frontiers | Frontiers in Signal Processing

ORIGINAL RESEARCH
published: 02 May 2022
doi: 10.3389/frsip.2022.864392

A Tutorial on Bandit Learning and Its Applications in 5G Mobile Edge Computing (*Invited Paper*)

Sige Liu[1], Peng Cheng[1,2], Zhuo Chen[3], Branka Vucetic[1] and Yonghui Li[1]**

[1]School of Electrical and Information Engineering, The University of Sydney, Darlington, NSW, Australia, [2]Department of Computer Science and Information Technology, La Trobe University, Melbourne, VIC, Australia, [3]CSIRO DATA61, Eveleigh, NSW, Australia

Due to the rapid development of 5G and Internet-of-Things (IoT), various emerging applications have been catalyzed, ranging from face recognition, virtual reality to autonomous driving, demanding ubiquitous computation services beyond the capacity of mobile users (MUs). Mobile cloud computing (MCC) enables MUs to offload their tasks to the remote central cloud with substantial computation and storage, at the expense of long propagation latency. To solve the latency issue, mobile edge computing (MEC) pushes its servers to the edge of the network much closer to the MUs. It jointly considers the communication and computation to optimize network performance by satisfying quality-of-service (QoS) and quality-of-experience (QoE) requirements. However, MEC usually faces a complex combinatorial optimization problem with the complexity of exponential scale. Moreover, many important parameters might be unknown *a-priori* due to the dynamic nature of the offloading environment and network topology. In this paper, to deal with the above issues, we introduce bandit learning (BL), which enables each agent (MU/server) to make a sequential selection from a set of arms (servers/MUs) and then receive some numerical rewards. BL brings extra benefits to the joint consideration of offloading decision and resource allocation in MEC, including the matched mechanism, situation awareness through learning, and adaptability. We present a brief tutorial on BL of different variations, covering the mathematical formulations and corresponding solutions. Furthermore, we provide several applications of BL in MEC, including system models, problem formulations, proposed algorithms and simulation results. At last, we introduce several challenges and directions in the future research of BL in 5G MEC.

Keywords: 5G, mobile edge computing, bandit learning, task offloading, resource allocation

OPEN ACCESS

Edited by:
Boya Di,
Imperial College London,
United Kingdom

Reviewed by:
Zehui Xiong,
Nanyang Technological University,
Singapore
Youjia Chen,
Fuzhou University, China

****Correspondence:***
Peng Cheng
p.cheng@latrobe.edu.au
peng.cheng@sydney.edu.au
Yonghui Li
yonghui.li@sydney.edu.au

Specialty section:
This article was submitted to Signal Processing for Communications, a section of the journal Frontiers in Signal Processing

Received: *28 January 2022*
Accepted: *28 March 2022*
Published: *02 May 2022*

Citation:
Liu S, Cheng P, Chen Z, Vucetic B and Li Y (2022) A Tutorial on Bandit Learning and Its Applications in 5G Mobile Edge Computing (Invited Paper). Front. Sig. Proc. 2:864392. doi: 10.3389/frsip.2022.864392

1 INTRODUCTION

The ever-increasing deployment of the fifth-generation (5G) communication and the Internet-of-Things (IoT) has created a large number of emerging applications ranging from face recognition, virtual reality to autonomous driving (Shi et al., 2016; Teng et al., 2019), and generated enormous volumes of data for transmission, storage, and execution. However, these tremendous computational requirements are usually beyond the capacity of mobile users (MUs), making it impossible to complete tasks in a prompt manner. Mobile cloud computing (MCC) (Khan et al., 2014) is a

promising computational paradigm to relieve this situation by enabling MUs to offload their applications to the remote central cloud with strong computation and storage infrastructure. However, the inherent issue of MCC is a long communication distance between MUs and the remote cloud center, resulting in a long propagation and network delay.

Mobile edge computing (MEC) (Wang et al., 2017; Zhang et al., 2020; Asheralieva et al., 2021) has been envisioned as a key enabler to deal with the latency issue in MCC. MEC pushes the computation and storage resources to the network edge much closer to the local devices, benefiting from a low propagation latency and privacy/security enhancement (Mao et al., 2017). In MEC, joint consideration of communication and computation plays a pivotal role in network performance optimization to satisfy quality-of-service (QoS) and quality-of-experience (QoE) requirements (Yang et al., 2020; Lim et al., 2021).

Despite its potential benefits, MEC also suffers from several challenging issues. An MU needs to offload computation tasks to MEC servers in an opportunistic manner subject to the available edge computation and communication resources shared by a large number of MUs. Consequently the offloading decision and communication/computation resource allocation should be jointly optimized to maximize the network performance, typically leading to a complex combinatorial optimization problem with complexity of exponential scale. This is further exacerbated by the dynamic nature of the offloading environment and network topology, where many important parameters (e.g., channel state information and servers' computation workload) in the formulated problem are either impossible or difficult to obtain *a-priori*.

Bandit learning (BL) (Gittins et al., 2011; Sutton and Barto, 2018), a typical online learning approach, offers a promising solution to deal with the aforementioned issues. It enables each agent (MU/server) to make a sequential selection from a set of arms (servers/MUs) in order to receive some numerical rewards available to the agent after pulling the arm. BL aims to strike a tradeoff between exploitation (exploit the learned knowledge and select the empirically optimal arm) and exploration (explore other arms than the optimal one to get more reward information). Consequently, the arms can be iteratively learned, and selection decisions will be improved progressively. The major advantages that BL brings to the joint consideration of offloading decision and network resource allocation can be summarized as follows.

- Matched mechanism: The inherent idea behind MEC is to design policies to make a better selection for MUs or servers. This clearly coincides with the design purpose of BL. This match in mechanism provides selection policies to obtain better performance such as lower latency, lower energy consumption, and higher task completion ratio.
- Situation awareness through learning: In the arm selection process, BL is able to learn the corresponding parameters of the offloading environment.
- Adaptability: The structure of BL can be readily modified to accommodate a variety of characteristics, requirements, and constraints in an MEC system.

In this paper, we present a comprehensive tutorial on BL in the 5G MEC system. We first review the background of the BL, including its origin, concept of regret, objective, and workflow, then we introduce several basic mathematical parameters to formulate a general BL problem. To deal with the BL problem, we present several popular strategies, including ϵ-greedy, upper confidence bound (UCB) algorithm, and weighted policy. Based on the number of agents, BL can be classified as single-agent BL (SA-BL), multi-agent BL (MA-BL), and other types. Specifically, SA-BL can be classified as stateful and stateless for arms with and without states, respectively. The stateful and stateless BL can be further classified into stochastic/non-stochastic and rested/restless, respectively. For each above BL type, we provide several widely used solutions to deal with the corresponding features and issues. Apparently, SA-BL is a special case of MA-BL, which, due to the participation of multiple agents, increases the learning efficiency and enhances the system capacity, yet has two more unique issues: collisions and communications. Collisions happen when one arm is selected by different agents simultaneously, and the topology of the network could be potentially complicated by communication modeling. We present two popular collision models and two communication models, in which we provide the corresponding distributed solutions to alleviate collisions and optimize system performance. Apart from the BL types in SA-BL and MA-BL, we also introduce several important BL variations (i.e., contextual, sleeping, and combinatorial BL) to cover more features, improving the BL model structure from different perspectives. Furthermore, to show how to deploy BL into the 5G MEC system, we introduce three BL applications: contextual sleeping BL, restless BL and contextual calibrated BL. Specifically, we introduce their system models and problem formulations, then provide simulation results to illustrate the excellent performances of the BL algorithms. At last, we envision the possible development avenues of BL In 5G MEC, and identified several research directions together with the associated challenges.

The remainder of this paper is organized as follows. **Section 2** introduces the fundamentals of the BL problem and presents a representative classification. SA-BL, MA-BL, and other types are elaborated on, and their features and solutions are given in **Sections 3–5**, respectively. Several applications of BL in 5G MEC and the simulation results are introduced in **Section 6**. Finally, future challenges and directions are drawn in **Section 7**.

2 BANDIT LEARNING PROBLEM

2.1 Background

The classical BL problem comes from a hypothetical experiment where an agent pulls a gambling machine (arm) from a set of such machines, each successive selection of which yields a reward and a state. The agent attempts to obtain a higher total reward through a set of selections. However, due to the lack of prior information about arms, the agent may pull an inferior arm in terms of reward at each round, yielding regret measuring the expected performance loss of the BL process. In other words, regret

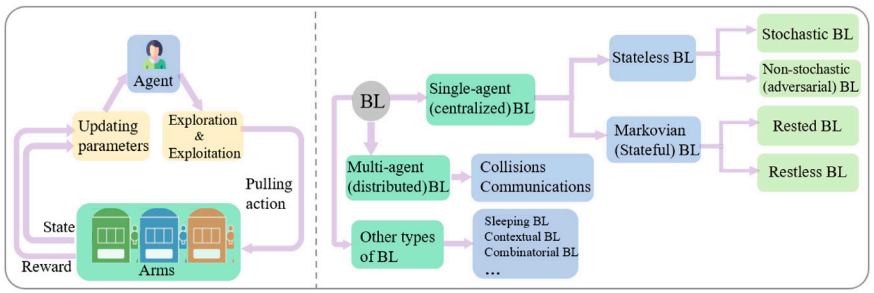

FIGURE 1 | The workflow of BL and a representative Classification of BL.

indicates the reward deviation of the pulled arm from the optimal one. Hence, the agent's objective is to find a policy to improve its arm selection decision by minimizing the expected cumulative regret in the long term.

The workflow of an agent in each learning round is presented in **Figure 1**, where the agent pulls one arm based on the previously collected knowledge of each arm (i.e., historical reward and the number of selections). The idea is to strike a tradeoff between exploitation and exploration, where exploitation is defined as the investigation of the learned knowledge about the arms and selection of the empirically optimal one and the exploration is defined as the exploration of other arms than the optimal one to get more reward information. The selected arm yields a reward and state associated with itself and/or time. The agent then updates the corresponding parameters (i.e., empirical reward and pulled times) for each arm. Thus, the agent iteratively learns the reward performance of each arm and progressively improves the arm selection.

2.2 Mathematical Formulation

We consider a BL system with K arms indexed by $\mathcal{K} = \{1, \ldots, k, \ldots K\}$ and the timeline is divided into T slots indexed by $\mathcal{T} = \{1, \ldots, t, \ldots, T\}$. We denote by $X_{k,t}$ and $\mu_{k,t} = \mathbb{E}[X_{k,t}]$ the reward and the expected reward of pulling arm k at slot t, respectively. We define a_t as the action selection of an arm at slot t. The indicator function $1\{a_t = k\} = 1$ indicates that arm k is pulled at slot t. We then denote by $r_{k,t}$ the reward yielded by pulling arm k at slot t. The objective of the agent is to find a policy $\mathcal{G} = \{a_1, \ldots a_t, \ldots, a_T\}$ to select the optimal arm with the highest expected reward and decrease the regret of the BL process accordingly. Therefore, we define the optimal policy $\mathcal{G}^\star$, which has the prior knowledge of all arms. It makes sure that the agent can always pull the optimal arm a_t' at slot t with the highest reward. Hereafter, each symbol with superscript "$\star$" corresponds to that achieved by the optimal policy. Then the general form of the regret can be written as

$$R(T) = \mathbb{E}_{\mathcal{G}^\star}\left[\sum_{t=1}^{T} r^\star_{a_t',t}\right] - \mathbb{E}_{\mathcal{G}}\left[\sum_{t=1}^{T} r_{a_t,t}\right]. \tag{1}$$

On this basis, the agent decides which arms to pull in a sequence of trials to minimize its BL regret across slots by striking a tradeoff between exploitation and exploration.

2.3 Bandit Learning Approaches

In the sequel, we review several widely used strategies to tackle the BL problem. These strategies will be modified and extended to obtain a few state-of-the-art algorithms to resolve specific problems, to be detailed in **Sections 3–6**.

2.3.1 ϵ-Greedy

As an intuitive strategy, ϵ-greedy (Kuleshov and Precup, 2014) enables the agent to select an arm with the maximum observed value of the reward based on the current knowledge with a probability smaller than ϵ ($0 < \epsilon < 1$). If the probability is larger than ϵ, it randomly selects arms. For ϵ-greedy, its regret grows linearly in time and its performance can be improved by adjusting the value of ϵ.

2.3.2 Upper Confidence Bound Algorithm

UCB (Auer et al., 2002a) utilizes the confidence intervals on the empirical estimate of the reward of arms and calculates a UCB index for each arm. The UCB index consists of the average reward of each arm and a padding function. The padding function adjusts the exploration-exploitation according to the current slot and the pulled times of each arm. The arm with the highest UCB index will be pulled by the agent at each slot, and the index will be updated according to the received reward and selected times.

2.3.3 Weighted Policy

The weighted policy (Bubeck and Cesa-Bianchi, 2012) enables the agent to select an arm at each slot based on a mixed probability distribution. The distribution combines a uniform distribution and another one, which weights the arms according to their average regret performance in the past. This method is usually exploited to deal with the non-stochastic reward problems where the reward generation model of each arm cannot be classified into any specific probability distribution.

2.4 Classification of Bandit Learning Models

We can classify BL into several different models as shown in **Figure 1** based on its settings in terms of the number of agents and the reward generation models. Note that the classification cannot cover all types of BL, and we chose the most popular and representative ones in this paper.

2.4.1 SA-BL

When the system has only one agent or multiple agents but with a centralized controller, it is referred to as single-agent BL (SA-BL) or centralized BL model. SA-BL can be further classified as stateful (Markov) and stateless for arms with and without states, respectively. The stateless BL can be classified into two forms. It is referred to as the stochastic BL, if the reward is stochastically drawn from a probability distribution, and as the non-stochastic (adversarial) BL otherwise. For the stateful BL, it could also be embodied in two forms: the rested BL if only the state of the pulled arm changes at each slot, and the restless BL if the states of all the arms change.

2.4.2 MA-BL

Furthermore, when the system has multiple agents without a centralized controller, it is referred to as multi-agent BL (MA-BL) or distributed BL model. Consequently, the issue of collision arises due to multiple agents simultaneously pulling the same arm, adding to the calculation complexity of the regret and the reward allocation. In addition, the communication model needs to be carefully designed to accommodate the information exchange among multiple agents.

2.4.3 Other BL Models

Apart from the SA/MA-BL models aforementioned, other significant BL variations, such as sleeping, contextual, and combinatorial BL, can cover more features (e.g., arms availability, contextual information, and multiple selections). These features can also be incorporated into SA-BL and MA-BL, resulting in more complex BL models such as contextual distributed BL (Lu et al., 2010; Chu et al., 2011), restless combinatorial BL (Gai et al., 2012).

3 SINGLE-AGENT (CENTRALIZED) BANDIT LEARNING

In this section, we elaborate on the single-agent (centralized) BL following the classification in **Figure 1**. Specifically, the stateless BL and stateful (Markov) BL will be covered.

3.1 Stateless Bandit Learning

In stateless BL, the agent only receives the reward of the pulled arm, and independence holds for reward across slots for each arm. In other words, arms do not have states. We have two typical stateless BL types based on different reward generation models: stochastic BL and non-stochastic (adversarial) BL.

3.1.1 Stochastic Bandit Learning

For stochastic BL, the reward of pulling each arm is stochastically drawn from a specific probability distribution which can be stationary or non-stationary. For the stationary case, the expected reward of each arm is time-independent, i.e., $\mu_{k,t} = \mu_k$. On this basis, the regret function can be written as

$$R_{\text{stationary}}(T) = T \cdot \mu^{\star} - \mathbb{E}_{\mathcal{G}}\left[\sum_{t=1}^{T} r_{a_t,t}\right]. \tag{2}$$

For the non-stationary case, the expected reward of each arm might change with time. In this case, the previously learned knowledge may not truly reflect the real expected reward of current arms, potentially rendering the historical observations of the pulled arms less useful. Consequently, the BL problem becomes complicated, and the probability of pulling a suboptimal arm increases. Then we have its regret function as

$$R_{\text{non-sta}}(T) = \sum_{t=1}^{T} \mu_t' - \mathbb{E}_{\mathcal{G}}\left[\sum_{t=1}^{T} r_{a_t,t}\right]. \tag{3}$$

UCB family algorithms have been widely adopted to resolve the exploration-exploitation dilemma of the stochastic BL problems. For the stationary case, the basic UCB algorithms suffice. To deal with the non-stationary cases, the sliding-window UCB algorithm (Ding et al., 2019) could be adopted, only considering the previous observation of a fixed length. Furthermore, we can utilize the discount UCB algorithm (Garivier and Moulines, 2008), which emphasizes the recent actions by averaging the rewards of arms with a discount factor placing more weight on the recent observations. In addition, some statistical test methods [e.g., generalized likelihood ratio or Page-Hinkly test (Maghsudi and Hossain, 2016)] can be drawn upon to detect the expected reward changes to improve the arm selection process.

3.1.2 Non-Stochastic (Adversarial) Bandit Learning

For non-stochastic (adversarial) BL, the reward generation of each arm does not have any specific probability distribution. In other words, the reward of each arm is determined by an adversary at each slot rather than by a stochastic generation process. A special case of regret, referred to as weak regret, is usually used to measure the loss of adversarial BL. It considers the single globally optimal arm and can be written as

$$W(T) = \max_{i} \sum_{t=1}^{T} r_{i,t} - \mathbb{E}_{\mathcal{G}}\left[\sum_{t=1}^{T} r_{a_t,t}\right]. \tag{4}$$

Weighted policy algorithms are usually used to resolve the adversarial reward generation problem in the non-stochastic BL. For full information cases, where the agent observes the total rewards after each selection, we introduce the *Hedge* algorithm (Slivkins, 2019), whose main idea is to pull an arm with a probability proportional to the average performance of arms in the past. The arms with high rewards quickly gain a high probability of being pulled. For partial information cases, where the agent only observes the reward of the pulled arms, we can utilize the EXP3 algorithm (Kuleshov and Precup, 2014), which performs the Hedge algorithm as a subroutine with a mixed probability distribution. The distribution combines the uniform distribution and a distribution, which is determined by the weight of each arm. In addition, there are many other variations of the EXP3 algorithm, including EXP3.S, EXP3.P, and EXP4 algorithms. Interested readers are referred to (Auer et al., 2002b) for details.

3.2 Stateful (Markov) Bandit Learning

In the stateful (Markov BL), each arm has some finite states and a Markov chain, where the probability of the following state only depends on the current state. At each slot, the pulled arm yields a reward drawn from a probability distribution, and the state of the arm changes to a new one based on the Markov state evolution probability. According to different state evolution models, stateful BL can be classified into two types: rested (frozen) BL and restless BL.

- In the rested BL model, at each round, only the state of the selected arm evolves with time and the states of other arms are frozen.
- In the restless BL model, at each round, all the states of arms (including the unselected arms) might evolve with time.

In order to resolve the stateful BL problems, we usually leverage index policies (Whittle, 1980; Liu and Zhao, 2010a), which, for each arm, calculate a defined index and provide a proxy to measure the expected reward in the current state. To deal with rested BL, the Gittins index policy (Whittle, 1980) is usually adopted. This policy works under the Bayesian framework and transforms an N-dimension rested BL problem into N independent 1-dimension ones, significantly reducing the computational complexity. Furthermore, in the restless BL model, we can leverage Whittle's index policy (Whittle, 1988). The policy first needs to prove that each arm is indexable, which guarantees the existence of Whittle's index in restless BL. Then it decouples the restless BL problem into multiple sub-problems by applying Lagrangian relaxation for computational simplification.

4 MULTI-AGENT (DISTRIBUTED) BANDIT LEARNING

In this section, we extend single-agent (centralized) BL to multi-agent (distributed) BL. The participation of multiple agents in the arm selection process brings the benefits of increased learning efficiency and enhanced system capacity at the expense of increasing network complication and computational complexity. Two critical issues, collisions and communications between agents, naturally arise due to the introduction of multiple agents. Collisions occur when different agents pull the same arm simultaneously, and communication modeling could potentially complicate the topology of the network.

Apparently, SA-BL is a particular case of MA-BL, and the aforementioned different features of the classifications in SA-BL, shown in **Figure 1**, are also applicable to MA-BL. For simplicity, we do not repeat the classifications and focus on the issues of collisions and communications.

4.1 Modelling of Collisions

We consider M agents in MA-BL indexed by $\mathcal{M} = \{1, \ldots, m, \ldots, M\}$, and the policy $\mathcal{G}$ for single agent is extended to $\mathcal{G} = \{\mathbf{a}_1, \ldots, \mathbf{a}_t, \ldots, \mathbf{a}_T\}$, where the vector $\mathbf{a}_t = [a_{1,t}, \ldots, a_{m,t}, \ldots, a_{M,t}]$ indicates the actions of all the M agents at slot t. Other settings are in line with those of the mathematical formulation in **Section 2.2**. Two popular collision models are summarized as follows.

- Collision model I: When multiple agents pull the same arm, they share the reward of the arm in a specific (e.g., uniform and arbitrary) manner. In this model, the reward at slot t can be written as

$$X_t = \sum_{k=1}^{K} 1_k^{\mathrm{I}} \cdot r_{k,t}, \tag{5}$$

where 1_k^{I} equals 1 if arm k is pulled at least once, and 0 otherwise.

- Collision model II: When multiple agents pull the same arm, none of them obtains a reward. In this model, the reward at slot t can be written as

$$X_t = \sum_{k=1}^{K} 1_k^{\mathrm{II}} \cdot r_{k,t}, \tag{6}$$

where 1_k^{II} equals 1 if arm k is pulled exactly once, and 0 otherwise.

Note that if a collision happens and all the collided agents have the full reward of the arm, it is a trivial problem as it carries no difference from SA-BL.

Following the above models, the regret function of MA-BL can be written as

$$R_{\mathrm{MA}}(T) = \mathbb{E}_{\mathcal{G}^*}\left[\sum_{m=1}^{M} r_{a_{m,t}^*,t}^*\right] - \mathbb{E}_{\mathcal{G}}\left[\sum_{t=1}^{T} X_t\right]. \tag{7}$$

To minimize the regret function, a distributed policy needs to be carefully designed to alleviate collisions, although such a policy might not be optimal from the perspective of a single agent.

4.2 Modelling of Communications

The level of information exchange among agents could vary significantly. Here we introduce two communication models and give the corresponding solutions in the following.

4.2.1 No Communication

One extreme case is that no information exchange among agents takes place. In such a case, each agent pulls an arm only based on its local observation of the previous selections and rewards. To alleviate collisions, we usually utilize order-optimal policies (Liu and Zhao, 2010b). These policies design a pre-determined slot allocation pattern so that each agent can use a different slot to select the optimal arm independently. We exemplify such a concept for the case with two agents. At even slots, agent one selects the arm with the highest BL index (e.g., UCB index) and agent two selects the arm with the second-highest index, and vice versa for the odd slots.

4.2.2 Partial Communications

In partial communications, different agents partially communicate with each other. For instance, either an agent can observe other agents' selections only when they select the same arm, or an agent can only communicate with agents being its neighbor or within a given number of hops. With this

information exchange, agents involved can make a collaborative policy to facilitate the collision reduction, usually with the aid of a graph-based method. We define an undirected coordinate graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of nodes and $\mathcal{E}$ is a set of edges between nodes. We model the MA-BL network as a coordinate graph, where each node $v \in \mathcal{V}$ can be viewed as an agent and each edge between a pair of nodes $(v_p, v_g) \in \mathcal{E}, \forall v_p, v_q \in \mathcal{V}$ indicates that the agent v_p and v_q are neighbors and the two agents may make arm selection collaboratively. Based on the graph, coop-UCB and coop-UCL algorithms (Landgren, 2019) are developed to deal with the issue of collisions in the BL with partial communications. As another solution, calibrated-based BL (Foster and Vohra, 1997) enables each agent to simultaneously learn the reward performance of arms and predict the selections of other agents. The equilibrium point will be progressively reached in the BL process, reducing the collision frequency.

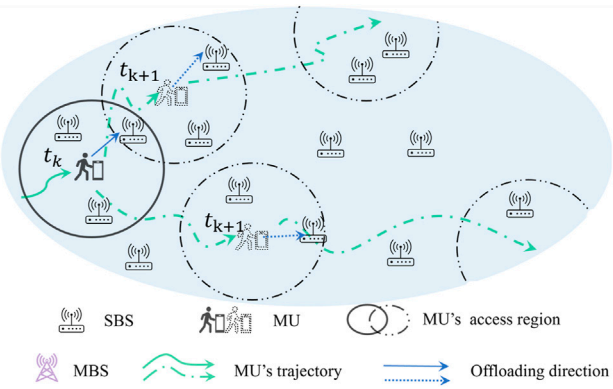

FIGURE 2 | Illustration of user-centric task offloading for MEC in UDN. The two dotted lines represent the uncertain future tracks of the MU.

5 OTHER TYPES OF BANDIT LEARNING

Apart from the classifications in SA-BL and MA-BL, shown in **Figure 1**, there are many other BL models and we will introduce some of them in this section. Here many other BL models exist, which are independent of those in SA-BL and MA-BL, potentially covering more features. They can be incorporated into SA-BL and MA-BL, generating more comprehensive BL models. For simplicity, we only introduce these models with a single agent.

5.1 Contextual BL

Contextual BL assumes that the expected reward of each arm is a function of the contextual information. It enables the agent to make a sequential selection from a set of arms based on the observed contextual information and previous knowledge, followed by receiving numerical rewards. This extra information can accelerate the convergence process by learning the underlying connection between arms and contextual information.

LinUCB algorithm (Chu et al., 2011), developed based on the classical UCB algorithm, is usually leveraged in contextual BL. It takes advantage of the contextual information by maintaining a context-related vector, assuming that the reward of pulling an arm is a function of the vector and the corresponding contextual information. In addition, there are many other solutions, e.g., LinREL and KernelUCB, and interested readers are referred to (Zhou, 2015) for details.

5.2 Sleeping BL

In sleeping BL, the set of available arms is time-varying and each arm's state can be exchanged between "awakening" (can be pulled) and "sleeping" (cannot be pulled).

The optimal arm may be time-varying and the arm selection becomes more complex, decreasing the learning speed. Moreover, apart from awakening and sleeping, the states of arms can also be mortal (Chakrabarti et al., 2008). It means that arms are available only for a finite time period, which can be known/unknown and deterministic/stochastic.

The sleeping BL is often resolved by using FTAL and AUER algorithms (Kleinberg et al., 2010). Since the optimal arm may be sleeping in some slot, these algorithms aim to order in advance all the arms in terms of the expected rewards, and view the optimal ordering of the selections as the optimal policy. As for the mortal case, DetOpt algorithm (Chakrabarti et al., 2008) solves the selection by pulling each arm several times and abandoning the arm unless it seems promising.

5.3 Combinatorial BL

In combinatorial BL, we relax the setting that the agent selects one arm for each slot, and assume that a set of arms (a super arm) can be pulled at each slot.

The reward of the pulled super arm is a sum function of weighted value of all the pulled arms' rewards, due to the dependencies of arms. Here, the selection space exponentially increases due to the explosion of combinations of arms.

To deal with the issues of selection space and dependencies of arms, we can utilize LLR algorithm (Gai et al., 2012), which selects a super arm and records the observations both for the arms and the super arm. As an arm might belong to different super arms, LLR could exploit this dependency information to accelerate the learning process. Other algorithms for solving combinatorial BL, such as CUCB and ComBand, can be found in (Cesa-Bianchi and Lugosi, 2012) for interested readers.

6 APPLICATIONS OF BL IN 5G MOBILE EDGE COMPUTING

In this section, we introduce several applications of BL in the 5G MEC system. We first present their system models and problem formulations, then provide their simulation results to show the excellent performances of the proposed different BL algorithms. We consider SA-BL in the first two applications and MA-BL in the last one.

6.1 Contextual User-Centric Task Offloading for Mobile Edge Computing

In Liu et al. (2022), we consider a general user-centric task offloading scheme for MEC in ultra-dense networks (UDN), where a mobile user (MU) randomly moves around the whole

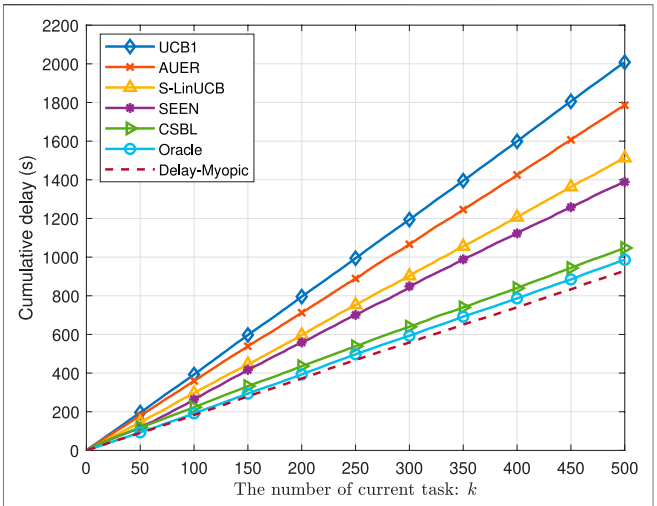

FIGURE 3 | Performance comparison in terms of the cumulative delay.

network without any predictable tracks. The MU may remain static or move in any direction and computational tasks can be generated sequentially at any location. These tasks will be offloaded to its nearby small base stations (SBSs), which can be referred to as arms. As shown in **Figure 2**, the dotted lines represent the uncertain future tracks of the MU, and t_k and t_{k+1} indicate the time when kth and $(k + 1)$-th tasks are generated, respectively.

We formulate the mobile task offloading problem, aiming to minimize the long-term total delay in finishing the tasks of the MU. However, due to the unpredictability of the MU's tracks, many information (e.g., moving tracks, SBS computation capacity, and channel fading gain) cannot be obtained in advance. Besides, the channel conditions (between SBSs and MU) and the delays (of executing different tasks) are changing over time and exhibit randomness. Therefore, the conventional methods cannot tackle the problem.

To address the challenges, we propose the contextual sleeping bandit learning (CSBL) algorithm. The idea is to incorporate the contextual information (e.g., SBS location, service provider, and task type) into the bandit learning to accelerate the convergence process by exploring the underlying relationship between contextual information and arms. At each selection round, the MU first collects the current contextual information of the SBSs and then makes a selection based on the contextual information and previous knowledge of the delay performance of the SBSs. From the perspective of the MU, an available SBS may disappear and then appear again due to the random movement. Hence, we leverage the sleeping bandit learning to enable the MU to identify the status of the arms (sleeping or awakening), thereby accelerating the learning process. We compare the proposed CSBL algorithm with the existing BL algorithms in terms of the cumulative delay, i.e., the cumulated delay over slots, in **Figure 3**. It can be seen that CSBL outperforms other BL algorithms [i.e., UCB1 (Auer et al., 2002a), AUER (Kleinberg et al., 2010), S-LinUCB (Mohamed et al., 2021), and SEEN (Chen et al., 2018)] and close to Oracle and Delay-Myopic, which are two algorithms with prior knowledge of arms.

6.2 Task Offloading for Large-Scale Asynchronous Mobile Edge Computing: An Index Policy Approach

In Xu et al. (2021), we consider a MEC system, shown in **Figure 4**, which has one MEC server and multiple users. Each user runs mission-critical tasks generated randomly, and each task has a deadline. The local computation resource of each user is not sufficient to meet the deadline demand of the tasks. Therefore, a user seeks assistance from resourceful MEC servers by task offloading at the expense of transmission delay and energy.

Our objective is to design a task offloading policy to meet the deadline requirement of each task while keeping the total transmission energy costs at a low level. However, because the task arrival pattern is stochastic and unpredictable, it is impossible to solve a static combinatorial optimization problem before task arrival. Besides, the computational resources at the MEC server are limited. Thus, it is pretty challenging to balance the transmission energy costs and the deadline requirements.

To deal with the challenges, we develop a new reward function, considering both deadline requirements and transmission energy costs. Then we formulate the task offloading problem as a restless BL one with the objective to maximize the total discounted reward over the time horizon. To solve this problem, we propose a Whittle's index (WI) based policy and rigorously prove the indexability, which guarantees the existence of WI in restless BL. Then we focus on the task completion ratio and propose a shorter slack time less remaining workload (STLW) rule, which identifies the criticality of the tasks by giving priority to the tasks with shorter slack time and less remaining workload. By applying STLW into WI, we develop the STLW-WI algorithm, which could improve the performance of the policy by selecting the users with the highest WI value without violating the STLW. Then we compare our proposed algorithm STLW-WI with the existing BL algorithms in terms of task completion ratio, i.e., the proportion of tasks that can be completed before their deadlines. It is clearly shown in **Figure 5** that the WI algorithm outperforms other existing algorithms (i.e., Greedy (Xu et al., 2021), LST (Davis et al., 1993), EDF (Liu and Layland, 1973)), and applying STLW into WI can further improve the task completion ratio compared with WI.

6.3 Multi-Agent Calibrated Bandit Learning With Contextual Information in Mobile Edge Computing

In Zhang et al. (2022), we consider an MA-BL MEC system composed of one macro base station (MaBS), multiple microcell base stations (MiBSs), and several randomly located users shown in **Figure 6**. At each round, the users select MiBSs independently to offload their tasks, and the MaBS monitors the task offloading decisions of all users in this round and broadcasts such information. We incorporate contextual information into our model, such as task size and type. The competition of computational resources among users requires an advanced strategy to reduce the collision.

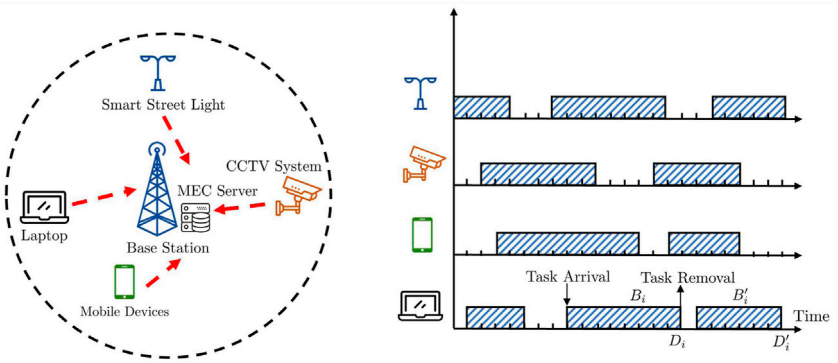

FIGURE 4 | A typical asynchronous MEC system with 4 users.

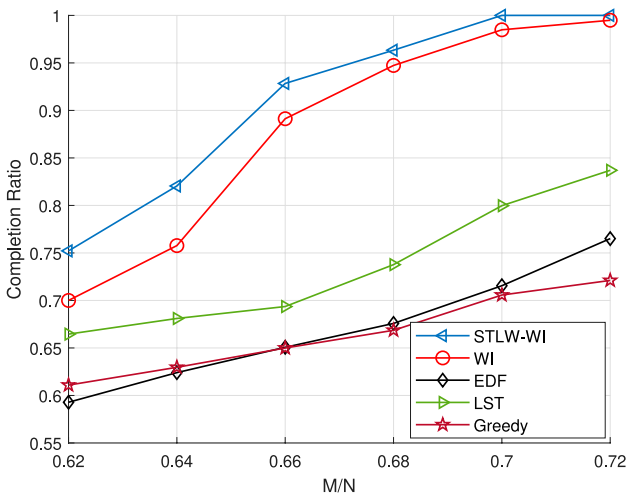

FIGURE 5 | Performance comparison in terms of the task completion ratio.

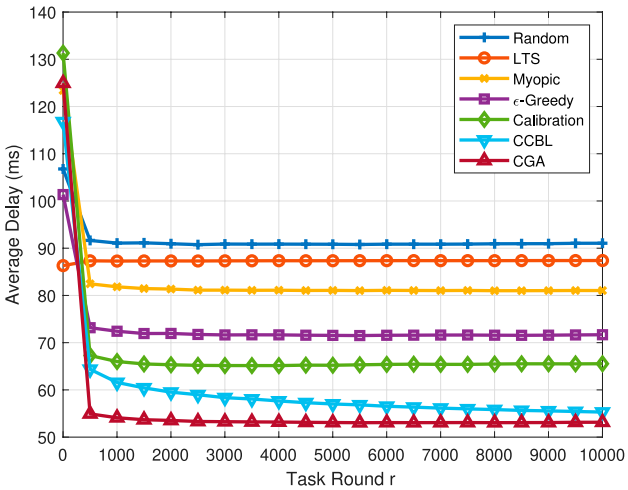

FIGURE 7 | The average delay of all users versus task round *r*.

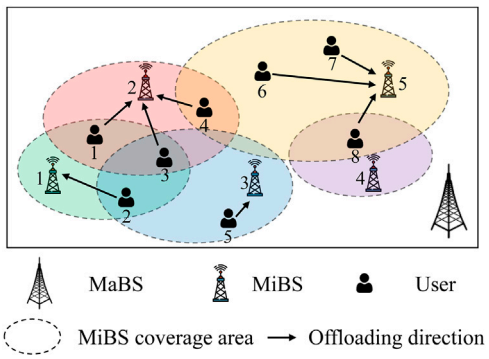

FIGURE 6 | An example of the system model with 5 MiBS and 8 users.

We formulate the task offloading as an optimization problem, aiming to minimize the long-term average task delay of all users with a restricted number of MiBSs. Different from the conventional static optimization, our problem is quite challenging due to the lack of information about MiBSs and other users. Thus, it is impossible to design a task offloading policy by solving a static non-convex nonlinear integer programming problem. We can develop a centralized task offloading policy by utilizing a collaborative BL method. However, centralized decision making is required, collecting all users' task features and then returning the task offloading decisions. A significant transmission delay may be introduced, which is highly undesirable in practice.

To deal with the problem, we develop a decentralized task offloading policy without peer-to-peer information exchange, where all users can make decisions locally. We decouple the formulated problem into several independent contextual BL problems; each user minimizes its own long-term average task delay. We also leverage the calibrated method by developing a calibrated forecaster for each user to predict others' actions. On this basis, we propose a contextual calibrated bandit learning (CCBL) algorithm by fusing the BL and calibrated learning. We compare CCBL with the existing BL algorithms in terms of the

average delay in **Figure 7**. It is clearly shown that the proposed CCBL algorithm outperforms other algorithms [i.e., LTS (Zhang et al., 2022), Myopic (Wang et al., 2020), ϵ-Greedy (Kuleshov and Precup, 2014), and Calibration (Foster and Vohra, 1997)] and is only slightly inferior to CGA (Feng et al., 2019), which is the benchmark with a centralized algorithm.

7 FUTURE RESEARCH CHALLENGES AND DIRECTIONS

Although dozens of BL algorithms and several applications have been introduced and elaborated on, there are still many issues and solutions which are not involved. Here we present a few possible future directions and challenges on BL in 5G and beyond MEC.

7.1 UDN

UDN is a promising paradigm to enable future wireless communications supporting efficient and flexible massive connectivity. This is achieved by deploying a large number of MEC servers, each with a small transmission range at the network edge. Naturally, combining MEC with UDN will significantly increase the coverage of edge computing and provide ubiquitous task offloading services consistently across the whole network (Sun et al., 2017). From the perspective of users, they have more alternative selections of different MEC servers, increasing the selection space. However, the first issue in the UDN-MEC system is the energy constraint because the servers are usually powered by batteries. Another issue is the migration cost caused by a user selecting different servers to perform task offloading. This issue becomes even severe when users are constantly moving across the network. In addition, other issues, such as the Doppler effect, resource competition, and mobility management, should also be considered. These issues must be carefully addressed when we design BL algorithms. For instance, sleeping/mortal characteristics of servers can be leveraged to model the system, and the contextual information of the offloading environment can be utilized to facilitate the arm selection process.

7.2 IIoT

The booming of the IIoT brings an exponential increase in industrial devices, calling for more flexible and low-cost communications. The 5G-MEC technologies bring a cyber revolution and considerable benefits to the industry. This is achieved by supporting flexible communication services with low hardware complexity and providing massive and robust connectivity. Numerous industrial devices generate a massive volume of data, which can be utilized for the deployment of BL methods.

Note that industrial data has unique features. For example, some machine-type communication data is usually stable and has a relatively long transmission horizon. Meanwhile, some mission-critical applications generate ultra-reliable and low-latency data, which has short-length packet and require instance execution (Sisinni et al., 2018). Therefore, different application characteristics and requirements should be considered in the designing and deployment process of BL.

7.3 FGFA

In most existing communication networks, users trying to access wireless channels have to obtain access permission via a contention-based random access (RA) process with multiple handshakes (Shahab et al., 2020). However, the excessive delay and signaling overhead involved are unacceptable for many emerging mission-critical applications and those with small-size tasks. Therefore, to improve the access success ratio and the throughput of the MEC system, FGFA is developed to pre-allocate dedicated channels to specific users so that extra handshakes can be spared (Mahmood et al., 2019). Meanwhile, if the pre-allocated channels are not utilized by users, it will cost unnecessary network resource costs. As a solution, BL could be incorporated to learn the performance of each user and utilize the contextual information to streamline the design of pre-allocation selection policies achieving more efficient network resource utilization.

8 CONCLUSION

In this paper, we provided a comprehensive tutorial on BL in 5G MEC, where BL is incorporated for joint consideration of the offloading decision and communication/computation resource allocation. Specifically, we reviewed the fundamental of BL, including background, mathematical formulation, and several popular solutions. We classified BL into three forms, ranging from SA-BL, MA-BL to other types, then we elaborated on each of them and presented several corresponding widely used algorithms. Furthermore, to show how to deploy BL in 5G MEC, we introduced several applications, where the system models and problem formulations were presented, followed by the simulation results to show the excellent performances of the BL algorithms. In addition, we introduced several future directions and challenges on BL in 5G MEC.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding authors.

AUTHOR CONTRIBUTIONS

In this paper, BV and YL developed the whole framework. SL, PC, and ZC proposed the specific methodologies and conducted experimental results.

REFERENCES

Asheralieva, A., Niyato, D., and Xiong, Z. (2021). Auction-and-learning Based lagrange Coded Computing Model for Privacy-Preserving, Secure, and Resilient mobile Edge Computing. *IEEE Trans. Mobile Comput.*, 1. doi:10.1109/tmc.2021.3097380

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time Analysis of the Multiarmed Bandit Problem. *Mach. Learn.* 47 (2), 235–256. doi:10.1023/a:1013689704352

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (2002). The Nonstochastic Multiarmed Bandit Problem. *SIAM J. Comput.* 32 (1), 48–77. doi:10.1137/s0097539701398375

Bubeck, S., and Cesa-Bianchi, N. (2012). Regret Analysis of Stochastic and Nonstochastic Multi-Armed Bandit Problems. *FNT Machine Learn.* 5 (1), 1–122. doi:10.1561/2200000024

Cesa-Bianchi, N., and Lugosi, G. (2012). Combinatorial Bandits. *J. Comput. Syst. Sci.* 78 (5), 1404–1422. doi:10.1016/j.jcss.2012.01.001

Chakrabarti, D., Kumar, R., Radlinski, F., and Upfal, E. (2008). "Mortal Multi-Armed Bandits," in Proceedings of the 21st International Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada. (Red Hook, NY, USA: Curran Associates Inc.), 273–280.

Chen, L., Xu, J., Ren, S., and Zhou, P. (2018). Spatio-Temporal Edge Service Placement: A Bandit Learning Approach. *IEEE Trans. Wireless Commun.* 17 (12), 8388–8401. doi:10.1109/twc.2018.2876823

Chu, W., Li, L., Reyzin, L., and Schapire, R. (2011). "Contextual Bandits with Linear Payoff Functions," in Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 11–13 Apr, 208–214.

Davis, R. I., Tindell, K. W., and Burns, A. (1993). "Scheduling Slack Time in Fixed Priority Pre-emptive Systems," in 1993 Proceedings Real-Time Systems Symposium, Raleigh-Durham, NC, USA, December, 222–231.

Ding, T., Yuan, X., and Liew, S. C. (2019). Sparsity Learning-Based Multiuser Detection in grant-free Massive-Device Multiple Access. *IEEE Trans. Wireless Commun.* 18 (7), 3569–3582. doi:10.1109/twc.2019.2915955

Feng, W.-J., Yang, C.-H., and Zhou, X.-S. (2019). Multi-user and Multi-Task Offloading Decision Algorithms Based on Imbalanced Edge Cloud. *IEEE Access* 7, 95 970–995 977. doi:10.1109/access.2019.2928377

Foster, D. P., and Vohra, R. V. (1997). Calibrated Learning and Correlated Equilibrium. *Games Econ. Behav.* 21 (1-2), 40. doi:10.1006/game.1997.0595

Gai, Y., Krishnamachari, B., and Jain, R. (2012). Combinatorial Network Optimization with Unknown Variables: Multi-Armed Bandits with Linear Rewards and Individual Observations. *IEEE/ACM Trans. Network.* 20 (5), 1466–1478. doi:10.1109/tnet.2011.2181864

Garivier, A., and Moulines, E. (2008). On Upper-Confidence Bound Policies for Non-stationary Bandit Problems. *arXiv*. arXiv preprint arXiv:0805.3415.

Gittins, J., Glazebrook, K., and Weber, R. (2011). *Multi-armed Bandit Allocation Indices*. Hoboken, NJ, USA: Wiley.

Khan, A. u. R., Othman, M., Madani, S. A., and Khan, S. U. (2014). A Survey of mobile Cloud Computing Application Models. *IEEE Commun. Surv. Tutor.* 16 (1), 393–413. doi:10.1109/surv.2013.062613.00160

Kleinberg, R., Niculescu-Mizil, A., and Sharma, Y. (2010). Regret Bounds for Sleeping Experts and Bandits. *Mach. Learn.* 80 (2), 245–272. doi:10.1007/s10994-010-5178-7

Kuleshov, V., and Precup, D. (2014). Algorithms for Multi-Armed Bandit Problems. *CoRR*. arXiv: 1402.6028 abs/1402.6028.

Landgren, P. C. (2019). *Distributed Multi-Agent Multi-Armed Bandits*. Ph.d. Dissertation. Princeton, NJ, USA: Princeton University.

Lim, W. Y. B., Ng, J. S., Xiong, Z., Niyato, D., Miao, C., and Kim, D. I. (2021). Dynamic Edge Association and Resource Allocation in Self-Organizing Hierarchical Federated Learning Networks. *IEEE J. Select. Areas Commun.* 39 (12), 3640–3653. doi:10.1109/jsac.2021.3118401

Liu, C. L., and Layland, J. W. (1973). Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. *J. ACM* 20 (1), 46–61. doi:10.1145/321738.321743

Liu, K., and Zhao, Q. (2010). Indexability of Restless Bandit Problems and Optimality of Whittle index for Dynamic Multichannel Access. *IEEE Trans. Inform. Theor.* 56 (11), 5547–5567. doi:10.1109/tit.2010.2068950

Liu, K., and Zhao, Q. (2010). Distributed Learning in Multi-Armed Bandit with Multiple Players. *IEEE Trans. Signal. Process.* 58 (11), 5667–5681. doi:10.1109/tsp.2010.2062509

Liu, S., Cheng, P., Chen, Z., Xiang, W., Vucetic, B., and Li, Y. (2022). Contextual User-Centric Task Offloading for mobile Edge Computing in Ultra-dense Networks. *IEEE Trans. Mobile Comput.*

Lu, T., Pál, D., and Pál, M. (2010). "Contextual Multi-Armed Bandits," in Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Chia Laguna Resort, Sardinia, Italy, 13–15 May, 485–492.

Maghsudi, S., and Hossain, E. (2016). Multi-armed Bandits with Application to 5g Small Cells. *IEEE Wireless Commun.* 23 (3), 64–73. doi:10.1109/mwc.2016.7498076

Mahmood, N. H., Abreu, R., Böhnke, R., Schubert, M., Berardinelli, G., and Jacobsen, T. H. (2019). "Uplink grant-free Access Solutions for Urllc Services in 5g New Radio," in 2019 16th International Symposium on Wireless Communication Systems (ISWCS), Oulu, Finland, August, 607–612. doi:10.1109/iswcs.2019.8877253

Mao, Y., You, C., Zhang, J., Huang, K., and Letaief, K. B. (2017). A Survey on mobile Edge Computing: The Communication Perspective. *IEEE Commun. Surv. Tutor.* 19 (4), 2322–2358. doi:10.1109/comst.2017.2745201

Mohamed, E. M., Hashima, S., Hatano, K., Aldossari, S. A., Zareei, M., and Rihan, M. (2021). Two-hop Relay Probing in Wigig Device-To-Device Networks Using Sleeping Contextual Bandits. *IEEE Wireless Commun. Lett.* 10 (7), 1581–1585. doi:10.1109/lwc.2021.3074972

Shahab, M. B., Abbas, R., Shirvanimoghaddam, M., and Johnson, S. J. (2020). Grant-free Non-orthogonal Multiple Access for Iot: A Survey. *IEEE Commun. Surv. Tutor.* 22 (3), 1805–1838. doi:10.1109/comst.2020.2996032

Shi, W., Cao, J., Zhang, Q., Li, Y., and Xu, L. (2016). Edge Computing: Vision and Challenges. *IEEE Internet Things J.* 3 (5), 637–646. doi:10.1109/jiot.2016.2579198

Sisinni, E., Saifullah, A., Han, S., Jennehag, U., and Gidlund, M. (2018). Industrial Internet of Things: Challenges, Opportunities, and Directions. *IEEE Trans. Ind. Inf.* 14 (11), 4724–4734. doi:10.1109/tii.2018.2852491

Slivkins, A. (2019). Introduction to Multi-Armed Bandits. *arXiv*. arXiv preprint arXiv:1904.07272. doi:10.1561/9781680836219

Sun, Y., Zhou, S., and Xu, J. (2017). Emm: Energy-Aware Mobility Management for mobile Edge Computing in Ultra Dense Networks. *IEEE J. Select. Areas Commun.* 35 (11), 2637–2646. doi:10.1109/jsac.2017.2760160

Sutton, R. S., and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT press.

Teng, Y., Liu, M., Yu, F. R., Leung, V. C. M., Song, M., and Zhang, Y. (2019). Resource Allocation for Ultra-dense Networks: A Survey, Some Research Issues and Challenges. *IEEE Commun. Surv. Tutor.* 21 (3), 2134–2168. doi:10.1109/comst.2018.2867268

Wang, S., Zhang, X., Zhang, Y., Wang, L., Yang, J., and Wang, W. (2017). A Survey on mobile Edge Networks: Convergence of Computing, Caching and Communications. *IEEE Access* 5, 6757–6779. doi:10.1109/access.2017.2685434

Wang, F., Xu, J., and Cui, S. (2020). Optimal Energy Allocation and Task Offloading Policy for Wireless Powered mobile Edge Computing Systems. *IEEE Trans. Wireless Commun.* 19 (4), 2443–2459. doi:10.1109/twc.2020.2964765

Whittle, P. (1980). Multi-armed Bandits and the Gittins index. *J. R. Stat. Soc. Ser. B Methodol.* 42 (2), 143–149. doi:10.1111/j.2517-6161.1980.tb01111.x

Whittle, P. (1988). Restless Bandits: Activity Allocation in a Changing World. *J. Appl. Probab.* 25, 287–298. doi:10.1017/s0021900200040420

Xu, Y., Cheng, P., Chen, Z., Ding, M., Li, Y., and Vucetic, B. (2021). Task Offloading for Large-Scale Asynchronous mobile Edge Computing: An index Policy Approach. *IEEE Trans. Signal. Process.* 69, 401–416. doi:10.1109/tsp.2020.3046311

Yang, H., Alphones, A., Xiong, Z., Niyato, D., Zhao, J., and Wu, K. (2020). Artificial-intelligence-enabled Intelligent 6g Networks. *IEEE Netw.* 34 (6), 272–280. doi:10.1109/mnet.011.2000195

Zhang, R., Cheng, P., Chen, Z., Liu, S., Li, Y., and Vucetic, B. (2020). Online Learning Enabled Task Offloading for Vehicular Edge Computing. *IEEE Wireless Commun. Lett.* 9 (7), 928–932. doi:10.1109/lwc.2020.2973985

Zhang, R., Cheng, P., Chen, Z., Liu, S., Vucetic, B., and Li, Y. (2022). Calibrated Bandit Learning for Decentralized Task Offloading in Ultra-dense Networks. *IEEE Trans. Commun.* doi:10.1109/tcomm.2022.3152262

Zhou, L. (2015). A Survey on Contextual Multi-Armed Bandits. *arXiv*. arXiv preprint arXiv:1508.03326.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.