Privacy-Preserving Collaborative Learning for Multiarmed Bandits in IoT

Shuzhen Chen, Youming Tao, Dongxiao Yu, *Member, IEEE*, Feng Li, *Member, IEEE*, Bei Gong, and Xiuzhen Cheng, *Fellow, IEEE*

***Abstract*—This article studies privacy-preserving collaborative learning in decentralized Internet-of-Things (IoT) networks, where the agents exchange information constantly to improve the learnability, and meanwhile make the privacy of agents protected during communications. However, the harsh constraints in IoT make executing collaborative learning much more difficult than well-connected systems composed by servers with strong computation power, due to the weak capacity of devices, limited bandwidth for exchanging information, the asynchronous communication environment, and the necessity of privacy preserving. We show that even if with the harsh constraints in IoT, it still can devise efficient privacy-preserving collaborative learning algorithms, by proposing the first known decentralized collaborative learning algorithm for the fundamental multiarmed bandits problem under the framework of local differential privacy. Rigorous analysis shows that the proposed learning algorithm can make every agent learn the best arm with a high probability and keep the privacy preserved meanwhile. Extensive experiments illustrate that our learning algorithm performs well in real settings.**

***Index Terms*—Collaborative learning, local differential privacy (LDP), multiarmed bandits (MAB), privacy preserving.**

I. Introduction

With the rapid advancement of Internet of Things (IoT) and edge computing, collaborative learning algorithms for large-scale networked and decentralized systems come into focus in recent years [4], [7], [10], [32], [45]. During the learning process, a set of IoT devices collect data and exchange information with each other through some communication infrastructure, to improve the learning efficiency. However, it is much more difficult to implement collaborative learning in IoT comparing to well-connected systems with powerful devices, such as data centers. First, the IoT network is usually composed by a variety of devices with limited computation capacity and bounded memory, which makes it hard to implement complicated learning tasks, especially those with high memory-demanding requirements. Second, the network is more frequently implemented in an asynchronous mode, as synchronization is extremely costly in both time and power consumption. Third, the information shared by the agents is likely to reflect some sensitive information. Therefore, the unrestricted exchange of sensitive data would inevitably leak individuals' privacy. To tackle this problem, we have to consider privacy-preserving learning algorithms for IoT networks.

In this work, we target at devising privacy-preserving collaborative learning algorithms that can be efficiently and generally implemented in IoT the networks. Specifically, we focus on the classical multiarmed bandits (MAB) problem, which is a fundamental problem in machine learning. MAB has been widely used to model some fundamental problems in IoT, such as spatiotemporal edge service placement [15], task replication [36], adaptive task offloading [35] for vehicular edge computing, cognitive radio policies [13], real-time health monitoring [48], and resource allocation in IoT networks [37]. In this problem, the agents need to make a sequence of decisions among a fixed finite collection of arms (options), whose rewards can be regarded as stochastic, to minimize the cumulative regret [3], [8], [23], [24] or identify the best arm [2], [18], [19], [44]. We here pay our attention to the objective of identifying the best arm. It has been shown that with the harsh constraint in the IoT network, it is impossible to identify the best arm for an isolated agent [16]. Fortunately, with the aid of interaction, multiple agents are able to learn the best arm collaboratively [34]. However, it is still open whether the best arm can be identified collaboratively with the privacy preserved when sharing the information. We answer this open problem affirmatively by proposing a collaborative learning algorithm that can converge while preserving the privacy of agents when they communicate with each other.

Specifically, the collaborative learning algorithm is presented under the framework of local differential privacy (LDP). LDP provides provable privacy preservation for sharing data without the assumption of the trusted data collector, and hence is suitable to be implemented in decentralized IoT networks. In the LDP setting, agents randomly perturb their raw data independently according to a mechanism that satisfies the definition of LDP, then shares the perturbed data to others, such that the privacy can be preserved. However,

Manuscript received May 16, 2020; revised July 8, 2020; accepted August 4, 2020. Date of publication August 12, 2020; date of current version February 19, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB2102600; in part by NSFC under Grant 61971269, Grant 61832012, and Grant 61702304; in part by the Shandong Provincial Natural Science Foundation under Grant ZR2017QF005; and in part by the Industrial Internet Innovation and Development Project in 2019 of China. *(Corresponding authors: Dongxiao Yu; Feng Li.)*

Shuzhen Chen, Youming Tao, Dongxiao Yu, Feng Li, and Xiuzhen Cheng are with the School of Computer Science and Technology, Shandong University, Qingdao 266237, China (e-mail: szchen@mail.sdu.edu.cn; youming.tao@mail.sdu.edu.cn; dxyu@sdu.edu.cn; fli@sdu.edu.cn; xzcheng@sdu.edu.cn).

Bei Gong is with the Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China (e-mail: gongbei@bjut.edu.cn).

Digital Object Identifier 10.1109/JIOT.2020.3015986

it is very challenging to keep a learning process converging with the perturbed data.

To well reflect the reality of IoT networks, we adopt a system model assuming that an agent has bounded memory, due to the limited capacity of nodes in IoT, which makes the nodes may not store the learning history. Hence, some classical learning methods for MAB, such as UCB [40], which relies on reward history of past pulls, do not work anymore in the harsh IoT environment. Time is assumed to be continuous and each agent has an independent Poisson clock with a common parameter. An agent performs a preference update only when an agent's local clock ticks such that our learning can proceed in an asynchronous environment. During the communication, agents are subject to a proposed LDP mechanism which promises that the true options of agents can be hidden.

Under the framework of LDP, we propose the first known decentralized collaborative learning algorithm that can protect the privacy of agents when they communicate with each other. With rigorous analysis, we show that every agent can choose the best arm eventually with a high probability, even if the communicated information is perturbed by the privacy-preserving mechanism. We also conducted extensive experiments to evaluate the performances of the collaborative learning algorithm in real settings. The experimental results show that the learning process converge quickly with reasonable privacy-preserving requirements.

Organization: The remainder of this article is organized as follows. Section II introduces the related works. Section III presents our system model. Section IV shows the learning algorithm. We give the detailed analysis for the convergence and correctness in Section V. Section VI shows the practical performance evaluation. Finally, we conclude this article in Section VII.

II. Related Work

The MAB problem is one of the most fundamental models that capture the exploration–exploitation tradeoffs in many application domains. In the centralized setting, the MAB problem has been extensively studied, with two commonly used objectives: regret minimization [3], [8], [25], [43] and best arm identification [2], [18], [19], [27], [42]. In the regret minimization problem, the goal is to minimize the difference between the cumulative reward obtained by the optimal strategy and that obtained by the agent, while in the best arm identification problem, the decision-maker first performs a pure-exploration phase by sampling from the arms, and then identifies an optimal (or nearly optimal) arm (first formulated in [6]). There are also some other exploration goals studied, such as identifying the top-k best arms [9], [11], [12], [14], [46], [47] or the set of arms whose means are above a given threshold [29]. These algorithms usually have specific requirements on capacities of agents, such as computation power and memory. When the capacity of agents is constrained, it may not be able to get efficient learning algorithms. For example, Cover and Hellman [16] proved that it is impossible to learn the best arm for an isolated agent with time-invariant finite memory.

Recently, there have been growing interests in collaborative learning models for MAB, which was first proposed by [22]. In [7], it aimed to handle the decentralized decision-making of spectrum access, to improve resource exploitation. Agarwal *et al.* [1] studied a limited collaborative learning problem where agents are not completely adaptive and must determine their strategies at the beginning of each round. Tao *et al.* [38] proposed a collaborative learning algorithm with limited interaction based on the round elimination technique used in [1] and showed tight bounds for distributed exploration. However, the learning process in [38] consists of centralized and decentralized algorithms and proceeds synchronously. Su *et al.* [34] gave a collaborative learning algorithm with bounded memory, and showed the learnability of the algorithm. All the above results do not take privacy preserving into consideration.

The privacy preserving has been studied in MAB under the centralized setting. Thakurta and Smith [39] presented the first differentially private algorithm for online learning algorithms. Mishra and Thakurta [31] gave a differentially private algorithm for the stochastic bandit problem. Shariff and Sheffet [33] proposed an algorithm adopting joint differential privacy to solve the contextual linear bandit problem, a version of standard stochastic MAB problem. Basu *et al.* [5] compared lower bounds on the regret of different bandit algorithms satisfying a series of privacy definitions for the MAB problem. In [30], a system that updates local agents by collecting feedback from other agents in a differentially private manner was proposed. Ciucanu *et al.* [15] mitigated the privacy concern within the best-arm identification problem. The above privacy-preserving mechanisms are considered in a centralized scenario where a trusted curator exists. From the perspective of privacy-aware data owners, the existence of curators poses a threat to their privacy. To tackle this problem, we design the collaborative learning algorithm under the requirement of LDP [17], [26], [41]. LDP has been widely studied. Fan and Jin [20] proposed a practical framework providing differential privacy guarantees in social networks. Li and Xu [28] proposed PrivPy, an efficient framework, for privacy-preserving collaborative data mining. To the best of our knowledge, this work is the first one studying distributed collaborative learning under the consideration of LDP.

III. Problem Definitions and Model

We consider the problem of collaboratively learning the best option in the K-armed bandit problem. Specifically, there are K arms/options, denoted as $\mathcal{K} = \{1, 2, \ldots, K\}$, and each arm $k \in \mathcal{K}$ yields a stochastic binary valued reward associated with a parameter $\varphi_k \in [0, 1]$ once it is pulled. When arm k is pulled at time t, the reward θ_k^t is drawn independently from a Bernoulli distribution parameterized by φ_k, i.e., $\theta_k^t \sim Bernoulli(\varphi_k)$. More formally, we have

$$\theta_k^t = \begin{cases} 1, & \text{with probability } \varphi_k \\ 0, & \text{with probability } 1 - \varphi_k. \end{cases} \tag{1}$$

The parameters $\varphi_1, \varphi_2, \ldots, \varphi_K$ can be seen as the qualities of the options, which are unknown in prior to the learning.

TABLE I
FREQUENT NOTATIONS AND DESCRIPTIONS

Notations	Descriptions
N	The number of agents
$\mathcal{N}$	Set of agents $\{1, 2, \cdots N\}$
K	The number of arms/options
$\mathcal{K}$	Set of arms $\{1, 2, \cdots K\}$
θ_k^t	Random reward of arm k at time t
φ_k	Quality(expected reward) of arm k
Δ	The arm gap between the best two arms (i.e., $\varphi_1 - \varphi_2$)
α	Threshold in the ***Update*** step of our algorithm
λ	Common rate of Poisson clock
μ	The lower bound of Δ
C_i	The option preference of agent i
$\boldsymbol{F^t}$	The popularity over $\mathcal{K}$ at time t
$\boldsymbol{U_i}$	The raw preference indicator for agent i
$\boldsymbol{V_i}$	The perturbed preference indicator for agent i
$\boldsymbol{H_i^t}$	The average of $\boldsymbol{V_i}$ received by i at time t
$\widetilde{\boldsymbol{F}}_{\boldsymbol{i}}^{\boldsymbol{t}}$	Estimate of $\boldsymbol{F^t}$ by agent i (unnormalized)
$\widehat{\boldsymbol{F}}_{\boldsymbol{i}}^{\boldsymbol{t}}$	Estimate of $\boldsymbol{F^t}$ by agent i (normalized)

Without loss of generality, it is assume that the best arm is unique and $\varphi_1 > \varphi_2 \geq \cdots \geq \varphi_K \geq 0$. Let $\Delta = \varphi_1 - \varphi_2$.

There are N agents collaboratively learning the best arm/option. In particular, the set of agents is denoted by $\mathcal{N} = \{1, 2, \ldots, N\}$. Any pair of agents $(i, j)_{i,j\in\mathcal{N}}$ can exchange messages with each other via a bidirectional communication channel. However, we consider the harsh case that the bandwidth of the communication channel is limited, and the message transmitted can contain $\mathcal{O}(K)$ bits at most. Initially, each agent possesses a private preference for arms, but it knows nothing about the arms except a lower bound μ on the quality gap Δ between the best two arms.

We assume that the memory of the agents are limited. They only constantly maintain their preferences on the arms, but do not store any information on the past learning process. The time among agents are asynchronous. Each agent is equipped with an independent Poisson clock with a common parameter λ. Only when the Poisson clock ticks, the agent takes action to update its preference.

An agent is said to learn the best option if, as $t \to \infty$, the agent prefers the arm with the highest quality, i.e., pulls only the arm with quality φ_1.

We study the collaborative learning under the requirement of LDP in each information exchange, as defined below.

Definition 1 (LDP): Let $\mathcal{A}$ be a randomized algorithm which takes input from a user's private data set $\mathcal{D}$ and $\mathsf{im}\mathcal{A}$ be the image of algorithm $\mathcal{A}$. For a given $\varepsilon \in \mathbb{R}^+$, randomized algorithm $\mathcal{A}$ is said to deliver ε-LDP if and only if for any $x, x' \in \mathcal{D}$ and any subset $\mathcal{S}$ of $\mathsf{im}\mathcal{A}$, we have

$$\frac{\mathbb{P}[\mathcal{A}(x) \in \mathcal{S}]}{\mathbb{P}[\mathcal{A}(x') \in \mathcal{S}]} \leq e^{\varepsilon}. \tag{2}$$

The notations are summarized in Table I.

IV. COLLABORATIVE LEARNING

We present our privacy-preserving collaborative learning algorithm (PPCL) in this section. Some notations are first introduced.

Algorithm 1: PPCL: Privacy-Preserving Distributed Best Option Learning

Input : K, N, μ, ε
Local variables: C_i, $\boldsymbol{U_i}$
Initilization: $C_i = ((i-1) \bmod K) + 1$
if *local Poison clock ticks at time t* **then**
  request all $i' \in \mathcal{N}$ for their preferences $\boldsymbol{V_{i'}^t}$s;
  $\boldsymbol{H_k} = \sum_{i'\in\mathcal{N}} \boldsymbol{V_{i'}^t}/N$;
  foreach $k \in \mathcal{K}$ **do**
    $\widetilde{F}_{i,k}^t = \min\left\{\max\left\{\frac{\exp(\varepsilon/2)+1}{\exp(\varepsilon/2)-1} H_{i,k}^t - \frac{1}{\exp(\varepsilon/2)-1}, 0\right\}, 1\right\}$;
  foreach $k \in \mathcal{K}$ **do**
    $\widehat{F}_{i,k}^t = \widetilde{F}_{i,k}^t / \sum_{\kappa\in\mathcal{K}} \widetilde{F}_{i,\kappa}^t$;
  if $\widehat{F}_{i,C_i^t}^t < \alpha$ **then**
    Choose an arm k randomly with probability proportional to $\widehat{\boldsymbol{F}}^t$;
    if $\theta_k^t = 1$ **then** $C_i = k$;
if *receive the request for preference from i′ at time t* **then**
  foreach $k \in \mathcal{K}$ **do**
    if $C_i^t = k$ **then** $U_{i,k}^t = 1$ **else** $U_{i,k}^t = 0$;
    With probability $\frac{\exp(\varepsilon/2)}{\exp(\varepsilon/2)+1}$, set $V_{i,k}^t = U_{i,k}^t$;
    With probability $\frac{1}{\exp(\varepsilon/2)+1}$, set $V_{i,k}^t = 1 - U_{i,k}^t$;
  Send $\boldsymbol{V_i^t}$ to i';

For each agent, it maintains a local variable $C_i \in \mathcal{K}$ to store the arm it prefers currently. In order to distinguish the values of C_i at different times, we use C_i^t to represent the value of C_i at time t. For $\forall k \in \mathcal{K}$, let $\boldsymbol{F^t} = (F_1^t, F_2^t, \ldots, F_K^t)$ denote the arms' popularity at time t, namely the fraction of the agents preferring each option at time t

$$F_k^t := \frac{\sum_{i=1}^N \mathbf{1}_{\{C_i^t = k\}}}{N}. \tag{3}$$

In the learning process, each agent constantly transmits a K-dimensional vector indicating its preference toward each option, which we call preference-indicating vector. Intuitively, we define the raw indicating vector $\boldsymbol{U_i^t}$ for agent i as follows, for $\forall k \in \mathcal{K}$

$$U_{i,k}^t = \begin{cases} 1, & \text{if } k = C_i^t \\ 0, & \text{if } k \neq C_i^t. \end{cases} \tag{4}$$

The raw vector $\boldsymbol{U_i^t}$ reflects the agent i's true preference.

The algorithm is shown in Algorithm 1. Let $C_i = ((i-1) \bmod K) + 1$ for $\forall$ agent $i \in \mathcal{N}$ in the beginning. When an agent i's local Poison clock ticks, i refines its preference C_i^t according to Algorithm 1. The refining process contains three steps as follows.

1) *Collection:* In this step, agent i aims to collect other agents' preferences. Agent i first requests all agents for their preferences. For $\forall j \in \mathcal{N}$, if a request from i is received, agent j generates its perturbed preference-indicating vector $\boldsymbol{V_j^t}$ based on its raw vector $\boldsymbol{U_j^t}$ and

delivers the vector $\boldsymbol{V_j^t}$ to agent i, where

$$V_{j,k}^t = \begin{cases} U_{j,k}^t, & \text{with probability} \frac{\exp(\varepsilon/2)}{\exp(\varepsilon/2)+1} \\ 1 - U_{j,k}^t, & \text{with probability} \frac{1}{\exp(\varepsilon/2)+1} \end{cases} \tag{5}$$

where ε is the parameter in the LDP requirement. After receiving all the perturbed vectors, i calculates the perturbed popularity, denoted as $\boldsymbol{H_i^t} \in \mathbb{R}^K$, for each arm by averaging these vectors as follows:

$$\boldsymbol{H_i^t} := \frac{\sum_{i' \in \mathcal{N}} \boldsymbol{V_{i'}^t}}{N}. \tag{6}$$

2) *Estimation:* In this step, agent i estimates the true popularity for options at time t. Let $\widetilde{\boldsymbol{F}}_{\boldsymbol{i}}^{\boldsymbol{t}}$ be $i's$ unnormalized estimate for $\boldsymbol{F^t}$ and $\widehat{\boldsymbol{F}}_{\boldsymbol{i}}^{\boldsymbol{t}}$ be the normalized estimate. Specifically, for $\forall k \in \mathcal{K}$

$$\widetilde{F}_{i,k}^t = \min\left\{\max\left\{\frac{\exp(\varepsilon/2)+1}{\exp(\varepsilon/2)-1} H_{i,k}^t - \frac{1}{\exp(\varepsilon/2)-1}, 0\right\}, 1\right\} \tag{7}$$

which is then normalized by

$$\widehat{F}_{i,k}^t = \frac{\widetilde{F}_{i,k}^t}{\sum_{j \in \mathcal{K}} \widetilde{F}_{i,j}^t} \tag{8}$$

such that

$$\sum_{k \in \mathcal{K}} \widehat{F}_{i,k}^t = 1. \tag{9}$$

Note that for any agent whose local clock ticks at time t they can get the same estimate. We denote the common estimate at time t of these agents by $\widehat{\boldsymbol{F}}^{\boldsymbol{t}}$ and use $\widehat{\boldsymbol{F}}^{\boldsymbol{t}}$ instead of $\widehat{\boldsymbol{F}}_{\boldsymbol{i}}^{\boldsymbol{t}}$ for ease of exposition.

3) *Update:* In this step, agent i updates its preference. Let $\alpha = 1-(1/2K)$. Agent i compares $\widehat{F}_{C_i}^t$ with the threshold α first. If $\widehat{F}_{C_i}^t \geq \alpha$, i keeps its preference C_i unaltered. Otherwise i chooses an arm k randomly with probability proportional to $\widehat{\boldsymbol{F}}^{\boldsymbol{t}}$ and then pull this arm. If the reward θ_k^t is 1, update C_i to k, otherwise C_i remains unchanged. In our algorithm, each agent i needs to store a local integer C_i and a local vector $\boldsymbol{U_i}$. Since $C_i \in \{1, 2, \ldots, K\}$ and $\boldsymbol{U_i} \in \{0,1\}^K$, $\log K$ bits is used to store C_i and K bits is used for $\boldsymbol{U_i}$. Hence, each agent only needs a local memory of $O(K)$ bits. Furthermore, at each time t, $O(K)$ bits are enough for an agent i to send the perturbed vector to its neighbor when receiving a request.

V. Theoretical Analysis

In this section, we analyze the theoretical performance of the proposed collaborative learning algorithm PPCL. In particular, we first show the convergence of the learning process, and then that the presented algorithm can preserve the privacy as required by ε-LDP.

A. Learnability

Given $\mathcal{N}$ and $\mathcal{K}$, the learning process can be viewed as a continuous-time random process $(\boldsymbol{X}^N(t) : t \in \mathbb{R}^+)$ where

$$\boldsymbol{X}^N(t) = \left[X_1^N(t), X_2^N(t), \ldots, X_K^N(t)\right]. \tag{10}$$

Here $\forall k \in \mathcal{K}, X_k^N(t) \in \mathbb{Z}$ is the number of agents whose preference is k at time t. In fact, a system state can be seen as a partition of integer N into K non-negative parts, and the state space of $(\boldsymbol{X}^N(t) : t \in \mathbb{R}^+)$ contains all such partitions.

Let $\boldsymbol{s} = (s_1, s_2, \ldots, s_K)$ be a valid state vector, i.e., a proper partition of integer N. Denote by $\mathcal{S}$ the state space consisting of all valid states. Obviously $\mathcal{S}$ is a finite state set.

Let $G^N = (g_{ij} : i, j \in \mathcal{S})$ be the *generator matrix* $G^N = (g_{ij} : i, j \in \mathcal{S})$ of the learning dynamics. Initially, the fraction of selecting every arm is $(1/K)$ by the initial setting of the learning process. This means that each of their estimate popularity is less than the threshold α. Then, the *generator matrix* G^N can be derived as follows:

$$g_{s,s+\ell}^N = \begin{cases} s_\kappa \lambda \widehat{F}_k^t \varphi_k, & \text{if } \ell = e^k - e^\kappa \\ & \text{for } \kappa \neq k \ \& \ \kappa, k \in \mathcal{K} \\ -\sum_{k=1}^K \sum_{\kappa \neq k} s_\kappa \lambda \widehat{F}_k^t \varphi_k, & \text{if } \ell = \boldsymbol{0} \in \mathbb{R}^K \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

As the learning proceeds, G^N will retain to this form unless some unique arm whose estimate popularity attains (be greater than or equal to) α. Denote this arm as k^*. Then the *generator matrix* G^N becomes

$$g_{s,s+\ell}^N = \begin{cases} s_\kappa \lambda \widehat{F}_k^t \varphi_k, & \text{if } \ell = e^k - e^\kappa \\ & \text{for } k \neq \kappa \\ & \kappa \in \mathcal{K} \backslash \{k^*\}, k \in \mathcal{K} \\ -\sum_{k=1}^K \sum_{\substack{\kappa \in \mathcal{K} \backslash \{k^*\} \\ \kappa \neq k}} s_\kappa \lambda \widehat{F}_k^t \varphi_k, & \text{if } \ell = \boldsymbol{0} \in \mathbb{R}^K \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

From (11) and (12), we can get that the random process $(\boldsymbol{X}^N(t) : t \in \mathbb{R}^+)$ defined in (10) has exactly K absorbing states.[1] The state $x^* = [N, 0, \ldots, 0]$ is one absorbing state. And our learning goal is to hit the state x^* with a high probability. Let $\boldsymbol{e^k} \in \mathbb{R}^K$ be the unit vector with the kth component being 1 and other components being 0. Then, each $N \cdot \boldsymbol{e^k}$ (for $k \in \mathcal{K}$) is an absorbing state, since the rate to leave $N \cdot \boldsymbol{e^k}$ is zero. To show the convergence of the learning process, we only need to bound the probability that the random process $(\boldsymbol{X}^N(t) : t \in \mathbb{R}^+)$ goes into the absorbing state $x^* = [N, 0, \ldots, 0]$.

Denote by E^N the success event that every agent learns the best arm as $t \to \infty$, i.e., $\lim_{t \to \infty} X^N(t) = x^*$. For fixed $\mathcal{N}$ and $\mathcal{K}$, the state transition of the Markov chain $\boldsymbol{X}^N(t)$ can be captured by its embedded *jump process* $(X^{J,N}(l) : l \in \mathbb{N})$ defined as

$$\boldsymbol{X}^{J,N}(l) = \left[X_1^{J,N}(l), X_2^{J,N}(l), \ldots, X_K^{J,N}(l)\right] \tag{13}$$

[1] An absorbing state is a fixed point or steady state that, once reached, the system never leaves.

where for $\forall k \in \mathcal{K}$, $\boldsymbol{X}_k^{J,N}(l) \in \mathbb{Z}$ is the number of agents preferring arm k at the lth jump. Thus, the success event E^N can be expressed as follows:

$$\begin{aligned} E^N &= \left\{ \lim_{t\to\infty} \boldsymbol{X}^N(t) = x^* \right\} \\ &= \left\{ \lim_{l\to\infty} \boldsymbol{X}^{J,N}(l) = x^* \right\} \\ &= \left\{ \lim_{l\to\infty} X_1^{J,N}(l) = N \right\}. \end{aligned} \tag{14}$$

The coordinate process $(X_1^{J,N}(l) : l \in \mathbb{N})$ can be seen as a random walk. For each jump, $X_1^{J,N}$ either increases by one or decreases by one, or stays unchanged. Denote by $(W(l) : l \in \mathbb{N})$ the *jump process* of the coordinate process $(X_1^{J,N}(l) : l \in \mathbb{N})$. For each jump, $(W(l) : l \in \mathbb{N})$ either increases by one or decreases by one.

We next consider a standard random walk $(\widehat{W}(l) : l \in \mathbb{N})$ where $\widehat{W}(l) \in \mathbb{N}$ is defined as

$$\widehat{W}(l+1) = \begin{cases} \widehat{W}(l) + 1, & \text{with probability } p \\ \widehat{W}(l) - 1, & \text{with probability } 1-p \end{cases} \tag{15}$$

and $(1/2) < p \leq 1$. Let S be the event that $(\widehat{W}(l) : l \in \mathbb{N})$ converges to infinity without ever reaching 0. The following result has been proved in [21].

Proposition 1: Consider a standard biased random walk $(\widehat{W}(l) : l \in \mathbb{N})$ with initial value $\widehat{W}(0) = z_0$ (z_0 is a positive integer) and $(1/2) < p \leq 1$. Then

$$\mathbb{P}\left(S \middle| \widehat{W}(0) = z_0\right) = 1 - \left(\frac{1-p}{p}\right)^{z_0}. \tag{16}$$

Define B_1 and D_1 as the birth rate and the death rate for the best arm (Arm 1). Based on the generator matrix G^N, we have $B_1 = -\sum_{\boldsymbol{s}' \,:\, s_1'=s_1+1} g_{s,s'}$ and $D_1 = -\sum_{\boldsymbol{s}' \,:\, s_1'=s_1-1} g_{s,s'}$. Using a similar proof as that in [34], we can bound the probability for the *jump process* $(W(l) : l \in \mathbb{N})$ to move one step up, i.e., $\mathbb{P}(W(l+1) = W(l)+1)$, using the birth rate and death rate, which is formalized in the following lemma.

Lemma 1: Consider the *jump process* $(W(l) : l \in \mathbb{N})$, for a fixed $l \in \mathbb{N}$

$$\mathbb{P}(W(l+1) = W(l)+1) \geq \frac{B_1}{B_1 + D_1}. \tag{17}$$

Proof: For ease of exposition, for a fixed $k \in \mathbb{Z}_+$, define

$$F^k \triangleq \{\omega : W(k+1) = W(k) + 1 \text{ given } W(k) \notin \{0, N\}\}.$$

We first link the random walk back to the first-order space-time structure of the original continuous-time Markov chain as follows:

$$\begin{aligned} F^k &= \{\omega : W(k+1) = W(k) + 1 \text{ given } W(k) \notin 0, N\} \\ &= \cup_{l=1}^{\infty} \{\omega : \text{the } k+1\text{th move of } W \text{occurs at the} \\ &\qquad \times\, l+1\text{th jump of } X^{J,N} \;\&\; X_1^{J,N}(l+1) \\ &\qquad = X_1^{J,N}(l) + 1 \text{given } W(k) \notin \{0, N\}\Big\}. \end{aligned}$$

For ease of exposition, define G_l^k as

$$\begin{aligned} G_l^k &= \{\omega : \text{the } k+1\text{th move of } W \text{ occurs at the} \\ &\qquad \times\, l+1\text{th jump of } X^{J,N} \;\&\; X_1^{J,N}(l+1) \\ &\qquad = X_1^{J,N}(l) + 1 \text{given } W(k) \notin \{0, N\}\Big\} \\ &= \{\omega : \text{the } k+1\text{th move of } W \text{ occurs at the} \\ &\qquad \times\, l+1\text{th jump of } X^{J,N} \;\&\; X_1^{J,N}(l+1) \\ &\qquad = X_1^{J,N}(l) + 1 \text{given } X_1^{J,N} \notin \{0, N\}\Big\}. \end{aligned}$$

For $l < k$, $G_l^k = \varnothing$. Then, we can obtain $F^k = \cup_{l \geq k} G_l^k$. Furthermore, according to the definition, $G_l^k \cap G_{l'}^k = \varnothing$ $\forall l \neq l'$. So

$$\mathbb{P}\Big\{F^k\Big\} = \mathbb{P}\Big\{\cup_{l\geq k} G_l^k\Big\} = \sum_{l \geq k} \mathbb{P}\Big\{G_l^k\Big\}. \tag{18}$$

Next, we concentrate on $\mathbb{P}\{G_l^k\}$. Recalling s defined above, we establish a relationship between $\mathbb{P}\{G_l^k\}$ and s as follows:

$$\begin{aligned} \mathbb{P}\Big\{G_l^k\Big\} = \sum_{s \in \mathcal{S}^N} &\mathbb{P}\Big\{X^{J,N}(l) = s \Big| X_1^{J,N}(l) \notin \{0, N\}\Big\} \\ &\times \mathbb{P}\Big\{G_l^k | X^{J,N}(l) = s\Big\} \end{aligned} \tag{19}$$

where

$$\sum_{s \in \mathcal{S}^N} \mathbb{P}\Big\{X^{J,N}(l) = s \Big| X_1^{J,N}(l) \notin \{0, N\}\Big\} = 1.$$

Now, we define event O_l^k as follows:

$$\begin{aligned} O_l^k \triangleq \{\omega : &\text{the } k+1\text{th move of } W \text{ occurs at the} \\ &\times\, l+1\text{th jump of } X^{J,N} \text{ given } X_1^{J,N}(l) \notin \{0, N\}\Big\}. \end{aligned}$$

For a fixed k, we have

$$\sum_{l \geq k} \mathbb{P}\Big\{O_l^k\Big\} = 1.$$

It is easy to see that for $l \geq k$

$$\begin{aligned} &\mathbb{P}\Big\{G_l^k | X^{J,N}(l) = s\Big\} \\ &= \mathbb{P}\Big\{O_l^k \big| X^{J,N}(l) = s\Big\} \\ &\quad \times \mathbb{P}\Big\{X_1^{J,N}(l+1) = X_1^{J,N}(l) + 1 \Big| \quad O_l^k, X^{J,N}(l) = s\Big\}. \end{aligned} \tag{20}$$

Then, we have

$$\begin{aligned} &\mathbb{P}\Big\{X_1^{J,N}(l+1) = X_1^{J,N}(l) + 1 | O_l^k, X^{J,N}(l) = s\Big\} \\ &= \mathbb{P}\Big\{X_1^{J,N}(l+1) = X_1^{J,N}(l) + 1 | X^{J,N}(l) = s \\ &\qquad \times\, s_1 \notin \{0, N\}, \text{the } k+1\text{th move of } W \text{ occurs at the} \\ &\qquad \times\, l+1\text{th jump of} X^{J,N} \text{ given } W(k) \notin \{0, N\}\Big\} \\ &= \mathbb{P}\Big\{X_1^{J,N}(l+1) = X_1^{J,N}(l) + 1 | X^{J,N}(l) = s \\ &\qquad \times\, s_1 \notin \{0, N\} \text{ one move of } W \text{ occurs at the} \\ &\qquad \times\, l+1\text{th jump of } X^{J,N}, \text{and there are } k \text{ moves of } W \\ &\qquad \times\, \text{occur among the first } l\text{jumps of } X^{J,N}\Big\} \end{aligned}$$

$$\begin{aligned}
&\overset{(a)}{=} \mathbb{P}\Big\{X_1^{J,N}(l+1) = X_1^{J,N}(l)+1 | X^{J,N}(l)=s \\
&\qquad \times s_1 \notin \{0,N\} \text{ one move of } W \text{ occurs at the} \\
&\qquad \times\ l+1\text{th jump of } X^{J,N}\Big\} \\
&= \frac{\mathbb{P}\Big\{X_1^{J,N}(l+1) = X_1^{J,N}(l)+1 | X^{J,N}(l)=s, s\in\mathcal{S}^N\Big\}}{\mathbb{P}\left\{\begin{array}{c}\text{one move of } W \text{ occurs at the } l+1\text{thjump of} \\ \times\ X^{J,N} | X^{J,N}(l)=s, s\in\mathcal{S}^N\end{array}\right\}} \qquad (21)
\end{aligned}$$

where equality (a) follows the Markov property of $X^{J,N}$. We know

$$\begin{aligned}
&\mathbb{P}\Big\{X_1^{J,N}(l+1) = X_1^{J,N}(l)+1 | X^{J,N}(l)=s, s\in\mathcal{S}^N\Big\} \\
&\quad = \frac{B_1}{\mathbb{P}\big\{X^{J,N}(l)=s, s\in\mathcal{S}^N\big\}}. \qquad (22)
\end{aligned}$$

Meanwhile, we have

$$\begin{aligned}
&\mathbb{P}\{\text{one move of } W \text{ occurs at the } l+1\text{th} \\
&\qquad \times \text{ jump of } X^{J,N} | X^{J,N}(l)=s, s\in\mathcal{S}^N\big\} \\
&\quad = \frac{B_1+D_1}{\mathbb{P}\big\{X^{J,N}(l)=s, s\in\mathcal{S}^N\big\}}. \qquad (23)
\end{aligned}$$

Thus, by (21)–(23), we have

$$\begin{aligned}
&\mathbb{P}\Big\{X_1^{J,N}(l+1) = X_1^{J,N}(l)+1 | O_l^k, X^{J,N}(l)=s\Big\} \\
&\quad = \frac{B_1}{B_1+D_1}.
\end{aligned}$$

In particular, (20) becomes

$$\mathbb{P}\Big\{G_l^k | X^{J,N}(l)=s\Big\} = \mathbb{P}\Big\{O_l^k | X^{J,N}(l)=s\Big\}\frac{B_1}{B_1+D_1}. \qquad (24)$$

By (18), (19), and (24), we have

$$\begin{aligned}
\mathbb{P}\Big\{F^k\Big\} &= \sum_{l\geq k}\sum_{s\in\mathcal{S}^N} \mathbb{P}\Big\{X^{J,N}(l)=s | X_1^{J,N}(l)\notin\{0,N\}\Big\} \\
&\qquad \mathbb{P}\Big\{G_l^k | X^{J,N}(l)=s\Big\} \\
&= \sum_{l\geq k}\sum_{s\in\mathcal{S}^N} \mathbb{P}\Big\{X^{J,N}(l)=s | X_1^{J,N}(l)\notin\{0,N\}\Big\} \\
&\qquad \mathbb{P}\Big\{O_l^k | X^{J,N}(l)=s\Big\}\frac{B_1}{B_1+D_1} \\
&= \frac{B_1}{B_1+D_1}\sum_{l\geq k}\sum_{s\in\mathcal{S}^N} \mathbb{P}\Big\{X^{J,N}(l)=s, O_l^k\Big\} \\
&= \frac{B_1}{B_1+D_1}\sum_{l\geq k}\sum_{s\in\mathcal{S}^N} \mathbb{P}\Big\{O_l^k\Big\}\mathbb{P}\Big\{X^{J,N}(l)=s | O_l^k\Big\} \\
&= \frac{B_1}{B_1+D_1}.
\end{aligned}$$

Therefore, the proof of Lemma 1 is complete. ■

We next intend to bound $\mathbb{P}(W(l+1)=W(l)+1)$ based on Lemma 1. First, we bound the deviation between the estimated popularity of the best arm and the true value in the following lemma.

Lemma 2: When an agent performs the estimation in the algorithm at time t, with probability at least $1-2K\exp(-2R^2N)$, it holds

$$\left|\widehat{F}_1^t - F_1^t\right| \leq \delta = \frac{\mu(1-\alpha)}{2K(2-\mu)+\mu} \qquad (25)$$

where $R = (1-(2/[\exp(\varepsilon/2)+1]))([\mu(1-\alpha)]/[(2K+1)[2K(2-\mu)+\mu]])$ is a constant.

Proof: By the perturbation formula given in (5), for $\forall i\in\mathcal{N}$ and a given $k\in\mathcal{K}$, we have

$$\begin{aligned}
\mathbb{E}\big[V_{i,k}^t\big] &= F_k^t\cdot\frac{\exp(\varepsilon/2)}{\exp(\varepsilon/2)+1} + \big(1-F_k^t\big)\cdot\frac{1}{\exp(\varepsilon/2)+1} \\
&= F_k^t\cdot\frac{\exp(\varepsilon/2)-1}{\exp(\varepsilon/2)+1} + \frac{1}{\exp(\varepsilon/2)+1}. \qquad (26)
\end{aligned}$$

Then, the kth component of the perturbed vector $\boldsymbol{H_i^t}$ can be calculated as

$$\begin{aligned}
\mathbb{E}\big[H_{i,k}^t\big] &= \frac{1}{N}\cdot\mathbb{E}\left[\sum_{i\in\mathcal{N}} V_{i,k}^t\right] \\
&= \frac{1}{N}\cdot\sum_{i\in\mathcal{N}}\mathbb{E}\big[V_{i,k}^t\big] \\
&= \frac{1}{N}\cdot N\cdot F_k^t\cdot\left(\frac{\exp(\varepsilon/2)-1}{\exp(\varepsilon/2)+1}+\frac{1}{\exp(\varepsilon/2)+1}\right) \\
&= F_k^t\cdot\frac{\exp(\varepsilon/2)-1}{\exp(\varepsilon/2)+1}+\frac{1}{\exp(\varepsilon/2)+1}. \qquad (27)
\end{aligned}$$

Using Hoeffding's inequality, we have

$$\mathbb{P}\big(\big|H_{i,k}^t - \mathbb{E}\big[H_{i,k}^t\big]\big|\leq R\big)\geq 1-2\exp\Big(-2R^2N\Big). \qquad (28)$$

Take the union bound on all $k\in\mathcal{K}$, we can get

$$\mathbb{P}\big(\big|H_{i,k}^t - \mathbb{E}\big[H_{i,k}^t\big]\big|\leq R\big)\geq 1-2K\exp\Big(-2R^2N\Big). \qquad (29)$$

Since for $\forall k\in K$

$$\left(\frac{\exp(\varepsilon/2)+1}{\exp(\varepsilon/2)-1}\mathbb{E}\big[H_{i,k}^t\big]-\frac{1}{\exp(\varepsilon/2)-1}\right) = F_k^t \qquad (30)$$

with probability at least $1-2K\exp(-2R^2N)$, the following holds:

$$\begin{aligned}
\big|\widetilde{F}_k^t - F_k^t\big| &\leq \left|\left(\frac{\exp(\varepsilon/2)+1}{\exp(\varepsilon/2)-1}H_{i,k}^t-\frac{1}{\exp(\varepsilon/2)-1}\right)-F_k^t\right| \\
&= \left|\left(\frac{\exp(\varepsilon/2)+1}{\exp(\varepsilon/2)-1}H_{i,k}^t-\frac{1}{\exp(\varepsilon/2)-1}\right)\right. \\
&\quad \left.-\left(\frac{\exp(\varepsilon/2)+1}{\exp(\varepsilon/2)-1}\mathbb{E}\big[H_{i,k}^t\big]-\frac{1}{\exp(\varepsilon/2)-1}\right)\right| \\
&= \left|\frac{\exp(\varepsilon/2)+1}{\exp(\varepsilon/2)-1}H_{i,k}^t-\frac{\exp(\varepsilon/2)+1}{\exp(\varepsilon/2)-1}\mathbb{E}\big[H_{i,k}^t\big]\right| \\
&= \frac{\exp(\varepsilon/2)+1}{\exp(\varepsilon/2)-1}\big|H_{i,k}^t - E\big[H_{i,k}^t\big]\big| \\
&\leq \frac{\exp(\varepsilon/2)+1}{\exp(\varepsilon/2)-1}R \\
&= \frac{\mu(1-\alpha)}{(2K+1)[2K(2-\mu)+\mu]}. \qquad (31)
\end{aligned}$$

Let $\delta' = ([\mu(1-\alpha)]/[(2K+1)[2K(2-\mu)+\mu]])$. Note that $K\cdot\delta' < (1/2)$. After normalizing $\widetilde{F}_k^t$ by $\widehat{F}_k^t = \widetilde{F}_k^t/\sum_{j=1}^K\widetilde{F}_j^t$, we have

$$\big|\widehat{F}_k^t - \widetilde{F}_k^t\big| = \left|\frac{\widetilde{F}_k^t}{\sum_{j=1}^K\widetilde{F}_j^t} - \widetilde{F}_k^t\right|$$

$$
\begin{aligned}
&= \left| \frac{\widetilde{F}_k^t}{\sum_{j=1}^K \left(F_j^t - \delta\right)'} - \widetilde{F}_k^t \right| \\
&\le \frac{\widetilde{F}_k^t}{1 - K\delta'} - \widetilde{F}_k^t \\
&\le \left(1 + 2K\delta'\right)\widetilde{F}_k^t - \widetilde{F}_k^t \\
&\le 2K\delta'. \qquad (32)
\end{aligned}
$$

Hence, with probability at least $1 - 2K\exp(-2R^2N)$

$$
\begin{aligned}
\left|\widehat{F}_k^t - F_k^t\right| &= \left|\widehat{F}_k^t - \widetilde{F}_k^t + \widetilde{F}_k^t - F_k^t\right| \\
&\le \left|\widehat{F}_k^t - \widetilde{F}_k^t\right| + \left|\widetilde{F}_k^t - F_k^t\right| \\
&= (2K+1)\delta' \\
&= \frac{\mu(1-\alpha)}{2K(2-\mu)+\mu} \qquad (33)
\end{aligned}
$$

by which we can get the result in the lemma. ■

Now we are ready to bound $\mathbb{P}(W(l+1) = W(l)+1)$ based on Lemma 1. We first consider the case that $\widehat{F}_k^t < \alpha$ for $\forall k \in \mathcal{K}$.

Lemma 3: If for a given $l \in \mathbb{N}$, $W(l) > (N/2K)$ and $\widehat{F}_k^t < \alpha$ for $\forall k \in \mathcal{K}$, where t is the time when $W(l)$ jumps to $W(l+1))$, then it holds that

$$
\mathbb{P}(W(l+1) = W(l)+1) \ge \frac{2}{4-\mu} > 1/2. \qquad (34)
$$

Proof: By Lemma 2, it can be concluded (w.h.p) that $|\widehat{F}_1^t - F_1^t| \le \delta = ([\mu(1-\alpha)]/[2K(2-\mu)+\mu])$. Then it can be obtained that

$$
\begin{aligned}
\frac{|\widehat{F}_1^t - F_1^t|}{F_1^t} &< \frac{\delta}{\frac{N}{2K}\cdot\frac{1}{N}} \\
&= 2K\cdot\delta \\
&= \frac{(1-\alpha-\delta)\mu}{2-\mu} \\
&< \frac{(1-\alpha-\delta)\left(\Delta - \frac{\mu}{2}\varphi_1\right)}{\left(1-\frac{\mu}{2}\right)\varphi_1 - (\alpha+\delta)\left(\Delta - \frac{\mu}{2}\varphi_1\right)} \\
&< \frac{(1-\alpha-\delta)\left(\Delta - \frac{\mu}{2}\varphi_1\right)}{\left(1-\frac{\mu}{2}\right)\varphi_1 - (\alpha+\delta)\left(\Delta - \frac{\mu}{2}\varphi_1\right)}. \qquad (35)
\end{aligned}
$$

By the above inequality, we can set $\widehat{F}_1^t = \gamma \cdot F_1^t$, where $\gamma \in (\sigma_1, \sigma_2)$ and $\sigma_1 = \varphi_2(1 - [\mu/2])\varphi_1 - (\alpha+\delta)(\Delta - [\mu/2]\varphi_1)$, $\sigma_2 = ([(2-\mu)\varphi_1 - \varphi_2 - 2(\alpha+\delta)(\Delta - [\mu/2]\varphi_1)]/[(1 - [\mu/2])\varphi_1 - (\alpha+\delta)(\Delta - [\mu/2]\varphi_1)])$.

We next bound the probability $\mathbb{P}(W(l+1) = W(l)+1)$ in two cases, $\gamma \in (\sigma_1, 1)$ and $\gamma \in [1, \sigma_2)$.

Case 1 [$\gamma \in (\sigma_1, 1)$]:

In this case, by Lemma 1

$$
\begin{aligned}
&\mathbb{P}(W(l+1) = W(l)+1) \\
&\ge \frac{B_1}{B_1 + D_1} \\
&= \frac{\left(\sum_{j\ge2} s_j\lambda\right)\widehat{F}_1^t\varphi_1}{\left(\sum_{j\ge2} s_j\lambda\right)\widehat{F}_1^t\varphi_1 + s_1\lambda\left(\sum_{j\ge2}\widehat{F}_j^t\varphi_j\right)} \\
&\ge \frac{(N-s_1)\gamma F_1^t\varphi_1}{(N-s_1)\gamma F_1^t\varphi_1 + s_1\varphi_2\left(1-\gamma F_1^t\right)} \\
&= \frac{(N-s_1)\gamma\cdot\frac{s_1}{N}\cdot\varphi_1}{(N-s_1)\gamma\cdot\frac{s_1}{N}\cdot\varphi_1 + s_1\varphi_2\left(1-\gamma\cdot\frac{s_1}{N}\right)} \\
&= \frac{(N-s_1)\gamma\varphi_1}{(N-s_1)\gamma\varphi_1 + \varphi_2(N-\gamma s_1)} \\
&= \frac{1}{1+\frac{\varphi_2(N-\gamma s_1)}{\varphi_1\gamma(N-s_1)}} \\
&= \frac{1}{1+\frac{\varphi_2(N/\gamma-s_1)}{\varphi_1(N-s_1)}} \\
&\ge \frac{1}{1+\frac{\varphi_2\left(N\cdot\frac{\left(1-\frac{\mu}{2}\right)\varphi_1-(\alpha+\delta)\left(\Delta-\frac{\mu}{2}\varphi_1\right)}{\varphi_2}-s_1\right)}{\varphi_1(N-s_1)}} \\
&= \frac{1}{1+\frac{N\left(1-\frac{\mu}{2}\right)\varphi_1-(\alpha+\delta)N\left(\Delta-\frac{\mu}{2}\varphi_1\right)-s_1\varphi_2}{N\varphi_1-s_1\varphi_1}} \\
&= \frac{1}{1+\frac{N\cdot\frac{2-\mu}{2}\varphi_1-(\alpha+\delta)N\left(\frac{2-\mu}{2}\varphi_1-\varphi_2\right)-s_1\varphi_2}{N\varphi_1-s_1\varphi_1}} \\
&\ge \frac{1}{1+\frac{N\cdot\frac{2-\mu}{2}\varphi_1-s_1\left(\frac{2-\mu}{2}\varphi_1-\varphi_2\right)-s_1\varphi_2}{N\varphi_1-s_1\varphi_1}} \\
&= \frac{1}{1+\frac{(N-s_1)\varphi_1}{(N-s_1)\varphi_1}\cdot\frac{2-\mu}{2}} \\
&= \frac{2}{4-\mu}.
\end{aligned}
$$

The last inequality holds because $\widehat{F}_1^t < \alpha$ and $s_1 < (\widehat{F}_1^t + \delta)N < (\alpha+\delta)N$.

Case 2 [$\gamma \in [1, \sigma_2)$]:

In this case, similarly, by Lemma 1, we can get that

$$
\begin{aligned}
&\mathbb{P}(W(l+1) = W(l)+1) \\
&\ge \frac{B_1}{B_1 + D_1} \\
&= \frac{\left(\sum_{j\ge2} s_j\lambda\right)\widehat{F}_1^t\varphi_1}{\left(\sum_{j\ge2} s_j\lambda\right)\widehat{F}_1^t\varphi_1 + s_1\lambda\left(\sum_{j\ge2}\widehat{F}_j^t\varphi_j\right)} \\
&= \frac{(N-s_1)\widehat{F}_1^t\varphi_1}{(N-s_1)\widehat{F}_1^t\varphi_1 + s_1\left(\sum_{j\ge2}\widehat{F}_j^t\varphi_j\right)} \\
&\ge \frac{(N-s_1)\widehat{F}_1^t\varphi_1}{(N-s_1)\widehat{F}_1^t\varphi_1 + s_1\varphi_2\sum_{j\ge2}\widehat{F}_j^t} \\
&= \frac{(N-s_1)\gamma F_1^t\varphi_1}{(N-s_1)\gamma F_1^t\varphi_1 + s_1\varphi_2\left(1-\gamma F_1^t\right)} \\
&\ge \frac{(N-s_1)\gamma F_1^t\varphi_1}{(N-s_1)\gamma F_1^t\varphi_1 + s_1\varphi_2\left(1-F_1^t\right)} \\
&= \frac{(N-s_1)\gamma\frac{s_1}{N}\varphi_1}{(N-s_1)\gamma\frac{s_1}{N}\varphi_1 + \frac{s_1}{N}\varphi_2(N-s_1)} \\
&= \frac{\gamma\varphi_1}{\gamma\varphi_1+\varphi_2} \\
&\ge \frac{\varphi_1}{\varphi_1+\varphi_2}.
\end{aligned}
$$

Note that

$$
\begin{aligned}
\left(1-\frac{\mu}{2}\right) - \frac{\varphi_2}{\varphi_1} &= \frac{\varphi_1(2-\mu)-2\varphi_2}{2\varphi_1} \\
&= \frac{2\Delta - \varphi_1\mu}{2\varphi_1}
\end{aligned}
$$

$$> \frac{2\Delta - \varphi_1\Delta}{2\varphi_1} = \frac{(2-\varphi_1)\Delta}{2\varphi_1} > 0.$$

Then it can be obtained that

$$\frac{\varphi_1}{\varphi_1+\varphi_2} - \frac{2}{4-\mu} = \frac{1}{1+\frac{\varphi_2}{\varphi_1}} - \frac{1}{1+1-\frac{\mu}{2}} > 0.$$

From above, we can get that

$$\mathbb{P}(W(l+1) = W(l)+1) \geq \frac{\varphi_1}{\varphi_1+\varphi_2} > \frac{2}{4-\mu}.$$

Combining the results in the above two cases, the lemma can be proved. ∎

Based on above lemmas, we can show the learnability of the proposed collaborative learning process, as shown in the following theorem.

Theorem 1: Considering the random process $(\boldsymbol{X}^N(t) : t \in \mathbb{R}^+)$ defined in (10) and the event E^N defined in (14), for any $\mu > 0$, it holds that

$$\mathbb{P}\{E^N\} \geq 1 - \left(1 - \frac{\mu}{2}\right)^{N/(2K)} \tag{36}$$

and as $N \to \infty$

$$\mathbb{P}\{E^N\} \to 1. \tag{37}$$

Proof: We consider the embedded random walk $(W(l) : l \in \mathbb{N})$. Initially, for $\forall k \in \mathcal{K}$, $F_k^0 = (1/K) < \alpha$ and $W(0) = (N/K) > (N/2K)$. By Lemma 3, $\mathbb{P}(W(l+1) = W(l)+1) \geq (2/4-\mu) > (1/2)$.

Coupling the walk $(W(l) : l \in \mathbb{N})$ with the standard walk $(\widehat{W}(l) : l \in \mathbb{N})$ and by Proposition 1, with probability at least $1-([1-(2/4-\mu)]/](2/4-\mu)])^{N/(2K)} = 1-(1-[\mu/2])^{N/(2K)}$, $W(l)$ converges to infinity without ever reaching $N/(2K)$.

When $W(l)$ reaches the range $[(\alpha-\delta)N, (\alpha+\delta)N)$, i.e., $F_1^t \in [\alpha-\delta, \alpha+\delta)$, $\widehat{F}_1^t \in [F_1^t - \delta, F_1^t + \delta]$, we consider two cases. If $\widehat{F}_1^t < \alpha$, $W(l)$ will move one step up with probability at least $(2/4-\mu)$ by Lemma 3. If $\widehat{F}_1^t \geq \alpha$, it can be obtained that $B_1 > 0$ and $D_1 = 0$. Hence, $W(l)$ will move one step up with probability of 1.

Combining the above together, with probability at least $1-(1-[\mu/2])^{N/(2K)}$, $W(l)$ will grow to $(\alpha+\delta)N$. After that, $W(l)$ will increase monotonically until reaching N, which is an absorbing state, since both birth rate and death rate for arm 1 are zero in this state. As N increases, the error probability for learnability decreases exponentially and all the agents can learn the best arm with a high probability. ∎

B. Privacy Preserving

In our learning process, we consider the setting that the agents' preferences as their privacy and seek to conceal their true preferences from other peers. To this end, each agent provides a perturbed indicating vector $\boldsymbol{V}_i \in \{0,1\}^K$ to the requester, instead of the raw indicating vector $\boldsymbol{U}_i$. We claim that this manner of perturbation satisfies the requirement of LDP.

Theorem 2: The perturbation mechanism described by (5) preserves ε-LDP in each message transmission for each agent.

Proof: Denote our perturbation mechanism by $\mathcal{M}$, which takes agents' raw indicating vectors as input and outputs the perturbed vectors. For any input u_1, u_2 and any possible output $v \in \{0,1\}^K$, we have

$$\begin{aligned}
&\frac{\mathbb{P}[\mathcal{M}(u_1)=v]}{\mathbb{P}[\mathcal{M}(u_2)=v]} \\
&= \frac{\left(\frac{\exp(\varepsilon/2)}{\exp(\varepsilon/2)+1}\right)^{K-\|v-u_1\|_1}\left(\frac{1}{\exp(\varepsilon/2)+1}\right)^{\|v-u_1\|_1}}{\left(\frac{\exp(\varepsilon/2)}{\exp(\varepsilon/2)+1}\right)^{K-\|v-u_2\|_1}\left(\frac{1}{\exp(\varepsilon/2)+1}\right)^{\|v-u_2\|_1}} \\
&= \frac{\exp\left[\frac{\varepsilon}{2}(K-\|v-u_1\|_1)\right]}{\exp\left[\frac{\varepsilon}{2}(K-\|v-u_2\|_1)\right]} \\
&= \exp\left[\frac{\varepsilon}{2}(\|v-u_2\|_1 - \|v-u_1\|_1)\right] \\
&\leq \exp\left[\frac{\varepsilon}{2}(\|u_1-u_2\|_1)\right] \\
&\leq \exp(\varepsilon)
\end{aligned}$$

where the second-to-last inequality is by the triangle inequality and the last inequality is by the fact that u_1 and u_2 are distinct in at most two components, which is obvious because that any input vector u has only one component being 1 with the remainders being 0.

Hence, for any subset of output S

$$\begin{aligned}
\mathbb{P}[\mathcal{M}(u_1) \in S] &= \sum_{v\in S}\mathbb{P}[\mathcal{M}(u_1)=v] \\
&\leq \sum_{v\in S}\exp(\varepsilon)\mathbb{P}[\mathcal{M}(u_2)=v] \\
&= \exp(\varepsilon)\sum_{v\in S}\mathbb{P}[\mathcal{M}(u_2)=v] \\
&= \exp(\varepsilon)\mathbb{P}[\mathcal{M}(u_2)\in S]
\end{aligned}$$

i.e.,

$$\frac{\mathbb{P}[\mathcal{M}(u_1)\in S]}{\mathbb{P}[\mathcal{M}(u_2)\in S]} \leq \exp(\varepsilon)$$

following which we conclude the proof. ∎

VI. Experiments

In this section, we evaluate the performance of our learning algorithms in real setting. Specifically, we examine the impact of different parameters on the convergence of the learning process, including the number of agents N, the number of arms K, and the LDP parameter ε. The experiments are conducted on a laptop with Intel Core i5-7200U and 8 GB of RAM. Each result reported is the average value of over 3000 times of experiments.

In Fig. 1, we use $X_1^N(t)$ and $Y_1^N(t)$ to represent the number of and the fraction of agents preferring the best arm (the first arm) at time $t \in \mathbb{R}^+$. So $Y_1^N(t) = ([X_1^N(t)]/N)$. In the experiments, we set the rate of Poisson clock $\lambda = 1$.

In Fig. 1, we illustrate the performance of our algorithm by setting $N = 400$, $K = 2$, $\lambda = 1$, $\varepsilon = 3$, and $(\varphi_1, \varphi_2, \varphi_3, \varphi_4) = (0.95, 0.65, 0.35, 0.05)$. It can be seen that the learning process converges exponentially fast. We consider this setting as the

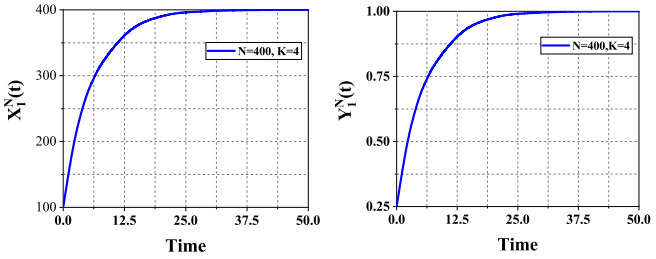


Fig. 1. Convergence of PPCL ($N = 400$, $K = 4$, and $\lambda = 1$).

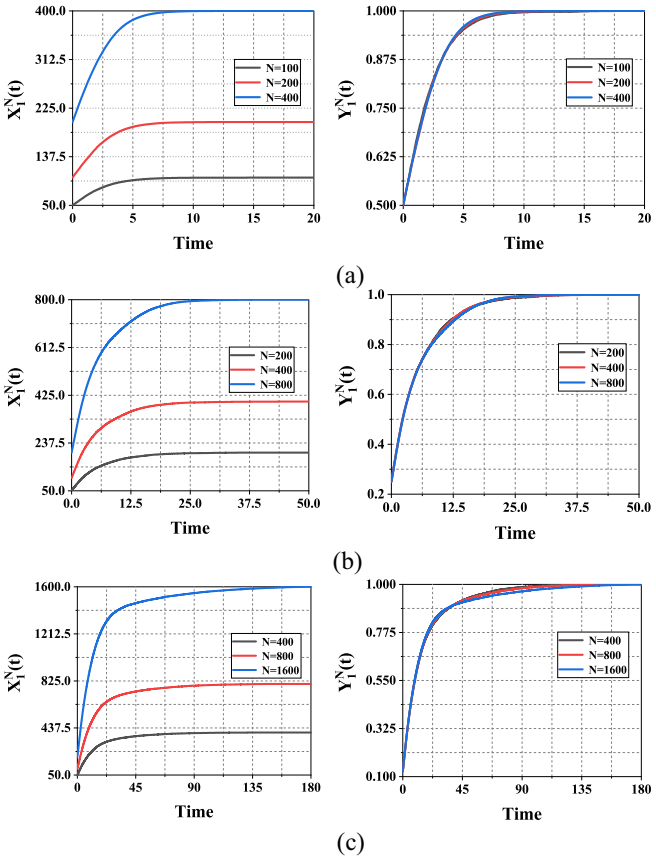


Fig. 2. Convergence of PPCL when N changes (a) $K = 2$, $\varepsilon = 3$, and $\lambda = 1$. (b) $K = 4$, $\varepsilon = 3$, and $\lambda = 1$. (c) $K = 8$, $\varepsilon = 3$, and $\lambda = 1$.

baseline scenario and study the impact of different parameters on the performance of PPCL base on it.

We first evaluate the impact of the number of agents N on the convergence of the learning process. The results are shown in Fig. 2. The learning processes were executed in the settings of $K = 2, 4, 8$ and $\varepsilon = 3$. The arm qualities in the cases of $K = 2, 4, 8$ were set as (0.8,0.2), (0.95,0.65,0.35,0.05), (0.92,0.80,0.68,0.56,0.44,0.32,0.20,0.08), respectively. As shown in the figure, it can be seen that in all settings, the learning process converges very quickly. For example, when there are four agents, all agents have learned the best arm in around 20 rounds. Furthermore, as shown in the figures, when the number of agents is sufficiently large compared with the number of arms, the impact of N on the convergence rate is very limited as the number of agents changes.

In Fig. 3, the convergence of the learning process under settings with different number of arms are illustrated. In the

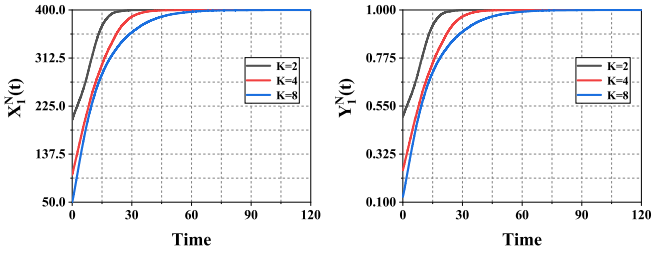


Fig. 3. Convergence of PPCL when K changes ($N = 400$, $\varepsilon = 3$, and $\lambda = 1$).

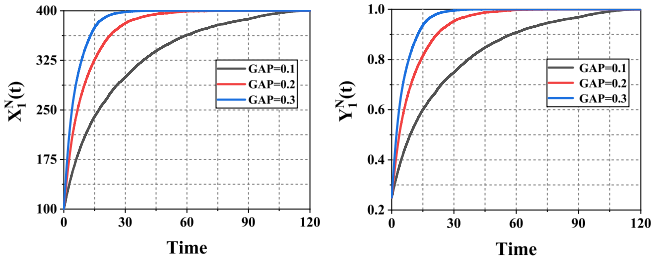


Fig. 4. Convergence of PPCL when the quality gap changes ($K = 4$, $\varepsilon = 3$, and $\lambda = 1$).

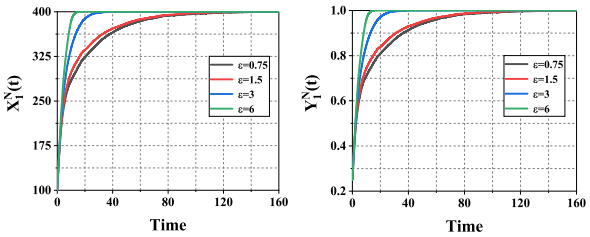


Fig. 5. Convergence of PPCL when ε changes ($N = 400$, $K = 4$, and $\lambda = 1$).

experiments, the parameters were set as $N = 400$, $\varepsilon = 3$. The quality of the best arm (i.e., φ_1) was set as 0.9 and the quality gap is fixed to 0.1. From the figures, it can be seen that as the number of arms increases, the convergence speed is decreased. The number of rounds for convergence is increased by around two times as the number of arms change from 2 to 8.

In Fig. 4, the impact of the quality gap is evaluated. In the experiments, the parameters were set as $N = 400$, $K = 4$, and $\varepsilon = 3$. The quality of the best arm (i.e., φ_1) is set as 0.95 and the gap varies from 0.1 to 0.3. From the figures, it can be found that the quality gap significantly affect the convergence rate. The convergence gets faster as the quality gap becomes larger. When the gap increases from 0.1 to 0.3, the number of rounds for convergence is reduced from around 110 to 30.

The impact of the privacy loss ε is shown in Fig. 5. In the experiments, the parameters were set as $N = 400$ and $K = 4$. From the figures, it can be seen that as the privacy loss becomes larger, the convergence rate can be significantly improved. This is because, the larger the privacy loss is, the more accurate popularity estimate of the arms the agents can obtain in the learning process, such that they can adopt a better selection.

Finally, we compare our algorithm with the one we called CBL given in [34] without privacy preserving. We first describe the CBL algorithm and then present the experiment results.

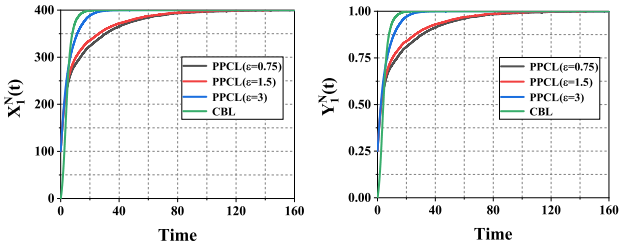


Fig. 6. Comparison of PPCL and CBL.

In CBL, each agent i keeps its local preference $C_i \in \mathcal{K}$. Initially $\forall i \in \mathcal{N}$, $C_i = 0$. When the clock at agent i ticks at time t, agent i refines its preference C_i by picking an arm a to pull and then deciding whether to substitute C_i with the picked a. If $C_i = 0$, agent i picks an arm a uniformly at random from $\mathcal{K}$ (with probability τ) or chooses one peer j (including itself) uniformly at random and picks j's preference C_j as a (with probability $1 - \tau$), where $\tau \in (0, 1]$. If $C_i = 1$, agent i only chooses one peer uniformly at random and picks the peer's preference as a. Agent i pulls arm a. If the reward is 1, i substitutes a for C_i. Otherwise, i keeps C_i unchanged.

The comparison between our proposed PPCL and CBL in [34] are presented in Fig. 6. It can be seen that our algorithm takes more time to converge than CBL. But if the privacy loss ε is not very small, the time increased for convergence can be acceptable. For example, when $\varepsilon = 3$, the convergence time increases only for ten rounds.

In summary, the experimental results show that even if we introduce a mechanism to preserve the privacy of agents, the learning process can still converge fast.

VII. Conclusion

By proposing a privacy-preserving collaborative learning algorithm for the MAB problem, this work showed that collaborative learning can be executed efficiently in decentralized IoT networks, though harsh constraints, such as weak capacity of devices, asynchronous communications, and the requirement on privacy preserving, make it much harder to devise collaborative learning algorithms. The proposed algorithm can converge fast while preserving the privacy of agents during communications. To the best of our knowledge, our algorithm is the first one on collaborative learning for MAB under the framework of LDP. Extensive experiments illustrated that the proposed algorithm can converge fast in real settings.

This work initializes the privacy-preserving collaborative learning algorithm studies for fundamental machine learning problems in the IoT networks. It deserves to investigating whether other learning tasks can be efficiently implemented in IoT networks while keeping the privacy of agents preserved. Furthermore, it is also meaningful to devise learning algorithms that can tolerate faults caused by faulty nodes or unreliable communications.

References

[1] A. Agarwal, S. Agarwal, S. Assadi, and S. Khanna, "Learning with limited rounds of adaptivity: Coin tossing, multi-armed bandits, and ranking from pairwise comparisons," in *Proc. 30th Conf. Learn. Theory (COLT)*, 2017, pp. 39–75.

[2] J. Audibert, S. Bubeck, and R. Munos, "Best arm identification in multi-armed bandits," in *Proc. 23rd Conf. Learn. Theory (COLT)*, 2010, pp. 41–53.

[3] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, nos. 2–3, pp. 235–256, 2002.

[4] O. Avner and S. Mannor, "Multi-user lax communications: A multi-armed bandit approach," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, 2016, pp. 1–9.

[5] D. Basu, C. Dimitrakakis, and A. C. Y. Tossou, "Differential privacy for multi-armed bandits: What is it and what is its cost?" 2019. [Online]. Available: arxiv.abs/1905.12298.

[6] R. E. Bechhofer, "A single-sample multiple decision procedure for ranking means of normal populations with known variances," *Ann. Math. Stat.*, vol. 25, no. 1, pp. 16–39, 1954.

[7] R. Bonnefoi, L. Besson, C. Moy, E. Kaufmann, and J. Palicot, "Multi-armed bandit learning in IoT networks: Learning helps even in non-stationary settings," 2018. [Online]. Available: arxiv.abs/1807.00491.

[8] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Found. Trends Mach. Learn.*, vol. 5, no. 1, pp. 1–122, 2012.

[9] S. Bubeck, T. Wang, and N. Viswanathan, "Multiple identifications in multi-armed bandits," in *Proc. 30th Int. Conf. Mach. Learn. (ICML)*, 2013, pp. 258–265.

[10] Z. Cai and Z. He, "Trading private range counting over big IoT data," in *Proc. 39th IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Dallas, TX, USA, Jul. 2019, pp. 144–153.

[11] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 2, pp. 766–775, Apr.–Jun. 2020.

[12] Z. Cai, X. Zheng, and J. Yu, "A differential-private framework for urban traffic flows estimation via taxi companies," *IEEE Trans. Ind. Informat.*, vol. 15, no. 12, pp. 6492–6499, Apr. 2019.

[13] Z. Chaczko, S. Slehar, and T. Shnoudi, "Game-theory based cognitive radio policies for jamming and anti-jamming in the IoT," in *Proc. 12th Int. Symp. Med. Inf. Commun. Technol. (ISMICT)*, 2018, pp. 1–6.

[14] J. Chen, X. Chen, Q. Zhang, and Y. Zhou, "Adaptive multiple-arm identification," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 722–730.

[15] R. Ciucanu, P. Lafourcade, M. Lombard-Platet, and M. Soare, "Secure best arm identification in multi-armed bandits," in *Proc. 15th Int. Conf. Inf. Security Pract. Exp. (ISPEC)*, 2019, pp. 152–171.

[16] T. M. Cover and M. E. Hellman, "The two-armed-bandit problem with time-invariant finite memory," *IEEE Trans. Inf. Theory*, vol. IT-16, no. 2, pp. 185–195, Mar. 1970.

[17] B. Ding, J. Kulkarni, and S. Yekhanin, "Collecting telemetry data privately," in *Proc. 30th NeurIPS*, 2017, pp. 3571–3580.

[18] E. Even-Dar, S. Mannor, and Y. Mansour, "PAC bounds for multi-armed bandit and Markov decision processes," in *Proc. 15th Annu. Conf. Comput. Learn. Theory Comput. Learn. Theory (COLT)*, 2002, pp. 255–270.

[19] E. Even-Dar, S. Mannor, and Y. Mansour, "Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems," *J. Mach. Learn. Res.*, vol. 7, pp. 1079–1105, Jun. 2006.

[20] L. Fan and H. Jin, "A practical framework for privacy-preserving data analytics," in *Proc. 24th WWW*, 2015, pp. 311–321.

[21] B. Hajek, *Random Processes for Engineers*. Cambridge, U.K.: Cambridge Univ. Press, 2015.

[22] E. Hillel, Z. Karnin, T. Koren, R. Lempel, and O. Somekh, "Distributed exploration in multi-armed bandits," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, vol. 1, 2013, pp. 854–862.

[23] Q. Hu, S. Wang, X. Cheng, J. Zhang, and W. Lv, "Cost-efficient mobile crowdsensing with spatial-temporal awareness," *IEEE Trans. Mobile Comput.*, early access, Nov. 18, 2016, doi: 10.1109/TMC.2019.2953911.

[24] Q. Hu, S. Wang, P. Ma, X. Cheng, W. Lv, and R. Bie, "Quality control in crowdsourcing using sequential zero-determinant strategies," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 5, pp. 998–1009, May 2020.

[25] Q.-S. Hua *et al.*, "Faster parallel core maintenance algorithms in dynamic graphs," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 6, pp. 1287–1300, Jun. 2020.

[26] P. Kairouz, S. Oh, and P. Viswanath, "Extremal mechanisms for local differential privacy," in *Proc. 27th NeurIPS*, 2014, pp. 2879–2887.

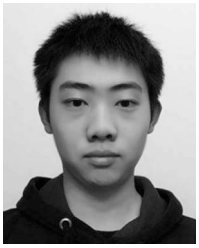
[27] F. Li, D. Yu, H. Yang, J. Yu, H. Karl, and X. Cheng, "Multi-armed-bandit-based spectrum scheduling algorithms in wireless networks: A survey," *IEEE Wireless Commun.*, vol. 27, no. 1, pp. 24–30, Feb. 2020.

[28] Y. Li and W. Xu, "PrivPy: General and scalable privacy-preserving data mining," in *Proc. 25th KDD*, 2019, pp. 1299–1307.

[29] A. Locatelli, M. Gutzeit, and A. Carpentier, "An optimal algorithm for the thresholding bandit problem," in *Proc. 33nd Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 1690–1698.
[30] M. Malekzadeh, D. Athanasakis, H. Haddadi, and B. Livshits, "Privacy-preserving bandits," 2019. [Online]. Available: arxiv.abs/1909.04421.
[31] N. Mishra and A. Thakurta, "(nearly) optimal differentially private stochastic multi-arm bandits," in *Proc. 31st Conf. Uncertainty Artif. Intell. (UAI)*, 2015, pp. 592–601.
[32] P. Semasinghe, S. Maghsudi, and E. Hossain, "Game theoretic mechanisms for resource management in massive wireless IoT systems," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 121–127, Feb. 2017.
[33] R. Shariff and O. Sheffet, "Differentially private contextual linear bandits," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 4301–4311.
[34] L. Su, M. Zubeldia, and N. A. Lynch, "Collaboratively learning the best option, using bounded memory," 2018. [Online]. Available: arxiv.abs/1802.08159.
[35] Y. Sun *et al.*, "Adaptive learning-based task offloading for vehicular edge computing systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3061–3074, Jan. 2019.
[36] Y. Sun, J. Song, S. Zhou, X. Guo, and Z. Niu, "Task replication for vehicular edge computing: A combinatorial multi-armed bandit based approach," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2018, pp. 1–7.
[37] D. Ta, K. Khawam, S. Lahoud, C. Adjih, and S. Martin, "LoRa-MAB: A flexible simulator for decentralized learning resource allocation in IoT networks," in *Proc. 12th IFIP Wireless Mobile Netw. Conf. (WMNC)*, 2019, pp. 55–62.
[38] C. Tao, Q. Zhang, and Y. Zhou, "Collaborative learning with limited interaction: Tight bounds for distributed exploration in multi-armed bandits," in *Proc. 60th IEEE Annu. Symp. Found. Comput. Sci. (FOCS)*, 2019, pp. 126–146.
[39] A. G. Thakurta and A. D. Smith, "(nearly) optimal algorithms for private online learning in full-information and bandit settings," in *Proc. 27th Annu. Conf. Neural Inf. Process. Syst.*, 2013, pp. 2733–2741.
[40] S. Vaswani, A. Mehrabian, A. Durand, and B. Kveton, "Old dog learns new tricks: Randomized UCB for bandit problems," 2019. [Online]. Available: arxiv.abs/1910.04928.
[41] W. Wang, L. Ying, and J. Zhang, "A minimax distortion view of differentially private query release," in *Proc. 49th ACSSC*, 2015, pp. 1046–1050.
[42] H. Yang, F. Li, D. Yu, Y. Zou, and J. Yu, "Reliable data storage in heterogeneous wireless sensor networks by jointly optimizing routing and storage node deployment," *Tsinghua Sci. Technol.*, vol. 26, no. 2, pp. 230–238, 2021.
[43] D. Yu *et al.*, "Implementing abstract MAC layer in dynamic networks," *IEEE Trans. Mobile Comput.*, early access, Feb. 4, 2020, doi: 10.1109/TMC.2020.2971599.
[44] J. Yu *et al.*, "Efficient link scheduling in wireless networks under rayleigh-fading and multiuser interference," *IEEE Trans. Wireless Commun.*, vol. 19, no. 8, pp. 5621–5634, Aug. 2020.
[45] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 5, pp. 968–979, Mar. 2020.
[46] X. Zheng, Z. Cai, and Y. Li, "Data linkage in smart Internet of Things systems: A consideration from a privacy perspective," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 55–61, Sep. 2018.
[47] Y. Zhou, X. Chen, and J. Li, "Optimal PAC multiple arm identification with applications to crowdsourcing," in *Proc. 31th Int. Conf. Mach. Learn. (ICML)*, 2014, pp. 217–225.
[48] D. Zois, "Sequential decision-making in healthcare IoT: Real-time health monitoring, treatments and interventions," in *Proc. IEEE 3rd World Forum Internet Things (WF-IoT)*, 2016, pp. 24–29.

Shuzhen Chen received the B.Sc. degree from the School of Computer Science and Technology, Shandong University, Qingdao, China, in 2019, where she is currently pursuing the postgraduation degree with the Department of Computer Science and Technology.

Her research interests include distributed computing and wireless and mobile security.



Youming Tao is currently pursuing the undergraduation degree with the Taishan College, Shandong University, Qingdao, China.

He currently focuses on fundamental problems in decentralized machine learning algorithm design, distributed computing, mechanism design, and privacy-preserving data analytics.

Dongxiao Yu (Member, IEEE) received the B.Sc. degree from the School of Mathematics, Shandong University, Qingdao, China, in 2006, and the Ph.D. degree from the Department of Computer Science, University of Hong Kong, Hong Kong, in 2014.

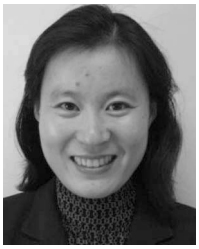
He became an Associate Professor with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, in 2016. He is currently a Professor with the School of Computer Science and Technology, Shandong University. His research interests include wireless networks, distributed computing, and graph algorithms.

Feng Li (Member, IEEE) received the B.S. degree in computer science from Shandong Normal University, Jinan, China, in 2007, the M.S. degree in computer science from Shandong University, Qingdao, China, in 2010, and the Ph.D. degree in computer science from Nanyang Technological University, Singapore, in 2015.

From 2014 to 2015, he worked as a Research Fellow with the National University of Singapore, Singapore. He then joined the School of Computer Science and Technology, Shandong University, where he is currently an Associate Professor. His research interests include distributed algorithms and systems, wireless networking, mobile computing, and Internet of Things.

Bei Gong received the B.S. degree from Shandong University, Qingdao, China, in 2005, and the Ph.D. degree from the Beijing University of Technology, Beijing, China, in 2012.

He has participated in six National invention patent and one monograph textbooks. In the past five years, he has published more than 30 papers in the first-class SCIEI and other international famous journals and top international conferences in relevant research fields. He has presided over eight national projects, such as the National Natural Science Foundation and six provincial and ministerial projects, such as the general science and technology program of Beijing Municipal Education Commission. His research interests include trusted computing, Internet-of-Things security, mobile Internet of Things, and mobile-edge computing.



Xiuzhen Cheng (Fellow, IEEE) received the M.S. and Ph.D. degrees in computer science from the University of Minnesota, Minneapolis, MN, USA, in 2000 and 2002, respectively.

She is a Professor with the School of Computer Science and Technology, Shandong University, Qingdao, China. Her current research interests include cyber–physical systems, wireless and mobile computing, sensor networking, wireless and mobile security, and algorithm design and analysis.

Prof. Cheng received the National Science Foundation (NSF) CAREER Award in 2004. She has served on the editorial boards of several technical journals and the technical program committees of various professional conferences/workshops. She also has chaired several international conferences. She worked as a Program Director for the U.S. NSF from April to October in 2006 (full time), and from April 2008 to May 2010 (part time). She is a Member of ACM.