp. 2491–2508, Oct. 2021.

[40] Z. Chang, Z. Zhou, T. Ristaniemi, and Z. Niu, "Energy efficient optimization for computation offloading in fog computing system," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–6.

[41] M. Mukherjee, S. Kumar, C. X. Mavromoustakis, G. Mastorakis, R. Matam, V. Kumar, and Q. Zhang, "Latency-driven parallel task data offloading in fog computing networks for industrial applications," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 6050–6058, Sep. 2020.

[42] A. Hazra and T. Amgoth, "CeCO: Cost-efficient computation offloading of IoT applications in green industrial fog networks," *IEEE Trans. Ind. Informat.*, vol. 18, no. 9, pp. 6255–6263, Sep. 2022.

[43] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1171–1181, Dec. 2016.

[44] T. Wang, Y. Liang, W. Jia, M. Arif, A. Liu, and M. Xie, "Coupling resource management based on fog computing in smart city systems," *J. Netw. Comput. Appl.*, vol. 135, pp. 11–19, Jun. 2019.

[45] G. Burtini, J. Loeppky, and R. Lawrence, "A survey of online experiment design with the stochastic multi-armed bandit," 2015, *arXiv:1510.00757*.

[46] A. Slivkins, "Introduction to multi-armed bandits," *Found. Trends Mach. Learn.*, vol. 12, nos. 1–2, pp. 1–286, 2019.

[47] D. H. Abdulazeez and S. K. Askar, "Offloading mechanisms based on reinforcement learning and deep learning algorithms in the fog computing environment," *IEEE Access*, vol. 11, pp. 12555–12586, 2023.

[48] J. Vermorel and M. Mohri, "Multi-armed bandit algorithms and empirical evaluation," in *Proc. 16th Eur. Conf. Mach. Learn. Mach. Learn. (ECML)*. Porto, Portugal: Springer, Oct. 2005, pp. 437–448.

[49] L. Zhou, "A survey on contextual multi-armed bandits," 2015, *arXiv:1508.03326*.

[50] W. Chu, L. Li, L. Reyzin, and R. Schapire, "Contextual bandits with linear payoff functions," in *Proc. 14th Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2011, pp. 208–214.

[51] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: Gradient descent without a gradient," in *Proc. 16th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA)*, 2004, pp. 385–394.

[52] W. Kumagai, "Introduction to bandit convex optimization algorithms," in *Proc. Int. Symp. Inf. Theory Appl. (ISITA)*, Oct. 2018, pp. 36–39.

[53] T. Chen and G. B. Giannakis, "Bandit convex optimization for scalable and dynamic IoT management," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 1276–1286, Feb. 2019.

[54] T. Chen and G. B. Giannakis, "Harnessing bandit online learning to low-latency fog computing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 6418–6422.

[55] S. Misra, S. P. Rachuri, P. K. Deb, and A. Mukherjee, "Multiarmed-bandit-based decentralized computation offloading in fog-enabled IoT," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 10010–10017, Jun. 2021.

[56] Z. Zhu, T. Liu, Y. Yang, and X. Luo, "BLOT: Bandit learning-based offloading of tasks in fog-enabled networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 12, pp. 2636–2649, Dec. 2019.

[57] K. Wang, Y. Tan, Z. Shao, S. Ci, and Y. Yang, "Learning-based task offloading for delay-sensitive applications in dynamic fog networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11399–11403, Nov. 2019.

[58] W. Chen, Y. Wang, and Y. Yuan, "Combinatorial multi-armed bandit: General framework and applications," in *Proc. 30th Int. Conf. Mach. Learn.* (Proceedings of Machine Learning Research), vol. 28, no. 1, S. Dasgupta and D. McAllester, Eds., 2013, pp. 151–159. [Online]. Available: https://proceedings.mlr.press/v28/chen13a.html

[59] B. Cho and Y. Xiao, "Learning-based decentralized offloading decision making in an adversarial environment," *IEEE Trans. Veh. Technol.*, vol. 70, no. 11, pp. 11308–11323, Nov. 2021.

[60] H. Zhu, K. Kang, X. Luo, and H. Qian, "Online learning for computation peer offloading with semi-bandit feedback," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 4524–4528.

[61] Y. Sun, S. Zhou, and Z. Niu, "Distributed task replication for vehicular edge computing: Performance analysis and learning-based algorithm," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1138–1151, Feb. 2021.

[62] B. Shabir, A. W. Malik, A. U. Rahman, M. A. Khan, and Z. Anwar, "A reliable learning based task offloading framework for vehicular edge computing," in *Proc. 2nd Int. Conf. Digit. Futures Transformative Technol. (ICoDT2)*, May 2022, pp. 1–6.

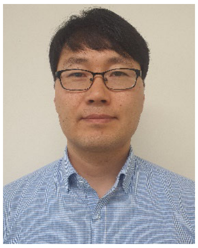
[63] R. Zhang, P. Cheng, Z. Chen, S. Liu, Y. Li, and B. Vucetic, "Online learning enabled task offloading for vehicular edge computing," *IEEE Wireless Commun. Lett.*, vol. 9, no. 7, pp. 928–932, Jul. 2020.

[64] Y. Shan, H. Wang, Z. Cao, and K. Yury, "Data offloading in heterogeneous dynamic fog computing network: A contextual bandit approach," in *Proc. 3rd Int. Conf. Comput. Commun. Internet (ICCCI)*, Jun. 2021, pp. 73–77.

[65] R. Zhang, P. Cheng, Z. Chen, S. Liu, B. Vucetic, and Y. Li, "Calibrated bandit learning for decentralized task offloading in ultra-dense networks," *IEEE Trans. Commun.*, vol. 70, no. 4, pp. 2547–2560, Apr. 2022.

[66] O. Chapelle and L. Li, "An empirical evaluation of Thompson sampling," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 24, 2011, pp. 1–10.
[67] H. Tran-Dang and D.-S. Kim, "BLM-DTO: Bandit learning and matching based distributed task offloading in fog networks," in *Proc. Int. Conf. Electron., Inf. Commun. (ICEIC)*, Feb. 2023, pp. 1–4.
[68] Z. Wang, L. Guo, J. Yin, and S. Li, "Bandit learning in many-to-one matching markets," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*. New York, NY, USA: Association for Computing Machinery, 2023, pp. 2088–2097.
[69] F. Kong, J. Yin, and S. Li, "Thompson sampling for bandit learning in matching markets," 2022, *arXiv:2204.12048*.

HOA TRAN-DANG (Member, IEEE) received the B.E. degree in electrical and electronics engineering from the Hanoi University of Science and Technology (HUST), Vietnam, in 2010, the M.S. degree in electronics engineering from the Kumoh National Institute of Technology (KIT), South Korea, in and the Ph.D. degree from the University of Lorraine, France, in 2017. He is currently with the NSL Laboratory, Department of ICT Convergence Engineering, KIT, as a Research Professor. His research interests include wireless sensor networks, the Internet of Things (IoT), physical internet, and radio resource management in wireless industrial networks.



KI-HYUP KWON (Member, IEEE) received the Ph.D. degree in electrical engineering from the Kumoh National Institute of Technology (KIT). He has been with the ICT Convergence Research Center, Kumoh National Institute for Technology, Republic of Korea, as a Researcher, since 2014. His research interests include AI-based system failure diagnosis, metaverse, and blockchain.

DONG-SEONG KIM (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2003. From 1994 to 2003, he was a full-time Researcher with ERC-ACI, Seoul National University. From March 2003 to February 2005, he was a Postdoctoral Researcher with the Wireless Network Laboratory, School of Electrical and Computer Engineering, Cornell University, NY, USA. From 2007 to 2009, he was a Visiting Professor with the Department of Computer Science, University of California at Davis, Davis, CA, USA. He is currently the Director of the Kit Convergence Research Institute and the ICT Convergence Research Center (ITRC Program) supported by Korean Government, Kumoh National Institute of Technology. His current research interests include the real-time IoT, industrial wireless control networks, networked embedded systems, and fieldbus. He is a Senior Member of ACM.