# Differentially-Private and Trustworthy Online Social Multimedia Big Data Retrieval in Edge Computing

Pan Zhou, *Member, IEEE*, Kehao Wang, *Member, IEEE*, Jie Xu, *Member, IEEE*, and Dapeng Wu, *Fellow, IEEE*

***Abstract*—The explosive growth of multimedia contents (MCs) in today's mobile social networks has pushed edge computing to face severe security and online big data-processing problems. On the one hand, the edge nodes (ENs) should help mobile users find, cache, and share MCs in the presence of an ever-increasing scale of multimedia big data. On the other hand, how to provide secure MC retrieval schemes to exclude *dishonest-and-malicious* untrusted ENs and to prevent privacy breaches from *honest-but-curious* ENs and users is a challenging issue. To tackle these problems, we study the privacy-preserving and trustworthy MCs retrieval system to make personalized MC recommendations from ENs to users with big data support. In our framework, each EN is modeled as a distributed context-aware online learner. ENs collaborate to learn users' preferences based on their contexts and previous behaviors and social intimacy. To support big data analytics, we establish an MC-cluster tree from top to the bottom to handle the dynamically varying cached MC datasets. A differentially private algorithm is proposed to preserve the data privacy among honest-but-curious ENs and users. To guarantee trustworthy edge computing, a trust evaluation mechanism is designed to evaluate the reliability of ENs. We further consider the structure of edge networks to improve the performance of our algorithm. Experimental results validate that our new framework can support increasing multimedia big datasets while striking a balance among privacy-preserving level, Trustworthy level, and caching MC prediction accuracy.**

***Index Terms*—Edge computing, differential privacy, online learning, big data, trustworthiness, social mutlimedia.**

## I. Introduction

THE proliferation of online multimedia services has made it possible for mobile users to consume, entertain and share personalized multimedia contents with other similar behavioral users through the mobile social networks (MSNs). In 2018, YouTube reports there are more than 300 hours' videos upload every minute, 10,113 YouTube videos generated over 1 billion views, and about 1 billion mobile YouTube video views per day [1]. So it has become increasingly difficult for the overloaded backbone network to discover (mobile) users' preferred multimedia contents (MCs) in those big datasets. In addition, the multimedia service providers (e.g., YouTube, YouKu, Tencent, etc) are usually located at remote sites, which brings large fetch latency that degrades users' quality of experience (QoE). To resolve these problems, mobile edge computing (MEC) [2] has appeared as a promising paradigm in the domain of Internet of Things (IoT), where MCs are cached on edge nodes (ENs), who are deployed in the proximity of users. As a result, users could retrieve preferred MCs from the nearest ENs to significantly reduce fetch latency. However, the massive data must be harnessed for MC popularity estimation and retrieval, where big data analytics schemes, in particular, machine learning mechanisms for large scale multi-dimensional data from various resources are indispensable in MEC [3]. Therefore, an MC retrieval system which can extract users' personalized information and recommend suitable MCs from ENs to users is desirable.

Unfortunately, MEC also brings novel challenges on big data analytics: heterogeneity of users, lack of historical visited records and analytical algorithms to support dynamic increasing social MCs. First, users have different preferences and goals. Thus, it is important but difficult to make personalized MC recommendations. Second, each user might only be interested in a small amount of MCs, so the classic approaches based on users' own historical visited results are infeasible. Third, due to the data volume of MCs cached on ENs, it becomes increasingly difficult for ENs to discover useful information to users. Furthermore, the ENs have to support caching new social MCs and replacing less visited ones. Considering the ever-increasing scale of MCs from MSNs, the computational and bandwidth constraints prevent a stand-alone EN from having access to all MCs. Hence, to promote system performance, ENs might tend to have low-rate user information sharing with other ENs, and they cooperate to discover relevant MCs and predict users' behaviors.

Moreover, the recommendation of suitable MCs to match users' preference (context) would face security and privacy issues. On the one hand, historical feedback information from different types of users and the similarities between users' context and MCs, if published, can be extremely beneficial. But they are usually sensitive and private. Hence, system designers must strike a balance between the beneficial uses of data and the privacy. Specially, to support big data analytics, a scalable

Manuscript received July 21, 2018; revised October 27, 2018; accepted December 3, 2018. Date of publication December 10, 2018; date of current version February 21, 2019. This work was supported in part by the National Science Foundation of China under Grants 61672395, 61401169. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Zhou Su. *(Corresponding author: Kehao Wang.)*

P. Zhou is with the School of Electrical Information and Communication Engineering, Huazhong University of Science & Technology, Wuhan 430074, China (e-mail: panzhou@hust.edu.cn).

K. Wang is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 USA and the Department of Information Engineering, Wuhan University of Technology, Wuhan 430074, China (e-mail: kehao.wang@whut.edu.cn).

J. Xu is with the Department of Electrical and Computer Engineering, University of Miami, Coral Gables, FL 33146 USA (e-mail: jiexu@miami.edu).

D. Wu is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: wu@ece.ufl.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2018.2885509

and widely applicable privacy-preserving scheme is required in MEC. Differential privacy (DP) is such a promising notion that impacts little on the prediction accuracy especially when applied to *large-scale* datasets [4]. Furthermore, DP does not require the background knowledge of adversaries [5].

On the other hand, ENs exchange information with each other and might bypass a central trusted system. This makes security of the distributed system hackable and untrustworthy. ENs can be submerged and controlled by malicious attackers to provide inferior MC services, untrustworthy MC sent by distrustful ENs has the potential to become the most harmful information within an edge network. For example, an EN might provide inferior-quality multimedia streaming service to lighten its load, thereby jeopardizing the QoE of users from other collaborative ENs. Trust is the subjective feeling of an individual to other strange ENs, which is based on the service experience [6]. Hence, users from a trusted EN need to balance between the benefits received in the reciprocal activities and the risks related to communications with strange ENs. In this context, it is important to figure out the ENs that are trustworthy with each other of other ENs in order to make social decision about how to disclose and share personal private multimedia information. An EN with high trustworthiness is considered to be much reliable and secure, and vice versa. Therefore, introducing trust scheme is needed to guarantee the security of edge networks such that malicious ENs will be removed.

To address the aforementioned challenges, we propose a tree-based privacy-preserving and trustworthy distributed MC retrieval system, where each EN is modelled as a contextual online learner. The set of connected ENs collaborate to make personalized predictions over edge networks by deploying differential privacy and trust mechanisms. Each EN experiences inflows of users to its caching server, where MCs are cached in ENs from MC service providers. Each user arrives with a *context* containing its background information (e.g., age, gender, locations, social profile and query conditions, etc). The system learns its preference based on the historical records, and then it selects a personalized MC to push to the user. The user will give a reward to the system according to his/her satisfaction degree. Context-awareness is a key component in IoT and is also indispensable for selecting an MC in MEC, since the feedback to an MC is highly correlated to user context [7] [8]. Different users might obtain different QoEs (rewards) from the same MC, even the same user will give different feedback when its context changes. Our proposed method derives from contextual multi-armed bandit (CMAB) [9]. CMAB keeps an variable that measures the estimated performance and uncertain reward of each arm, i.e., MC. The performance loss is measured in terms of *regret*, where a *sublinear* regret indicates converging to the optimal policy.

To support big data analysis, we adopt an MC-cluster tree structure, which expands over time to cluster MCs from top to the bottom rather than from bottom to up [23]. Our *top-down* structure supports the retrieval on dynamic changing and increasing large-scale cached datasets, since ENs perform recommendations on an MC-cluster level instead of a single MC level [23]. Therefore, the analyzed component is scalable for ENs and thus makes big data analysis feasible. In addition, we consider the social relations between users, which are also significant information for social multimedia retrieval and decision making [10]. We adopt an adaptive context partition method. This guarantees that ENs can learn the preference of current users through analyzing the data of the most similar previously arrived users, who have a high social intimacy with the current users. Therefore, it provides up-to-date personalized MC recommendations. It is worth mentioning that there is another research line focusing on the machine learning for the optimization of the cached contents (MC datasets) in ENs [11], which is from a different perspective compared to us since our algorithm supports changing and increasing cached contents. Therefore, our algorithm is compatible with any cache content replacement mechanism in MEC.

Moreover, in reality, there are usually thousands of heterogeneous ENs inter-connected together. Hence, it is of vital importance to investigate the online MC retrieval performance of our proposed distributed CMAB algorithm over general edge network graphs. Obviously, a simple deployment of the algorithm over a large number of ENs by enabling collaborations among them would result in massive computing burden. To tackle this challenge, we explore the network structure, e.g., the one-hop neighbors, which is capable to observe further and extensive learning performance improvement.

In terms of privacy, we adopt DP, which only requires that the query results given by the database curator (i.e., users and ENs) are insensible to the change of a single data entry (MC). In other words, if users' data is published satisfying DP, when attackers query the cache database of ENs by using some data analysis tools, the query results will be almost the same whether a single user's record is in or not. Then, the private information of users cannot be inferred by attackers. We store all ENs' historical records (MCs, contexts, rewards) provided by a trusted third party. An EN will send the historical records deploying a Laplace mechanism (LM) [21], when there is a request for shared information from its one-hop neighbors. Different from traditional methods which add noise to each record and will result in large distortions, we adopt a binary tree-based noise aggregation mechanism to reduce the performance loss and guarantee DP simultaneously. In addition, ENs release learning results of users' preference by deploying an Exponential mechanism (EM) [22]. We prove that our proposal can guarantee DP of ENs' shared information and users' sensitive personalized contexts. Moreover, to evaluate the trustworthiness of ENs, a trust evaluation scheme based on users' long term satisfaction is designed to assess the reliability of each EN. Our main contributions are as follows:

1) To our best knowledge, we propose a first privacy-preserving and trustworthy context-aware online learning algorithm with big data support in MEC, where ENs collaborate to learn and predict users' behaviors.
2) An adaptive algorithm considering the structure of edge networks is proposed to reduce performance loss extensively.
3) Our proposal makes personalized recommendations by partitioning users' context space adaptively, which

can support dynamically changing MCs datasets. Experimental results show that our proposal outperforms other methods and can support increasing big datasets in practice.

4) By defining rigorous attack models and integrating EM, LM and trust schemes into the system, our algorithms can guarantee the convergence to the optimal policy, small privacy loss and high trustworthy level in the long run, simultaneously.

The paper is organized as follows. Related works are in Section II. Section III formalize the problem. Section IV and V present our algorithms and theoretical analysis. Experiments results are in Section VI. Section VII concludes the paper.

## II. Related Works

Our work belongs to the family of *social recommender systems* for the MCs retrieval in the IoT applications [17]–[19]. Some researchers have studied on the context-aware social MC retrieval method [12]–[14] in MEC. In [14], the authors propose a social-aware content-centric cooperative MC retrieval service in MEC. However, it does not support big data analytics. Another line of existing works have focused on social IoTs [15], [16], where different problems have been studied, e.g., information diffusion [15], objects discovery [16], but few of them were designed for MC retrieval. For example, [16] proposes a framework to discovery and reputation assessment of services and objects for smart things.

The *Multi-armed Bandit (MAB)* approach has been shown to be an excellent solution for the exploration-exploitation tradeoff in IoT environment, which can achieve optimal learning performance in a finite time duration [29]. For example, Buccapatnam *et al.* [26] propose an online learning algorithm based on UCB1 in [29], where agents collaborate to promote prediction accuracy on small scale and fixed-size datasets. In [17], *LinUCB* algorithm providing sublinear regret was applied to recommend IoT services, which cannot support expanding large-scale dataset or consider user context.

Our proposed social MC retrieval method by using users' contexts derives from contextual multi-armed bandit (CMAB) [9]. The idea is to utilized the observable user context to guide the selection of the arm (i.e., MC) with the highest empirical reward at each time to reach the optimal policy over time. Thus, personalized recommendation is feasible. Previous works [23], [27], [28] have studied CMAB. Cem *et al.* [27], [28] propose an algorithm to make personalized predictions for MC set with fixed size. However, a static context partition method is adopted, which will result in tremendous computational complexity when the dataset is in large scale. Thus, it is not practical for big data analysis. So Song *et al.* propose a CMAB algorithm for a *single* agent with an adaptive context partition method [23]. But it builds the MC-cluster tree from *bottom to the top*, which constrains the number of MCs. Hence, it also does not support big data analysis in MEC.

Furthermore, none of these works have considered social relationships. Nowadays, social relationships can have a significant impact on human decision making process [24] in IoT on multimedia-sharing applications. In addition, the one agent MC retrieval system [23] can not handle big data from decentralized and multimedia sources. Collaborative machine learning among ENs is found to be necessary [25], where our distributed CMAB algorithm is such an effective solution.

A comprehensive survey for security and privacy issues of multimedia services in MSNs is available in [30]. In terms of the *privacy-preserving issue* for MC retrieval system, anonymity is a widely applied technology [31]. However, most anonymity algorithms appear to destroy the utility of high dimensional big data [32]. In addition, though being anonymous, users can be re-identification when adversaries collude with each other or have access to auxiliary information [37]. Furthermore, existing works usually rely on cryptography [33] to protect private data from outsiders. Though these methods can guarantee security, they usually have high computation and communication costs, which is prohibitive for big data analysis. DP has been adopted in machine learning in MEC by several studies [34]–[36] but with no big data support. In [38], DP is used in social graph for recommendation. While [38] focuses on the privacy of user preferences and social relations between ENs in the social network, our study focuses on the privacy of personal information in user context and privacy of ENs' MC repository. Moreover, we further consider the impact of the network structure on the performance.

Trust mechanism design in MEC is investigated in recent time. For example, [39]–[41] focus only on designing effective trust schemes for ENs, while [6] consider the optimal users and ENs content matching. However, all these works [6], [39]–[41] do not consider big data analytics and the privacy issue for social MC retrieval. Plus, trust mechanism for social relationship among users in online social networks have been extensively studied from the respective of each user, e.g., [42]–[44]. In contrast, our work assumes that all socially intimated users in the MSNs are trustworthy to each other, but different edge nodes as MC service providers might not be.

## III. Problem Formulation

### A. System Model

Our proposed tree-based privacy-preserving and trustworthy contextual distributed online learning MC retrieval system is composed of the following basic components: *ENs*, *MCs*, *users* and *trusted third party*, which is shown in Fig. 1. Note that the preliminaries about DP such as the Exponential mechanism (EM) Laplace mechanism (LM) have a quick reference in [51].

*ENs:* We model the ENs in the edge network as nodes in a graph. A set of $|A|$ ENs $A = \{1, 2, \ldots, |A|\}$ are connected by a fixed undirected network represented by graph $G = (A, E)$ where $E$ denotes the set of edges in the graph. The connection between ENs is represented by an adjacency matrix $e(i, j)_{i,j \in A} \in \{0, 1\}$. If $e(i, j) = 1, e(i, j) \in E$, $i$ is $j$'s one-hop neighbor (OHN) and $j$ is $i$'s OHN. As shown in Fig. 1, we have $e(1, 2) = e(2, a) = 1$. An EN provides MC services in practice. For example, a video EN could share different video web applications to provide more suitable videos for their users. Actually, users can also be MC service providers. Without loss of generality (wlog), we denote all MC service providers simply as ENs in $G$.

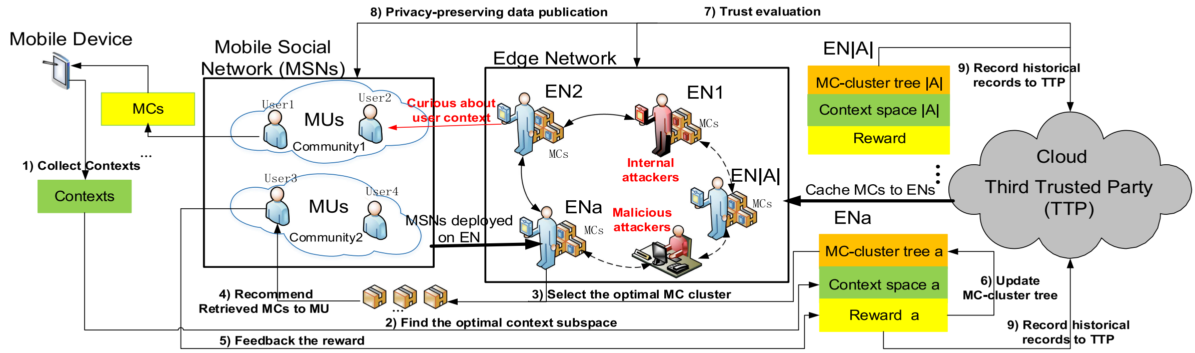


Fig. 1. Workflow for privacy-preserving and trustworthy distributed online social MC retrieval system.

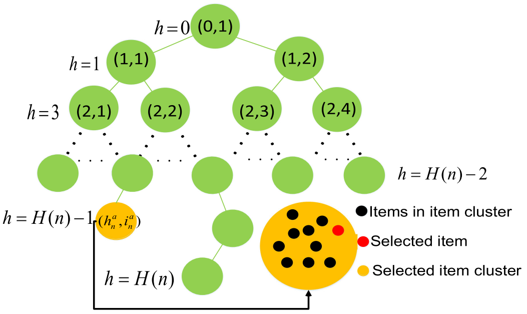


Fig. 2. MC-cluster tree.

*MCs:* Each EN possesses a set of MCs in its repository. Each MC is featured by a $d_i$-dimensional vector, where each entry of the vector represents a feature of the MC (e.g., image size, video qualtiy, title, MC category, etc). Some of these features are accessible directly while others require complex algorithms to do feature extraction works. The MC as a feature vector is drawn from a $d_i$-dimensional MC set $I^a = \{i_1^a, i_2^a, i_3^a, \dots\}$ in the repository of EN $a$.

*MC-cluster Tree Structure:* To support big data analytics, we build a binary *MC-cluster tree* structure for MC space $\mathcal{I}$, denoted by $\mathcal{T}^I$, for each EN in the network, to reduce the scale of analyzed MCs at each time slot. The MC-cluster tree $\mathcal{T}^I$ is an infinite binary tree. The $i$th node at depth $h$ from the root in $\mathcal{T}^I$ is denoted by $(h, i)$. The index $i$ of nodes at depth $h$ is restricted by $1 \leq i \leq 2^h$. Intuitively, the left child and right child of $(h, i)$ are $(h+1, 2i-1)$ and $(h+1, 2i+1)$, respectively. Each node in $\mathcal{T}^I$ represents an MC cluster in $\mathcal{I}$. For example, in Fig. 2, the root node (0,1) covers the whole MC space $I$, while its left child (1,1) and right child (1,2) cover two subspaces with same size (measured by dissimilarity $D_i$) of $I$. We define the depth of tree $\mathcal{T}^I$ as $H = \max_{(h,i) \in \mathcal{T}^I} h$. The region associated with cluster $(h, i)$, denoted by $\mathcal{R}_{h,i}^I$ satisfies the constraints: $\forall h \geq 0, 1 \leq i, i' \leq 2^h, \mathcal{R}_{h,i}^I \cap \mathcal{R}_{h,i'}^I = \phi, \mathcal{R}_{h,i}^I = \mathcal{R}_{h+1,2i-1}^I \cup \mathcal{R}_{h,2i+1}^I$.

One benefit of the MC-cluster tree structure is that it recommends MCs on a cluster level instead of a single MC level. This can greatly reduce the time complexity. Compared with the mechanism that recommends MCs on a single MC level, our proposal reduces the cost of finding suitable MCs from $\mathcal{O}(n)$ to $\mathcal{O}(\log n)$.

*Users:* Users arrive sequentially at each EN with its background context information. The context of each user $u$ is featured by a context vector $c$, which is drawn from a $d_c$-dimensional context space $C$. Each dimension of the user context vector describes one of its feature at the moment. For example, a context could be a user's camera type (e.g., Canon, Iphone), location, time, user's profile (e.g., age, gender, nationality, profession, etc). Contexts can be sensed, collected and quantified by deploying middleware applications on users.

*Adaptive Context Partition:* Wlog, each subspace of $C$ is modelled as a hypercube of $d_c$ dimensions with side length $m^{-l}$, where $l$ is the level of context subspace. The hypercube is used as a partitioning approach to cover the context subspace. We define the depth of context space as $L = \max_{S \in C} l_S$, where $S$ is the subspace of $C$ and $l_S$ is the level of $S$. Considering the fact that online user data is dynamically increasing, we set a threshold for each context subspace. Once the number of context arrivals in a context subspace exceeds the threshold, this subspace will be further partitioned. When a subspace of context is partitioned, each side of the hypercube is divided into $m$ parts uniformly (detailed in next section). Wlog, we normalize $C = [0, 1]^{d_C}$ and $m = 2$ to simplify our theoretical analysis.

*Trusted Third Party:* To prevent hackable and vulnerable behaviors to the system, we introduce the trusted third party as shown in Fig. 1. In the MEC, the trusted third party could be the master cloud in the hierarchical architecture. All historically visited data (i.e., users' contexts, rewards, recommended MCs) of ENs are reported and collected by the trusted third party. It is only responsible for the privacy-preserving publication of shared information and trustworthy evaluations among ENs. It is assumed to be fully trusted, the privacy leakage of which is not the concern in this paper.

## B. System Workflow

Consider a time-slotted system, where $n \in \{1, 2, \dots\}$ is the running round of the online decision-making algorithm and at each round a user arrives with his/her current context. Let $t_n$ be absolute time stamp of the beginning of round $n$. Then, using this absolute time stamp and the index of the EN, we can uniquely identify each MC, user, context and reward as follows. Let $u^a(t_n)$ and $c^a(t_n)$ be the user and context which arrive at EN $a \in A$ at round $n$. And let $r^a(t_n)$ and $i^a(t_n)$ be the corresponding received reward and retrieved MC. The workflow of the system is mainly composed of nine steps and we describe them with the illustrations in Fig. 1 as follows.

First, *1)* Users upload the MCs captured by their smart devices to the media sharing websites, e.g., User1 in Fig. 1, which are finally stored in service providers or cached in ENs. In this process, contents' features (e.g., times and location it was taken, camera type) are extracted by existing preprocessing methods. Meanwhile, a user $u(t_n)$, in Fig. 1 depicted as User3, arrives at the system and searches for its interested MCs. This query action will generate context and social information. We denote the context arriving at round $n$ by $c(t_n)$, which is inputed into the context space in EN $a$. *2)* Then, the social information of user $u(t_n)$ is input to the MC-cluster tree while the optimal context subspace relevant to $c(t_n)$ is found. The context space will be expanded to smaller subspaces when an existing subspace contains too much contexts. *3)* Then, EN $a$ learns the preference of its user $u^a(t_n)$ based on the historical feedbacks from all users who share similar context with user $u^a(t_n)$. The justification is based on Assumption 1 in Subsection III.C. An optimal MC-cluster as a node in the MC-cluster trees at round $n$ is predicted and selected based on these historical records (detailed in Section IV).

Next, *4)* An MC in the optimal MC-cluster is then retrieved and recommended to the user. Because an individual can be identified in the real world, the EN recommends the MC using an EM scheme instead of recommending the MC directly in the cluster, which is predicted to receive the highest reward. Then, a true reward $r^a(t_n)$ is received by the trusted third party based on the behaviors/feedbacks of user $u^a(t_n)$. And all the records are stored on the cloud provided by the trusted third party as in Fig. 1. *5)* The system receives a reward $r(t_n)$ normalized to $[0, 1]$ from $u(t_n)$ based on the user's browsing behavior of the MC and whether or not it shares the MCs with others (e.g., the User3 is in the same community 2 with User4 but not with User1 and User2 in Fig. 1). Then, the user feedbacks the reward to the MC-cluster tree at the context subspace that $c(t_n)$ located.

Then, *6)* Update the estimated upper bound reward of each MC cluster in MC-cluster tree and expand new nodes when the existing node contains too many MCs. *7)* For the situation that EN needs other ENs for help, EN $a$ will send a request to the trusted third party for sharing MCs from its OHNs, the trusted third party will evaluate their trustworthy levels. If any connected ENs have recommended an MC to users (indexed by contexts) that shows bad reputation (low trustworthy level to according to the trust value $TS^a_{h_k,i_k}(n-1)$ defined in (3)) in the prior round, the trusted third party needs to exclude the untrusted ENs at this time.

Afterward, *8)* Because attackers can infer the repositories from the released information, and privacy-preserving data publication is necessary. Each EN tends to protect the privacy of its repository from other ENs (competitors), because some MCs can bring high benefit while others only bring budget deficit. When an EN requests shared information from its OHNs, the trusted third party will add noises to this information (e.g., add Laplace noise and send an averaged value, etc.) and send back to the EN. Thus, the shared information is released by deploying a LM scheme to guarantee the privacy of ENs' repositories. Lastly, *9)* All the ENs store their historical visited records to trusted third party after the round $n$.

### C. Attack Model

We consider three types of adversaries for the edge network: internal honest ENs who are curious about users' contexts, internal honest ENs who are curious about other ENs' MCs (repositories), and malicious ENs. The internal attack from the trusted third party (e.g., colluding with other companies to sell users' and ENs' personal information) and users is beyond the scope of our discussion. The malicious ENs who always provide low quality MC contents to users, which would result in bad reputation and are untrustworthy. The case malicious users who always feedback unreasonable rewards to ruin the reputation of an EN and hired users who give high rewards (QoE) to promote the reputation of an EN are not the concern in this work. In our model, there are two kinds of adversaries:

*1) Honest ENs but Curious about users' contexts:* Malicious attackers try to identify a user in the real world. A malicious attacker chooses a historical record to attack and runs a data analysis program by which it sends similar queries as a curious EN to the trusted third party to obtain the user's reward of an MC to infer the user's sensitive contexts. For example, if a top hotel in the capital of China is predicted as an MC in which a user is interested, attackers could infer that this user probably has a high salary and is located in Beijing at the moment.

*2) Honest ENs but Curious about other ENs' MCs:* Agents are honest to the trusted third party about their information and will recommend MCs according to the designed algorithm. And, we assume that ENs will not tend to expose any user context. Now, we consider a simple case when we do not use any privacy-preserving mechanism (PPM) in our algorithm. At each round $n$ in the policy adaptation phase, for any EN $a \in A$ which requests shared information, supposing $i^a(t_n) \in (h^a_n, i^a_n)$ given $c^a(t_n) \in C_n$, EN $a$ will send a query to ask the trusted third party to tell EN $a$ the number of MC arrivals in which the context arrivals are in same certain context subspace with $C_n$ while the recommended MCs belong to the same MC cluster with $(h^a_n, i^a_n)$. Plus, the sum rewards of these users arrivals will be exposed to EN $a$. Using these information, EN $a$ could infer rewards of the MCs from its competitors.

*3) Malicious ENs (or users):* Malicious attackers on an EN will not follow the designed algorithm and provide low quality MCs to users during a long time observation. These ENs might recommend some of the high quality MCs in short time to attract users, but most of the time they offer very bad quality MCs (e.g., meaningless image, low quality video or ads). So a long term trust evaluation mechanism should be deployed in the trusted third party to remove malicious unreliable ENs over time. For the case users act as the MC SP (e.g., User1 in Fig. 3), the trusted third party will also need to assess their trustworthy level. As we mentioned above, wlog, we simply treat them as ENs.

### D. Performance Metric

For clarification, we define several important concepts.

*Dissimilarity and Diameter:* The dissimilarity over MC space $I$ is assumed to be a non-negative mapping $D_i\colon I^2 \to \mathbb{R}$ where $D_i(i, i) = 0$ and $D_i(i, i') \geq 0$ for any $i, i' \in I$. Note that the metric $\boldsymbol{D_i}$ can be any norm in the Euclidean space $I$. Similarly,

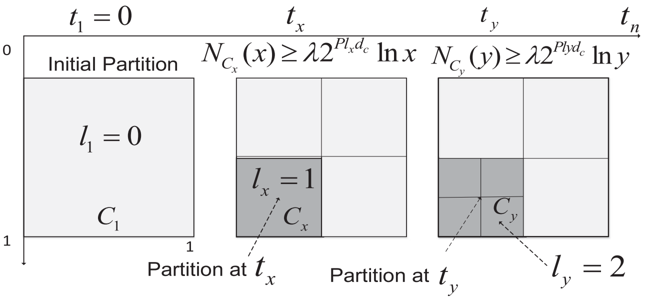


Fig. 3. Context space partition.

the user context space $C$ is assumed to be equipped with a dissimilarity function $D_c$ such that $D_c(c,c) = 0$ and $D_c(c,c') \geq 0$ for any $c, c' \in C$. For $\forall i, i' \in I$ and any subset $R \in I$, we denote by $\mathcal{DIAM}_R^I = \sup_{i,i' \in R} D_i(i,i')$ the *diameter* of subset $R$.

*Reward:* For any EN $a \in A$, at round $n$, a reward $r^a(t_n)$ is received from an unknown distribution depended on the arriving context and MC. Feedbacks (rewards) are given from the behavior of users interacting with the EN. Take the reward of an online media content sharing system (e.g., Flickr) as an example, if the user clicks the content or adds it to its favorite list, $r^a(t_n) = 1$, otherwise $r^a(t_n) = 0$. The distribution of reward is usually assumed to be independent and identically distributed (i.i.d.), since we can view the rewards as random variable drawn from an unknown distribution depended only on context and MC. Although the realistic situations of the reward might not be i.i.d., we can use the simple trick of using two i.i.d. processes to bound a non-i.i.d. process [27] while all the performance bounds still hold.

As mentioned in the system model, users will propagate MCs through social networks by sharing it with their online friends or uploading it to a community's forum stored in EN $a, \forall a \in A$. For example, User3 in Fig. 1 is a friend with User4 in the Community2. Each user keeps a friend list and each community keeps a list of its members. We specifically formalize the impact of social relations as follows.

*Friendship Factor:* Users keep lists of friends dynamically updated by themselves on their mobile devices. The friendship factor between user $u$ and user $u'$, denoted by $\mathcal{F}_{u,u'}$ at EN $a$, is computed as $\mathcal{F}_{u,u'}^a = \frac{F_u^a \cap F_{u'}^a}{F_u^a \cup F_{u'}^a}$, where $F_u^a$ denotes the friends of user *u* at EN $a$. Considering the situation when user *u* has no friend, it is included in its own friend list. Friendship factor $\mathcal{F}_{u,u'}^a$ reflects the ratio of common friends between user $u$ and $u'$ to the total number of friends possessed by $u$ and $u'$.

*Interest Factor:* Each user keeps a list of its interested communities on their devices. The interest factor between user $u$ and user $u'$ at EN $a$, denoted by $\mathcal{I}_{u,u'}^a$, is computed as: $\mathcal{I}_{u,u'}^a = \frac{\Gamma_u^a \cap \Gamma_{u'}^a}{\Gamma_u^a \cup \Gamma_{u'}^a}$. Interest factor $\mathcal{I}_{u,u'}^a$ reflects the ratio of common communities in which user $u$ and $u'$ are interested to the total number of communities $u$ and $u'$ paying attention to at EN $a$. Considering privacy issues, whenever user $u$ and $u'$ have a direct interaction, they only exchange their friends and interested communities in common.

*Social Intimacy:* We denote by $\omega_{u,u'}^a$ the social intimacy between users $u$ and $u'$ based on the combined effect of friendship factor and interest factor at EN $a$, which is computed as:

$$\omega_{u,u'}^a = \eta \mathcal{F}_{u,u'}^a + (1-\eta)\mathcal{I}_{u,u'}^a, \tag{1}$$

where $\eta$ reflects the impact of common friends between two users on social intimacy.

*Suboptimal Cluster:* Let $\mu_{i,c}$ be the expected reward of MC $i$ when the user context is $c$, we define $\mu_c^* = \max_{i \in I} \mu_{i,c}$ as the global optimal expected reward for the user who arrives with context $c$ and this value is known only in hindsight. In the MC-cluster tree, the *suboptimal factor* of cluster $(h,i)$, which is denoted by $\Delta_{h,i}$, is calculated as: $\Delta_{h,i} = \max_{c \in C} \mu_c^* - \max_{x \in (h,i)} \mu_{x,c}$. If there exists an MC $i* \in (h*, i*)$ and context $c \in C$ in the historical results, such that $\mu_c^* - \max_{x \in (h*,i*)} \mu_{x,c} = 0$, we define $(h*, i*)$ as an*optimal node/cluster*, otherwise, it is a *suboptimal node/cluster*. In a context-aware system, the reward of an MC with a similar personalized user's context is assumed to be similar [46]. Similar with [23], this assumption can be formalized as the following Lipschitz conditions.

*Assumption 1:* (Lipschitz conditions)

(a) (*Lipschitz condition for contexts*) For any MC $i \in I$ and context $c, c'$ respectively, there exists a positive Lipschitz constant $L_c$ such that $\mid \mu_{i,c} - \mu_{i,c'} \mid \leq L_c D_c(c,c')$.

(b) (*Weak Lipschitz condition for MCs*) For any MC $i, i' \in I$ and context $c \in C$, it is assumed that $\mid \mu_{i,c} - \mu_{i',c} \mid \leq \max\{D_i(i,i'), \mu_c^* - \mu_{i',c}\}$.

This assumption is quite reasonable in practice. For example, suppose that $d_c = 2$ and each context vector is featured by users' age and income level. A user of 40-year-old earning \$ 27000 per year and a 38-year-old user earning \$ 27800 per year are possible to give a similar low reward to luxuries.

*Regret as the Performance Metric:* Let $\mu^a(t_n)$ be the expected reward of $i^a(t_n)$ without using EM. And $\mu_E^a(t_n)$ is the expected reward of $i^a(t_n)$ when EM is applied. To measure the loss of accuracy from not selecting the optimal MC, for any $a \in A$, we calculate the one step *regret* at time stamp $t_n$, denoted by $\Delta_{t_n} = \mu_{c^a(t_n)}^* - r^a(t_n)$. Then, the expectation of cumulative regret (CR) over $G(A,E)$ is defined as:

$$\mathbb{E}R(n) = \mathbb{E}\left[\sum_{a \in A}\sum_{k=1}^{n} \Delta_{t_k}\right] = \sum_{a \in A}\sum_{k=1}^{n}\left(\mu_{c^a(t_n)}^* - \mu_E^a(t_k)\right),$$

where $\mu_{c^a(t_n)}^*$ is the global optimal expected reward for user $u^a(t_n)$ arriving with context $c^a(t_n)$ at EN $a$ at round $n$. If a recommender policy can achieve sublinear regret (i.e., $\mathbb{E}R(n) = O(n^\theta), \theta < 1$), we can derive that the policy converges to the optimal policy (i.e., $\lim_{n \to \infty} \frac{R(n)}{n} \to 0$). The object in this paper is to minimize the CR while protect privacy of users and ENs.

Due to the exploration process of MC-cluster tree (detailed in Section IV), the number of MCs in each node (i.e., cluster) is limited. Thus, the MCs in each node have an upper dissimilarity bound. In addition, since the MCs in the MC space $I$ are by nature discretely distributed, there is a lower dissimilarity bound for MCs in each node. We formalize this as the following assumption.

**Algorithm 1:** T-DPTDO

```
1  Input: $d_c, d_i, L_c, \beta_s, n, \gamma_s$, network structure $G(A, E)$;
2  Initialization: $t = 0, n = 1, H(n) = 0, \mathcal{T}_a^I(n) = \{(0,1), (1,1), (1,2)\}, B_{1,1}^a(n) = B_{1,2}^a(n) = infinity$;
3  Auxiliary procedure: $TNA$;
4  for $n = 1, 2, \cdots$ do
5      user $u^a(t_n)$ arrives at EN $a$ with context $c^a(t_n)$ at time stamp $t_n$; find the context subspace $C_n$, which $c^a(t_n)$ belongs to; Caculate the trust $TS_{h_k,i_k}^a(n-1)$ according to (3) in last round and obtain the untrusted set $\tilde{\mathrm{N}}_a(\tau)$;
6      if $TS_{h_k,i_k}^a(n-1) < TSL$ then
7          $\tilde{\mu}_{PC}^a(n) \leftarrow TNA(C_n, a, (h_n, i_n))$;
8          $\tilde{\mu}_{h_n^a,i_n^a}^a(n) = \frac{\tilde{\mu}_{PC}^a(n)}{M_{h_n^a,i_n^a}^a(n)}$;
9      else
10         $\tilde{\mu}_{h_n^a,i_n^a}^a(n) = \frac{1}{M_{h_n^a,i_n^a}^a(n)} \sum_{\tau \le n} \mathbb{I}_{\mathcal{E}_{h,i}^a(\tau)} r^a(t_\tau)$;
11     for any leaf node $(h, i) \in \mathcal{T}_a^I(n)$ do
12         Update $B_{h,i}^a(n)$;
13     for any leaf node $(h, i) \in \mathcal{T}_a^I(n)$ do
14         $\mathbb{P}[(h_n^a, i_n^a) = (h, i)] = \frac{\exp \frac{\epsilon' B_{h,i}^a(n)}{2\Delta B}}{\sum_{(h,i)\in\mathcal{T}_n^I} \exp\left(\frac{\epsilon' B_{h,i}^a(n)}{2\Delta B}\right)}$;
           /* $\varepsilon' = \varepsilon / n$ */
15     Select a leaf node $(h_n^a, i_n^a) \in \mathcal{T}_a^I(n)$ based on the above computed probability distribution, and randomly recommend an MC $i^a(t_n) \in (h_n^a, i_n^a)$ for user $u^a(t_n)$.
16     A reward $r^a(t_n)$ is received by EN $a$;
17     if $M_{h_n^a,i_n^a}^a(n) \ge \delta_{h_n^a} = \frac{c \ln n}{\beta_s^2 \gamma_s^{2h_n^a}}$ then
18         $\mathcal{T}_a^I(n+1) = \mathcal{T}_a^I(n) \cup \{(h_n^a+1, 2i_n^a-1), (h_n^a+1, 2i_n^a)\}$;
19         $B_{h_n^a+1,2i_n^a-1}^a(n) = B_{h_n^a+1,2i_n^a}^a(n) = infinity$;
20     $N_{C_n}^a(n) \leftarrow N_{C_n}^a(n) + 1$;
       /* $l_n$: level of $C_n$ at round $n$ */
21     if $N_{C_n}^a(n) \ge \delta'_{l_n} = \lambda 2^{p l_n d_c} \ln n$ then
22         Partition subspace $C_n$;
```

*Assumption 2:* For $\forall \mathcal{R}_{h,i}^I \in \mathcal{R}_{0,1}^I$, there exist constants $\beta_s > \beta_{s_1} > 0, \gamma_s \in (0, 1)$, such that the diameter of cluster at depth $h$ satisfies the following constraint: $\beta_{s_1} \gamma_s^h \le \mathcal{DIAM}_{\mathcal{R}_{h,i}}^I \le \beta_s \gamma_s^h$.

## IV. Proposed Algorithm

### A. Algorithm Description of T-DPTDO

In this subsection, we present our Tree-based Differentially-Private and Trustworthy social multimedia Distributed Online learning algorithm (T-DPTDO) in Algorithm 1. We first introduce some useful notations. We denote by $(h_n^a, i_n^a)$ the MC cluster selected at round $n$ by EN $a$ and let $C_n$ be the context subspace $c^a(t_n)$ belongs to at round $n$. Let $l_n$ be the level of $C_n$. We introduce probability event:

$$\mathcal{E}_{h,i}^a(\tau) = \{(h_\tau^a, i_\tau^a) = (h, i),$$
$$\tau < n, c^a(t_\tau) \in C_n, (h, i) \in \mathcal{T}_a^I(n), a \in A\},$$

**Algorithm 2:** TNA

```
23 Input: $C_n$, EN $a$, cluster $(h, i)$;
24 Initialization: $\mathcal{T}^a(n)_{C_n} = \{(0, 1)\}$;
25 Establish $\mathcal{T}^a(n)_{C_n}, \Xi^a(n)_{C_n}$;
     /* $r_{x,y}^{h,i}$: the sum of rewards of MCs in
        cluster $(h, i)$, which is stored in node
        $(x, y) \in \mathcal{T}^a(n)_{C_n}$;
     $\varepsilon' = \varepsilon / |A|$
26 $\tilde{\mu}_{C_n}^a(n) = \sum_{(x,y)\in\Xi^a(n)_{C_n}} \left(r_{x,y}^{h,i} + Lap(\frac{\ln n}{\varepsilon'})\right)$;
27 Output: $\tilde{\mu}_{C_n}^a(n)$.
```

where $\mathcal{T}_a^I(n)$ denotes the MC-cluster tree structure of EN $a$ at round $n$. Let $\mathbb{I}_{\mathcal{E}_{h,i}^a(\tau)}$ be the indicator function of event $\mathcal{E}_{h,i}^a(\tau)$. Let $H(n)$ be the depth of $\mathcal{T}_a^I(n)$ at round $n$. Let $T_{h,i}^a(n) = \sum_{\tau<n} \mathbb{I}_{\mathcal{E}_{h,i}^a(\tau)}$, which denotes the number of times the MC cluster $(h, i)$ has been selected at EN $a$ up to round $n$. Let $M_{h,i}^a(n) = \sum_{b\in \mathrm{N}_a/\tilde{\mathrm{N}}_a(n)} T_{h,i}^b(n)$, where $\mathrm{N}_a$ is the set composed of EN $a$ and its OHNs and $\tilde{\mathrm{N}}_a(n)$ is the set of untrusted OHNs of EN $a$ at round $n$ (see Section IV-C). We denote by $|\mathrm{N}_a| - |\tilde{\mathrm{N}}_a(n)|$ the number of trusted ENs at round $n$. Let $N_{C_n}^a(n)$ be the number of context arrivals in context subspace $C_n$ at EN $a$ till round $n$. We denote $\tilde{\mu}_{h,i}^a(n)$ by EN $a$'s empirical estimated mean reward of MCs in cluster $(h, i)$, which can be computed as

$$\tilde{\mu}_{h,i}^a(n) = \frac{1}{M_{h,i}^a(n)} \sum_{\tau<n} \omega_{u(t_n),u(t_\tau)}^a(t_n) r^a(t_\tau) \mathbb{I}_{\mathcal{E}_{h,i}^a(\tau)}, \quad (2)$$

where $\omega_{u(t_n),u(t_\tau)}^a(t_n)$ is the social intimacy between user $u(t_n)$ and user $u(t_\tau)$ at round $n$, where user $u(t_\tau)$ has arrived at the system in a previous round $\tau$. Note that we have a slight abuse of notation in (2): the $M_{h,i}^a(n)$ stands for $T_{h,i}^a(n)$ when EN $a$ performs stand-alone reward updates (line 10 in Algorithm 1), while it stands for the total number of sum rewards (line 26 in Algorithm 2) updates of all its trusted OHNs when the information sharing is feasible (line 7-8 in Algorithm 1).

When MCs in a cluster have high empirical estimated mean reward, this MC cluster has a good reputation and is probable to keep providing high rewards in the future. However, if we just choose MCs based on their empirical estimated mean rewards, we may neglect those MCs which do not provide a high reward recently but will have performance promotion in the future. Thus, to balance the trade-off between exploration and exploitation, we set a *B-value* for each MC cluster. The MCs in clusters with a bigger B-value are more likely to be selected by ENs (detailed in Algorithm 1). The B-value of cluster $(h, i)$ is denoted by $B_{h,i}^a(n)$. For $\forall(h, i) \in \mathcal{T}_a^I(n)$:

$$B_{h,i}^a(n) = \begin{cases} \tilde{\mu}_{h,i}^a(n) + \beta_s \gamma_s^h + \sqrt{c \frac{\ln n}{M_{h,i}^a(n)}} & M_{h,i}^a(n) > 0 \\ infinity, & \text{otherwise} \end{cases}$$

As shown above, as the running round increases, if an MC cluster has seldom been selected, its B-value will increase. Then, its possible future better performance will not be neglected. When $\beta_s \gamma_s^h > \sqrt{c \frac{\ln n}{M_{h,i}^a(n)}}$ as the threshold condition, i.e., $M_{h_n^a, i_n^a}^a(n) \geq \delta_{h_n^a} = \frac{c \ln n}{\beta_s^2 \gamma_s^{2 h_n^a}}$, it means the size of the cluster becomes a more important impact factor on B-value than the uncertainty caused by the randomness of the rewards. In other words, cluster $(h, i)$ has been fully explored and the number of data arrivals in $(h, i)$ is so large that we need to expand cluster $(h, i)$. We describe the T-DPTDO by the following four phases.

*Policy Adaptation:* First, the system decides whether or not it should adapt its strategy (line 6–10). A trust threshold $TSL$ is set by each EN to decide whether or not it should tend to its OHNs for shared information. The higher the $TSL$, the more information will be shared. Since the shared information (reward, context) might disclose the repository of service providers, the rewards received by each EN will be released using a differential private LM (line 7) by calling the auxiliary function TNA in Algorithm 2 (described in next subsection).

*Accuracy Prediction:* Then, T-DPTDO needs to predict the performance of each MC cluster, which is evaluated by B-values. To calculate B-values, T-DPTDO first finds the contexts subspace $C_n$ that $c^a(t_n)$ belongs to (line 5). Then, B-values of all leaf nodes in the MC-cluster tree will be updated (line 11–12). The level of B-values represents the expected reward of MCs in this MC cluster for the current users.

*MC Recommendation:* Since the selected MC cluster might be attacked or shared with other ENs, directly selecting the MC cluster with the highest B-value will expose user's context. Thus, EN $a$ selects an MC cluster according to the computed probability distribution using EM (line 13–14), where $\mathbb{P}[(h_n^a, i_n^a) = (h, i)]$ in line 14 is the score function for any leaf node $(h, i) \in \mathcal{T}_a^I(n)$ and $\Delta_B$ is the $l_1$ sensitivity of B-value. Since the MCs in the cluster expected to have the highest reward might not be selected, the attackers can not accurately infer the context of users. Then, T-DPTDO selects a leaf node $(h_n^a, i_n^a) \in \mathcal{T}_a^I(n)$ based on the above computed probability distribution, and randomly recommends an MC $i^a(t_n)$ for user $u^a(t_n)$ and receives the reward $r^a(t_n)$ (line 15–16).

*MC-cluster tree and Context Space Update:* To get accurate reward estimation of each MC cluster, EN $a$ needs to judge whether $(h_n^a, i_n^a)$ has been fully explored by setting a threshold (line 17–19). When $M_{h_n^a, i_n^a}^a(n)$ exceeds this threshold, it means MC cluster $(h_n^a, i_n^a)$ has been fully explored and then it is partitioned uniformly and new nodes are added to the tree structure (line 18–19). Finally, Algorithm 1 updates the number of context arrivals at EN $a$ in subspace $C_n$ (line 20). When there are sufficient context arrivals in $C_n$ (line 21), this means that this subspace needs to be partitioned to limit the size of the analyzed components. In this case, Algorithm 1 further partitions the current subspace $C_n$ (line 22). Finally, $C_n$ will be substituted by the new generated subspaces. For exmaple, in Fig. 3, $d_c = 2$, $C_x$ is substituted by four subspaces of level $l_y$ at $t_y$.

### B. Differential Private Framework and TNA algorithm

The differential private framework is mainly composed of two components: i) the privacy-preserving module for users and ii) the privacy-preserving module for ENs. The first module is realized in the Algorithm 1 (line 13–14) deploying an EM. The second module is actually aimed at protecting the privacy of information shared by ENs.

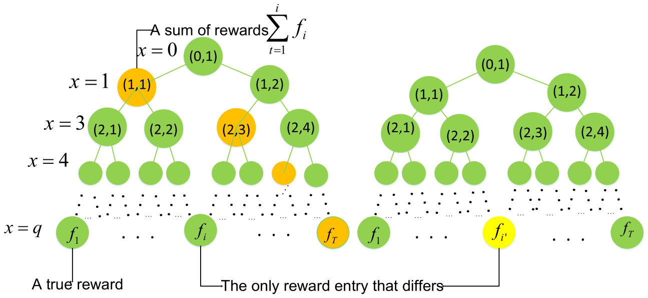


Fig. 4. Tree based method of adding noise.

We show an example why we need to apply EM to hide the true and best MC-cluster in [51]. The following theorem shows that our EM can guarantee DP of users' contexts.

*Theorem 1:* Algorithm 1 can guarantees $\varepsilon$-differential privacy for user's sensitive contexts.

*Proof:* See proof of Theorem 1 of [51]. ∎

Theorem 1 shows that the query results of any data analyst are insensitive to the change of one single user's record in the data base, so users' sensitive contexts cannot be inferred from the released recommendation results.

Since the request for MC sharing is a legitimate behavior, the trusted third party cannot stop an EN to infer other ENs' repositories. This is why we need to use a LM to protect to privacy of ENs (See another example in [51]). To guarantee the privacy of shared information, a naive method widely applied in DP is to add noise to each reward. But in a big data system, this will result in a large amount of noise, making the shared information useless for ENs. Thus, motivated by [48], [49], we propose a Binary Tree-Based Noise Aggregation algorithm (TNA) in Algorithm 2 and is described as follows.

*Algorithm Description of TNA:* When $a$ requests shared information from its one-hop neighbors at round $n$ (6–8), a binary tree, denoted by $\mathcal{T}^a(n)_{C_n}$, which stores all historical data arrivals in context subspace $C_n$ at all ENs in set $\mathrm{N}_a$ till round $n$ is created (line 24). As shown in Fig. 4, each leaf node stores a previously observed reward by an EN in $\mathrm{N}_a$, while the associated user context is in subspace $C_n$. Each internal node stores a sum of rewards stored in all the leaf nodes rooted at this node. Leaf nodes are ranked from left to right according to the absolute time stamps of their stored rewards. For example, in Fig. 4, if $b, c \in \mathrm{N}_a$ such that $f_1 = r^b(t_i)$, $f_T = r^c(t_j)$, where $i$ and $j$ are $f_1$'s and $f_T$'s local arriving round at EN $b$ and $c$ respectively, we have $t_i < t_j$. Meanwhile, a subset $\Xi^a(n)_{C_n}$ of nodes in $\mathcal{T}^a(n)_{C_n}$, such that records stored in nodes in $\Xi^a(n)_{C_n}$ are disjoint and cover all the historical data arrivals in context subspace $C_n$ at all ENs in $\mathrm{N}_a$ till round $n$, will be established (line 25). We get $\Xi^a(n)_{C_n}$ by selecting one node at each depth of $\mathcal{T}^a(n)_{C_n}$ (e.g., the nodes marked in orange in Fig. 4). Let $q$ be the depth of $\mathcal{T}^a(n)_{C_n}$. Then, we add a Laplace noise to each sum stored in nodes in $\Xi^a(n)_{C_n}$, and send these $q$ sums with $q$ Laplace noise to $a$ to preserve privacy of its neighbors' repository (line 7 and 25-26). For simplicity, the total number of data arrivals in subspace $C_n$ at ENs in set $\mathrm{N}_a$ is assumed to be $T = 2^{\nu} \leq n$, where $\nu \in \mathbb{Z}, \nu \geq 0$. Let $D, D'$ be two database differing in only one historical entries (i.e., $\|D - D'\|_1 \leq 1$). For any given $\mathrm{N}_a$ and $C_n$, the binary tree based on $D$, denoted by $\mathcal{T}_D$ (e.g., tree on the left in Fig. 4) and the binary tree based on $D'$, denoted by $\mathcal{T}_{D'}$ (e.g., tree on the right in Fig. 4), have only one different leaf node (e.g., in Fig. 4, $f_i$ and $f_{i'}$). Intuitively, $\mathcal{T}_D$ and $\mathcal{T}_{D'}$ differ in at most $\ln n$ reward sums.

Compared with traditional LMs, the benefit of our TNA is that it reduces the amount of Laplace noise from $\mathcal{O}(n)$ to $\mathcal{O}(logn)$. Thus, it can keep the utility of the aggregated data and preserve the privacy of ENs simultaneously. The following Theorem shows that T-DPTDO can guarantee the privacy of ENs' repositories.

*Theorem 2:* Algorithm 1 can preserve $\varepsilon$-differential privacy for ENs' revenue of rewards.

*Proof:* See proof of Theorem 2 of [51]. ■

Theorem 2 supports that an EN cannot extract information about MC repositories in its neighbors from the shared information, since the rewards received by two different MC clusters are in high probability the same. In summary, Theorem 1 and 2 prove that T-DPTDO guarantees the privacy of users and ENs simultaneously.

## *C. Trust Mechanism Design*

Before retrieving MCs, each user should assess the securities of nearby NEs and find a trustworthy one to recommend the MCs. We introduce a trust evaluation mechanism for users to select the secure and trustworthy ENs. The factors for security evaluation include: the QoE of users encoded by the cumulative MC reward obtained from the EN, the rating on the requested sharing MC size, the sharing price, and the time duration of the MC retrieval at each round.

At first, after a user $u$ retrieved an MC from an EN $a$, it obtains the empirical estimated mean reward $\tilde{\mu}^a_{h_k,i_k}(k)$ of MCs in the selected cluster $(h_k, i_k)$ in (2) under the EN $a$ at round $k$, which provides an important rating factor on the reliability and secure quality of this EN. From a long term view, if the EN is reliable and secure, the rating factor will be high. We define $RA^a_{h_k,i_k}(k) = log(1 + \tilde{\mu}^a_{h_k,i_k}(k))$, where $\tilde{\mu}_{h_k,i_k}(k) \in [0, 1]$. Then, the distrust rating factor is $DR^a_{h_k,i_k}(k) = 1 - RA^a_{h_k,i_k}(k)$.

Second, the requested sharing MC size and the sharing price in different interactions between users and EN $a$ may be different at different rounds upper to the current time stamp $t_n$. The rating factor for the service of MC with large size or sharing price has a high effect on the mechanism design of direct trust. Let $s^a_{h_k,i_k}(k)$ and $p^a_{h_k,i_k}(k)$ denote the the requested sharing MC size and the sharing price in the selected cluster $(h_k, i_k)$ at round $k$, respectively.

Third, as the empirical estimated mean reward $\tilde{\mu}^a_{h_k,i_k}(k)$ could usually vary significantly at short time, we need to observe the long term cumulative trust of ENs. Thus, we define an exponential decay function on an EN $a$ over time $\Gamma = e^{-\phi(t_n - (t_{k+1} - t_k))}$, where $\phi \geq 0$ and $t_{k+1} - t_k$ indicates the time duration of the MC retrieval at round $k$. Then, we obtain the *positive satisfaction* experience as:

$$PS^a_{h_k,i_k}(t_n) = \sum_{k=1}^{T^a_{h,i}(t_n)} s^a_{h_k,i_k}(k) p^a_{h_k,i_k}(k) RA^a_{h_k,i_k}(k)\Gamma.$$

Similarly, we have the *dissatisfactory* experience as:

$$DS^a_{h_k,i_k}(t_n) = \sum_{k=1}^{T^a_{h,i}(t_n)} s^a_{h_k,i_k}(k) p^a_{h_k,i_k}(k) DR^a_{h_k,i_k}(k)\Gamma.$$

Therefore, trust can be expressed as

$$TS^a_{h_k,i_k}(\tau) = \frac{PS^a_{h_k,i_k}(\tau)}{PS^a_{h_k,i_k}(\tau) + \delta DS^a_{h_k,i_k}(\tau)}, \tag{3}$$

where $\delta \geq 1$ is a punishment factor for the malicious ENs. The value of trust $DT^a_{h_k,i_k}(\tau)$ is compared with the *trustworthy level* $TSL$ decided by each EN itself at round $\tau$. Then, the set of untrusted OHNs of EN $a$ is denoted as $\tilde{\mathrm{N}}_a(\tau)$. We note that in work [6], the authors have proposed another trust scheme to evaluate the social relationship among users in MEC, while we have already considered the social intimacy among users in the empirical estimated mean reward (2) and so the social relationship is accounted in (3).

## *D. Regret Analysis*

In this subsection, we consider the prediction accuracy of Algorithm 1. The inaccuracy comes from the added noise which disturbs the ENs to make optimal selections, and deficiencies in historical results at the early stage of exploration which arises suboptimal predictions. These two kinds of inaccuracies are unavoidable, so there is a lower expected cumulative regret bound (LECR) of Algorithm 1. However, due to TNA, the following lemma shows that the noise added to each B-value is bounded by $\mathcal{O}\left(\frac{\ln n}{\varepsilon} \ln(\ln(\frac{n \ln n}{\rho}))\right)$ with probability $\geq 1 - \rho$.

*Lemma 1:* Given $G(A, E)$, for any MC cluster $(h, i) \in \mathcal{T}^I_a(n)$ and $\forall a \in A$, with probability $\geq 1 - \rho$, the amount of noise added to $B^a_{h,i}(n)$ till round $n$, denoted by $\daleth_{h,i}$, is $\frac{|A| \ln n}{\varepsilon} \ln(\ln(\frac{n \ln n}{\rho}))$ at most.

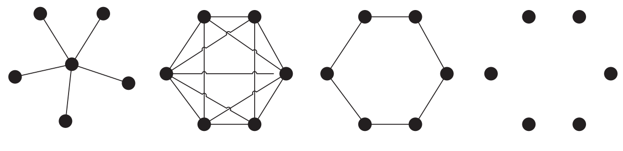
*Proof:* See proof of Lemma 1 of [51]. ■

Lemma 1 shows that due to the tree-based noise aggregation mechanism, there is a bound on the added noise to each B-value. Furthermore, the following lemma implies that a suboptimal node can hardly been selected by any EN in the network, after it has been selected for a certain number of times. We denote by $D_{C_n} = 2^{-L(n)}$ the smallest side length of context subspace in the context space at round $n$, where $L(n)$ denotes the depth of the context space at $n$.

*Lemma 2:* Given $G(A, E)$, for $\forall a \in A$ and any suboptimal node $(h, i)$, let $\varpi^a_{h,i} = \min\left\{\tau \leq n : M^a_{h,i}(\tau) \geq \kappa_{h,i} = \lceil \frac{4c\daleth_{h,i}}{(\Delta_{h,i} - 2L_c D_{C_n} - \beta_s \gamma^h_s)^2}\right\}\rceil$, where $\daleth_{h,i}$ is the same notation as that in Lemma 1 and $c > 4$ . Letting $\Delta_{(h*,i*)} = 0$, we have $\mathbb{P}\{B^a_{h,i}(n) > B^a_{h*,i*}(n), \forall n \geq \varpi^a_{h,i}\} \leq 2n^{-\frac{c}{2}}$.

*Proof:* See proof of Lemma 2 of [51]. ■

Then, based on the above lemmas, we can bound the upper expected cumulative regret (UECR) of T-DPTDO in the following theorem.



(a) (b) (c) (d)

Fig. 5. (a) 6-EN star edge network; (b) 6-EN fully connected edge network; (c) 6-EN circular edge network; (d) 6 stand alone ENs without connection to each other.

*Theorem 3:* Given $G(A,E)$, UECR of Algorithm 1 is

$$\begin{aligned}\mathbb{E}R(n) \leq\ & 14L_c\sqrt{d_c}(4\lambda)^{\frac{1}{(p+1)d_c}} \ln n^{\frac{1}{(p+1)d_c}} n^{1-\frac{1}{(p+1)d_c}} \\ & + 14\beta_s\,(\ln n)^{\frac{1}{3(p+1)d_c}} n^{1-\frac{1}{3(p+1)d_c}-\frac{2}{3pd_c}} \\ & + \frac{512}{3\varepsilon}cG_u|A|\frac{L_c\sqrt{d_c}}{\beta_s^2\beta_{s_1}^{d_0}}(\ln n)^{3+\frac{1-d_0}{3(p+1)d_c}} n^{\frac{3p+3pd_0+4+2d_0}{3p(p+1)d_c}} \\ & + \frac{2|A|L_c}{\varepsilon}(\ln n)^3,\end{aligned}$$

where $c > 4, \Delta' = (3d_c - 3d_0 - 6)^2 + 12d_c(2d_0 + 6) > 0$,

$$p > p_1' = \frac{-(3d_c - 3d_0 - 6) + \sqrt{\Delta'}}{6d_c},$$

and $G_u$ is a parameter associated with the edge network structure $G(A,E)$ among ENs. Particularly, $G_u$ is a parameter associated with $G(A,E)$ among ENs. Specifically, $\kappa_{h,i}$ being the same notation as that of Lemma 2, $G_u\kappa_{h,i}$ represents the maximum required number of reward samples in cluster $(h,i)$ by the entire edge network at round $n$ such that each EN has access to at least $\kappa_{h,i}$ samples (including the shared samples from the neighbors) of cluster $(h,i)$ (i.e., $M_{h,i}^a(n) \geq \kappa_{h,i}$). As shown in Fig. 5, from Fig. 1(a)–(d), we have $G_u = \lfloor\frac{|A|}{2}$ for star edge network, $G_u = 1$ for fully connected edge network, $G_u = |A| - 1$ for circular edge network and $G_u = |A|$ for stand alone ENs without connection. Note that $d_0$ is the near-optimality dimension of MC space $I$ (see Lemma 4 in [51]), and for most of the cases $d_0 = 0$ [47].

*Proof:* See proof of Theorem 3 of [51]. ■

*Remark 1:* In Theorem 3, the last term in the regret upper bound is caused by applying EM, which can be ignored compared to the regret term caused by inherent gap. In addition, we can see a trade-off between the privacy-preserving level and prediction accuracy (i.e., smaller $\varepsilon$ will results in bigger UECR). Furthermore, we can see that with the improvement of the connectivity (i.e., $G_u$ decreases), UECR decreases. This is because historical records to be shared and analyzed will be more sufficient as the edge network becomes much more densely connected.

## V. Proposed Adaptive Algorithm

There are usually some ENs in the edge network named as *dominator*, which act as centers of other ENs. Dominators are featured by having more OHNs than common ENs. Therefore, predictions from dominators are intuitively probable to be more accurate, since dominators have more side observations from their neighbors to expedite the distributed online learning process. A good choice for the EN is to choose the dominating set of the edge network graph. So in this section, we propose an adaptive Dominator-centered Tree-based Differentially-Private and Trustworthy Distributed Online learning algorithm (DT-DPTDO), which takes the dominators in the edge network as the main decision makers while other ENs provide side information. Specifically, we define *dominator subset* $D_G \in A$ as a subset composed of dominators in network $G(A,E)$ such that, for $\forall a \in A \setminus D_G$, $a$ has at least one OHN in $D_G$. DT-DPTDO works as follows: given a dominator subset $D_G \in A$, we partition $G(A,E)$ into $|D_G|$ groups such that each group is composed of one dominator, which belongs to dominator subset $D_G$ and its one-hop neighbors. Then, for all the ENs in the dominator subset $D_G$, they follow the same policy of T-DPTDO while others follow their dominators in their own groups.

```
Algorithm 3: DT-DPTDO
Input: $G(A,E)$, $D_G$;
Initialization:
    $t = 0, n = 1; \mathcal{T}_a^I(n) = \{(0,1),(1,1),(1,2)\}; B_{1,1}^a(n) = B_{1,2}^a(n) = infinity; H(n) = 0$;
Partition $G(A,E)$ based on $D_G$;
if EN $a \in D_G$ then
    Repeat line 2-21;
else
    Follow the selections of its group's dominator;
```

### A. Privacy and Regret Bound Analysis

First, we obtain privacy results as that of Algorithm 1 as follows.

*Theorem 4:* DT-DPTDO can guarantees $\varepsilon$-differential privacy for users' sensitive contexts and ENs' MC repositories.

*Proof:* Since the DP of the algorithm is independent of the edge network structure, following the same analysis in Theorem 1 and 2, we get Theorem 4. ■

Theorem 4 proves that DT-DPTDO guarantees $\varepsilon$-privacy for both users and ENs as T-DPTDO does. To derive UECR of DT-DPTDO, similar to Lemma 2, we bound the expected number of selections in suboptimal clusters.

*Lemma 3:* Given the edge network $G(A,E)$ and dominator subset $D_G$, for any suboptimal cluster $(h,i)$, we have

$$\mathbb{E}\left[\sum_{a\in A} T_{h,i}^a(n)\right] \leq \frac{4c|A||D_G|(\ln n)^3}{\varepsilon(\Delta_{h,i} - 2L_c D_{C_n} - \beta_s\gamma_s^h)^2} + 1 + |A| + |A|\frac{2}{c-4},$$

where $c > 4$ and $|D_G|$ denotes the number of groups in $G(A,E)$ partitioned according to $D_G$.

*Proof:* See proof of Lemma 7 of [51]. ■

Lemma 3 bounds the expectation number of samples an EN has access to. Based on Lemma 3, we can derive the upper regret bound of DT-DPTDO's cumulative regret.

*Theorem 5:* Given the edge network $G(A, E)$ and a dominator subset $D_G$, UECR of DT-DPTDO is

$$\begin{aligned}\mathbb{E}R(n) \leq\ & 14L_c\sqrt{d_c}(4\lambda)^{\frac{1}{(p+1)d_c}} \ln n^{\frac{1}{(p+1)d_c}} n^{1-\frac{1}{(p+1)d_c}} \\ & + 14\beta_s \left(\ln n\right)^{\frac{1}{3(p+1)d_c}} n^{1-\frac{1}{3(p+1)d_c}-\frac{2}{3pd_c}} \\ & + \frac{512}{3\varepsilon}c|D_G||A|\frac{L_c\sqrt{d_c}}{\beta_s^2\beta_{s_1}^{d_0}}(\ln n)^{3+\frac{1-d_0}{3(p+1)d_c}} n^{\frac{3p+3pd_0+4+2d_0}{3p(p+1)d_c}} \\ & + \frac{2|A|L_c}{\varepsilon}(\ln n)^3,\end{aligned}$$

where the constraints of $p$ and $c$ are the same as that of Theorem 3.

*Proof:* See proof of Theorem 5 of [51]. ■

*Remark 2:* From Theorem 3 and 5, we can see that DT-DPTDO decreases the second term of UECR to $\frac{|D_G|}{G_u}$ that of T-DPTDO. This is achieved by choosing the dominators in an edge network as the believable learners, who might give more accurate predictions. This effect will be obvious as the connectivity of the edge network decreases, i.e., $G_u$ increases. So DT-DPTDO can obviously promote the performance of the edge networks with relative low connectivity. From the second-to-last term in the regret upper bound result in Theorem 5, we can also see tradeoffs between the connectivity $D_G$ and privacy loss $\varepsilon$ of MCs in EN repositories.

## VI. Numeral Results

### A. Dataset Description

We use an open dataset, YFCC100M [50], provided by Yahoo in 2014. YFCC100M contains 100 million media objects. There are approximately 0.8 million videos and 99.2 million images. Each object in YFCC100M contains several metadata such as media source, owner name, camera, title, tags and location. We filter a subset of 33,6513 photos and 1799 videos, which have been annotated with visually detected concepts and information of cameras. We set the dimension of MC features $d_i = 6$, which include media type marker (photo=1, video =0), camera maker, camera model, longitude, latitude, timestamp. We filter the MC tagged with the top 25 of 1,570 visually detected concepts and top 25 cameras in the YFCC100M dataset. The information of user behaviors and user contexts (e.g., location, time, userID) is gained using Flickr API. If the user adds a photo to its favorite list or shares it with his/her friends, the reward is 1. Otherwise, we calculate the reward according to the browsing time. Let $t^a(t_n)$ be the longitude of time user $u^a(t_n)$ spent browsing $i^a(t_n)$ and $\bar{t}$ be the average time $u^a(t_n)$ spent browsing recommended MCs. For user $u^a(t_n)$, if $t^a(t_n) \geq \bar{t}$, $r^a(t_n) = 1$, otherwise $r^a(t_n) = 0$. We set the dimensions of context features $d_c = 9$, which contain longitude, latitude, timestamp, userID, etc. All experiments are performed on the computing platform of the first author's university super computing center. The CPU reach 18.46 TFlops. SSD cache is 1.25 TB.

TABLE I
Average Reward

| Algorithm | round $\times 10^4$ | | | | | | | Gain[1] |
|---|---|---|---|---|---|---|---|---|
| | $n=1$ | $n=2$ | $n=3$ | $n=4$ | $n=5$ | $n=6$ | $n=7$ | |
| **Random** | 0.26 | 0.28 | 0.27 | 0.27 | 0.26 | 0.26 | 0.26 | 238% |
| **UCB1** | 0.29 | 0.32 | 0.31 | 0.30 | 0.29 | 0.29 | 0.29 | 200% |
| **DUCB** | 0.33 | 0.33 | 0.34 | 0.34 | 0.34 | 0.34 | 0.34 | 159% |
| **LinUCB** | 0.46 | 0.47 | 0.48 | 0.47 | 0.48 | 0.49 | 0.50 | 76% |
| **DSC** | 0.50 | 0.51 | 0.54 | 0.56 | 0.58 | 0.58 | 0.58 | 51.7% |
| **ACR** | 0.57 | 0.55 | 0.56 | 0.57 | 0.59 | 0.65 | 0.70 | 25.7% |
| **T-DPTDO (one EN)** | 0.56 | 0.62 | 0.64 | 0.65 | 0.67 | 0.68 | 0.71 | 23.9% |
| **T-DPTDO** | 0.57 | 0.65 | 0.74 | 0.80 | 0.84 | 0.86 | 0.88 | |

[1] The gain of T-DPTDO over other algorithms in average reward.

### B. Comparison With Online Learning Algorithms

To show the performance of our proposed algorithms, we first compare them with several related works. First, to show the importance of user context on system performance, we compare the proposed T-DPTDO with those context-free algorithms. We compare T-DPTDO with three context-free algorithms. 1) Random: This algorithm randomly selects an MC at each round. This algorithm is viewed as the benchmark for other algorithms. 2) UCB1 [29]: This classical MAB algorithm performs well while recommending the best MC without considering personalization. Then, to show the importance of information sharing in the social network, we compare T-DPTDO with a distributed online learning system. 3) DUCB [26]: This is a context-free and distributed online learning system using UCB1 algorithm where ENs share their observations through a social network. To show our algorithm outperforms existing context-aware recommender system, we compare T-DPTDO with two context-aware system. 4) DSC [27]: It is a distributed online learning system with context space partitioned in advance. The partition of context space does not evolve over time. 5) ACR [23]: It is a centralized online contextual learning framework with fixed MC clusters. We set $K = 100$ MC clusters in ACR in this experiment. 6) LinUCB [17]: Assuming that the rewards of MCs are linear in contexts, this algorithm recommends the arm with the maximum index. The index is calculated by adding upper bound to the linear combination of previously observed contexts.

The performances are evaluated by the average reward (AD) up to round $n$. Using a PPM or an increasing dataset can decrease AD while information sharing among ENs can increase AD. Thus, for fair comparison, we use a static dataset and remove the PPMs of T-DPTDO in the comparison experiments. A 10-EN star network with one center is built for T-DPTDO, DUCB and DSC. Note that all the numerical results except that shown in Fig. 15 are performed under the situation when $TSL = 0.5$. For fair comparison, we also show the AD of T-DPTDO with one EN. The comparison results is listed in Table I.

*Comparison results with context-free algorithms:* From Table I, we can see that T-DPTDO significantly outperforms the context-free algorithms. The results show that T-DPTDO achieves a 200% performance gain, in terms of AD, over UCB1. Note that UCB1 learns faster than T-DPTDO (converges faster). Because context-free algorithms estimate the AD of an MC according to the historical results of all users, while a context-aware algorithm estimates the AD of an MC according to

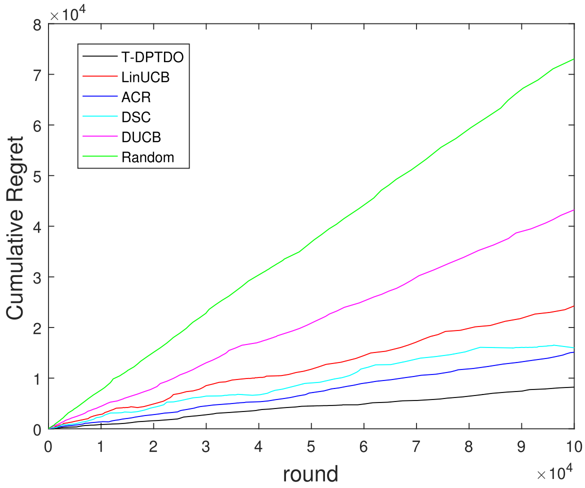


Fig. 6. Cumulative regret-static.

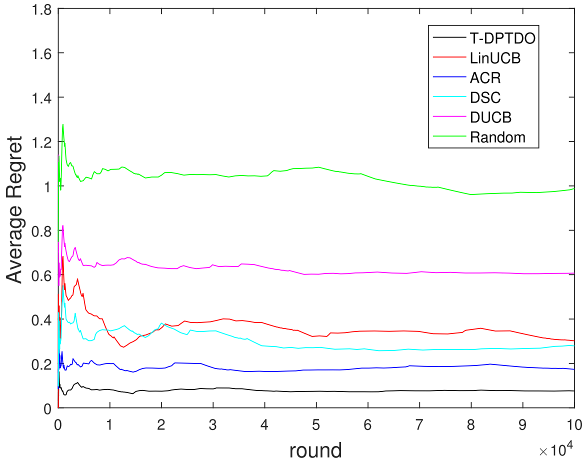


Fig. 7. Average regret-static.

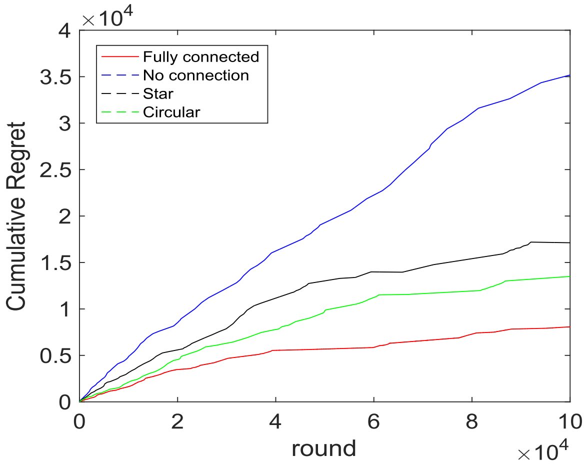


Fig. 8. Various networks-cumulative.

specific user contexts. However, as the number of user arrivals becomes sufficiently large, context-aware algorithms can predict more accurately on the expected rewards of MCs using the contextual information. Thus, the context-free algorithms converge faster (less than $n = 4 \times 10^4$ of user arrivals) than the context-aware algorithms. However, in the long run, contextual algorithms can achieve a higher AD than context-free algorithms.

*Comparison results with context-aware algorithms:* We can see that T-DPTDO significantly outperforms the existing context-aware algorithms. The simulation results show that the AD of T-DPTDO up to $n = 7 \times 10^4$ is 76% higher than that of LinUCB, 25.7% higher than that of ACR, and 51.7% higher than that of DSC. This is because T-DPTDO aggregates information in an adaptive subspace of the context space at each round, but DSC considers the fixed context subspace, which decreases the learning efficiency. In LinUCB, payoff function, which can cause inaccurate estimation of expected reward, is assumed to be linear, while our proposed algorithms do not. Note that DSC learns faster than T-DPTDO. However, DSC has a lower accuracy of learning. Furthermore, though DSC supports information sharing in OSNs, it still has a lower AD than that of ACR, which is centralized system without information sharing. This is because T-DPTDO and ACR adopt a dynamic context partition method while DSC adopts a static one. Hence, in a personalized recommender system, an adaptive context partition method can provide more accurate predictions, though it may slow the learning rate.

Extensively, we present the comparison with three most related algorithms in terms of average (AR) regret and cumulative regret (CR) in Fig. 6 and 7. For example, in Fig. 7, we can see that T-DPTDO outperforms DUCB, DSC and ACR with a decrease in average regret (AR) by 25.23%, 48.52% and 78.23% when $n = 2 \times 10^4$. Similar results of cumulative regret (CR) can be gotten in Fig. 6. And, from the slopes of the curves, we find that context-free algorithms learn faster than context-aware algorithms. The most obvious example is DSC. This is also consistent with the results in Table I.

## C. Evaluation of Other Performance

*Impact of Edge Network Structure:* To investigate the impact of edge network structure, we run T-DPTDO on 50-EN fully connected network, star network and circular network respectively according to topology in Fig. 5 with uniformly distributed connecting links, based on the static dataset composed of the whole filtered subset. In Fig. 8 and 9, it is shown that as connectivity in the edge network increases, T-DPTDO get smaller AR and CR. This is consistent with the theoretical analysis, because as connectivity increases, $G_l$ and $G_u$ decreases and each EN has access to more useful information.

*Impact of PPM:* Then, using the static dataset, we present the average accuracy (AC) of our proposed algorithms at different privacy-preserving levels (different $\varepsilon$ values) in Table II and Fig. 10 to show the impact of privacy-preserving level (PPL) on prediction accuracy. In Table II and Fig. 10, we can see that there is a trade-off between the PPL and AC. This is because as the PPL increases ($\varepsilon$ decreases), the noise added to each reward sum get higher, which can lower the reliability of the shared information. When the PPL increases to a degree, the regret will be too large that it will take a very long time to converge to the optimal policy. So the design of PPL is very important. Finally, to mea-

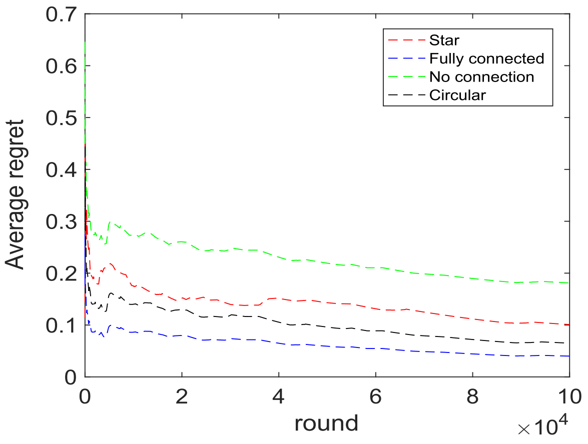


Fig. 9. Various networks-average.

TABLE II
AVERAGE ACCURACY

| **Algorithm** | $\varepsilon$ | **round** $\times 10^4$ $n=1$ | $n=2$ | $n=3$ | $n=4$ | $n=5$ | $n=6$ | $n=7$ |
|---|---|---|---|---|---|---|---|---|
| **DT-DPTDO** | 0.01 | 42.97% | 45.70% | 48.48% | 53.22% | 59.61% | 60.44% | 61.14% |
| | 0.5 | 71.64% | 78.42% | 73.98% | 76.77% | 79.50% | 85.21% | 88.78% |
| | 2 | 95.66% | 96.05% | 96.88% | 96.90% | 98.01% | 98.84% | 99.01% |
| **T-DPTDO** | 0.01 | 42.17% | 45.00% | 48.78% | 55.20% | 56.41% | 59.44% | 60.77% |
| | 0.5 | 70.74% | 71.92% | 72.08% | 74.17% | 76.82% | 85.01% | 87.24% |
| | 2 | 90.46% | 93.07% | 94.88% | 95.93% | 97.24% | 97.91% | 98.06% |

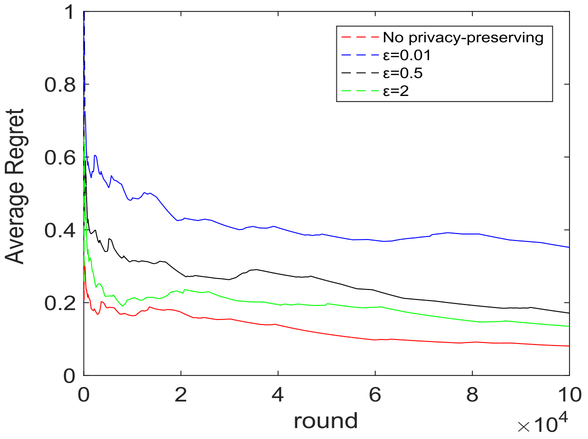


Fig. 10. Average regret-$\varepsilon$ level.

sure the PPL, we use privacy loss $PL = \max_O \ln(\frac{\mathbb{P}[\mathcal{M}(D)=O]}{\mathbb{P}[\mathcal{M}(D')=O]})$ [5] to measure the PPL of our algorithms. From Definition 1, $PL \leq \varepsilon$. Under same experimental setting of T-DPTDO in Table II, we can see in Fig. 11 that the experimental results are close to theoretical results and the PL of T-DPTDO is really small. Thus, our approach has a high PPL.

*Support of Increasing Dataset:* To study T-DPTDO's ability to handle increasing dataset, we run T-DPTDO and DT-DPTDO on 50-EN star network, i.e., 49 uniformly dispersed ENs connects to one central EN. As such, we can avoid the trustworthy level evaluation bias due to different transmission time of MCs under different connecting link qualities between ENs. We study an increasing dataset without any PPM. We set 250,000 images and 1,000 videos as the initial input, then, MCs from the rest filtered 86,513 images and 799 videos are added to database at round $n = 4 \times 10^4$. The results are shown in Figs. 12 and 13. We can see in Fig. 13 that the AR of the proposed algorithms increases at $n = 5 \times 10^4$ but the AR quickly begins to decrease after $n = 6 \times 10^5$. This implies that the proposed algorithms do support an increasing dataset. In Fig. 12, we can see that from $n = 1 \sim 7 \times 10^4$, T-DPTDO has a raise from 15% to 8% over DT-DPTDO in terms of CR. This shows that DT-DPTDO makes more prompt recommendations than T-DPTDO in a star network, but their difference decreases over time.

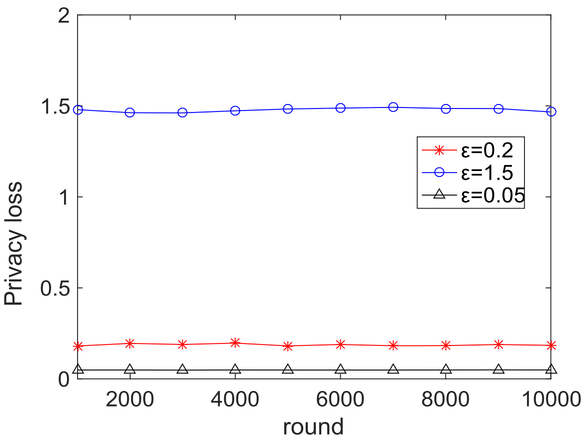


Fig. 11. Privacy loss.

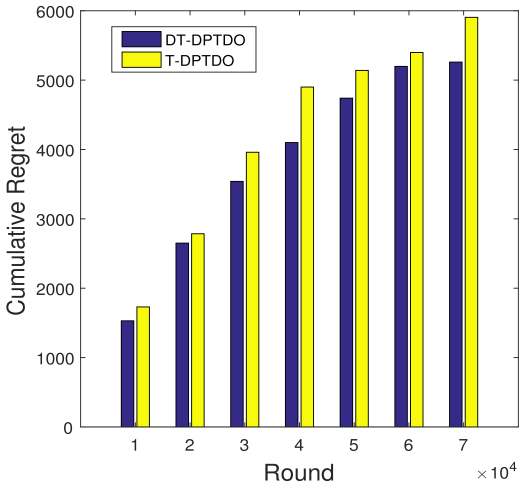


Fig. 12. Cumulative regret.

*Impact of $\eta$:* Each social community will keep a list of its members, and each user will keep a list of its interested communities by joining in communities. And the friend and family lists can be gained using Flickr API. We set different values of $\eta$ to study the role interest factor and friendship factor played in regret bound. From Fig. 14, we can see that when $\eta = 1$, where the Social Intimacy is totally determined by friendship factor, the average accuracy is the lowest. This is because people become online friends for a variety of complex reasons, which means the preference to multimedia contents only plays a tiny role. As $\eta$ decreases, we can see that the average accuracy is

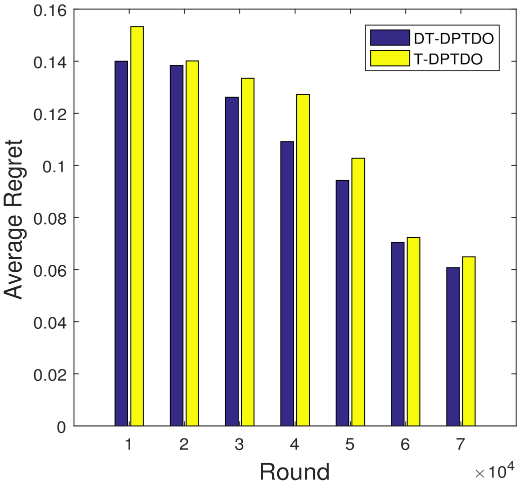


Fig. 13. Average regret.

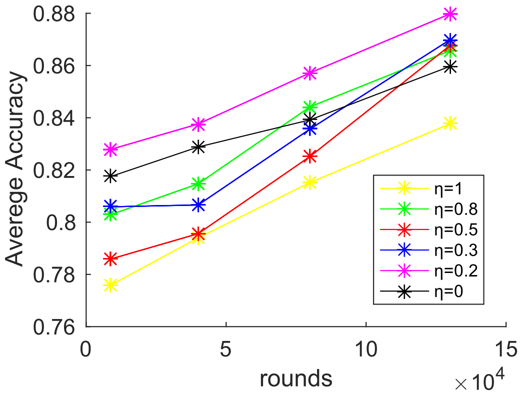


Fig. 14. Effect of $\eta$ on accuracy.

getting higher, because the interest factor is becoming more significant. But when $\eta = 0$, the accuracy is a little lower than that of $\eta = 0.2$. This is because although friendship factor is less important to average accuracy, it still provide some attributions to accuracy. So the optimal value of $\eta$ falls in 0.2–0.3.

This is also reasonable in the real situation, since social websites like Flickr are designed for users to share their media and discuss the contents they are interested in by spontaneously establishing communities composed of people caring about similar topics or making friends with other people. Sharing a content in a community will make it spread much more quickly than sharing it with online friends, since the number of each user's friends is much smaller than the number of members in a community. Furthermore, the reason of becoming a member of a community is usually based on the preference of certain types of multimedia contents. Thus, members in the same social groups share many similarities associated with preference, which makes interest factor contribute more to the real Social Intimacy.

*Impact of $TSL$ level:* The value of $TSL$ shows the frequency of information sharing among secure and trustworthy ENs. The higher $TSL$ is, the more frequent the share. We present the average regret at different $TSL$ values in Fig. 15, which are performed under a 50-EN star network. In our experiments, we set $s^a_{h_k,i_k}(k) = 1$, $p^a_{h_k,i_k}(k) = 1$, $\phi = 30$ and $\delta = 1$ to make fair comparison of the trustworthiness of each EN. We can see from Fig. 11 that AR is inversely proportional to the level of $TSL$. This means that the distributed system indeed improve the prediction accuracy. Further in Table II, average accuracy (AC) of T-DPTDO and DT-DPTDO of different $\varepsilon$ values is shown. We can see that the proposed algorithms achieve high AC in a long run while guarantee an appropriate PPL by selecting a suitable value of $\varepsilon$.

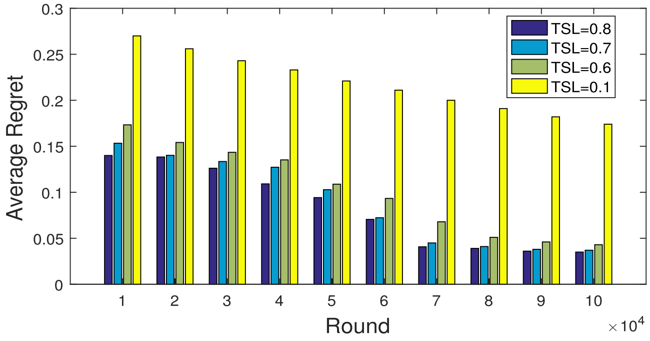


Fig. 15. Average regret-$TSL$ levels.

## VII. Conclusion

We propose a novel differentially-private and trustworthy distributed contextual learning algorithm for social multimedia big data analysis and an adaptive framework to promote the performance extensively. We provide detailed theoretical analysis and evaluate the performance on a real big dataset. Our proposal can achieve the following design objectives: differential users' context privacy, differential ENs' repository privacy, trustworthy ENs and accurate predictions based on users' preference.

## References

[1] 37 Mind Blowing YouTube Facts: Figures and Statistics, 2018. [Online]. Available: https://merchdope.com/youtube-statistics/

[2] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[3] E. Zeydan *et al.*, "Big data caching for networking: Moving from cloud to edge," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 36–42, Sep. 2016.

[4] E. Klarreich, "Privacy by the Numbers: A New Approach to Safeguarding Data," *Quanta Mag.*, vol. 10, 2012.

[5] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," in *Theor. Comput. Sci.*, vol. 9, no. 3–4, pp. 211–407, 2013.

[6] Q. Xu, Z. Su, and M. Dai, "Trustworthy caching for mobile big data in social networks," in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2018, pp. 808–812.

[7] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: A survey," *IEEE Commun. Surv. Tut.*, vol. 16, no. 1, pp. 414–454, Jan./Mar. 2014.

[8] D. Preuveneers and Y. Berbers, "Internet of Things: A context-awareness perspective," *The Internet of Things: From RFID to the Next-Generation Pervasive Networked Systems.* Boca Raton, FL, USA: CRC Press, 2008, pp. 287–307.

[9] A. Slivkins, "Contextual bandits with similarity informatio," in *Proc. Assoc. Comput. Linguistics*, 2011, pp. 679–701.

[10] T. Kameda *et al.*, "Centrality in sociocognitive networks and social influence: An illustration in a group decision making context," *J. Personality Social Psych.*, vol. 73, no. 2, pp. 296–309, Aug. 1997.

[11] Z. Chang, L. Lei, Z. Zhou, S. Mao, and T. Ristaniemi, "Learn to cache: Machine learning for network edge caching in the big data era," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 28–35, Jun. 2018.

[12] R. Dautov *et al.*, "Data processing in cyber-physical-social systems through edge computing," *IEEE Access*, vol. 6, pp. 29822–29835, 2018.

[13] X. Wang, L. T. Yang, X. Xie, J. Jin, and M. J. Deen, "A cloud-edge computing framework for cyber-physical-social services," *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 80–85, Nov. 2017.

[14] L. Pu, X. Chen, J. Xu, and X. Fu, "Content retrieval at the edge: A social-aware and named data cooperative framework," *IEEE Trans. Emerg. Topics Comput.*, p. 1, 2018.

[15] G. Araniti, A. Orsino, L. Militano, L. Wang, and A. Iera, "Context-aware information diffusion for alerting messages in 5G mobile social networks," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 427–436, Apr. 2017.

[16] F. Cicirelli *et al.*, "Edge computing and social internet of things for large-scale smart environments development," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2557–2571, Aug. 2018.

[17] Z. Su *et al.*, "Experience blocking ratio based game theoretic approach for spectrum sharing in heterogeneous networks," *IEEE Trans. Netw. Sci. Eng.*, p. 1, 2018, doi: 10.1109/TNSE.2018.2879674.

[18] Z. Su, L. Hui, and T. H. Luan, "Distributed task allocation to enable collaborative autonomous driving with network softwarization," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2175–2189, Oct. 2018, doi: 10.1109/JSAC.2018.2869948.

[19] Q. Xu *et al.*, "Game theoretical secure caching scheme in multi-homing edge computing-enabled heterogeneous networks," *IEEE Internet Things J.*, p. 1, 2018, doi: 10.1109/JIOT.2018.2876417.

[20] N. Wanigasekara, J. Schmalfuss, D. Carlson, and D. S. Rosenblum, "A bandit approach for intelligent IoT service composition across heterogeneous smart spaces," in *Proc. 6th Int. Conf. Internet Things*, 2016, pp. 121–129.

[21] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. 3rd Conf. Theory Cryptography*, 2006, pp. 265–284.

[22] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *Proc. 48th Annu. IEEE Symp. Found. Comput. Sci.* 2007, pp. 94–103.

[23] L. Song, C. Tekin, and M. van der Schaar, "Online learning in large-scale contextual recommender systems," *IEEE Trans. Services Comput.*, vol. 9, no. 3, pp. 433–445, May/Jun. 2016.

[24] A. M. Ortiz, D. Hussein, P. Soochang, S. N. Han, and C. Noel, "The cluster between internet of things and social networks: Review and research challenges," *IEEE Internet Things J.*, vol. 1, no. 3, pp. 206–215, Jun. 2014.

[25] K. Portelli and C. Anagnostopoulos, "Leveraging edge computing through collaborative machine learning," in *Proc. IEEE Future Internet Things Cloud Workshops*, 2017, pp. 164–169.

[26] S. Buccapatnam, A. Eryilmaz, and N. B. Shroff, "Multi-armed bandits in the presence of side observations in social networks," *IEEE Conf. Decis. Control*, 2013, pp. 7309–7314.

[27] C. Tekin and M. Schaar, "Distributed online big data classification using context information," in *Proc. IEEE 51st Annu. Allerton Conf. Commun., Control, Comput.,* 2013, pp. 1435–1442.

[28] C. Tekin, Z. Shaoting, and M. van der Schaar, "Distributed online learning in social recommender systems," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 4, pp. 638–652, Aug. 2014.

[29] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, no. 2–3, pp. 235–256, 2002.

[30] K. Zhang, X. Liang, R. Lu, and X. Shen, "Exploiting multimedia services in mobile social network from security and privacy perspectives," *IEEE Commun. Mag.*, vol. 52, no. 3, pp. 58–65, Mar. 2014.

[31] S. Shang, Y. Hui, and P. Hui, "Beyond personalization and anonymity: Towards a group-based recommender system," in *Proc. 29th Annu. ACM Symp. Appl. Comput.*, 2014, pp. 266–273.

[32] J. Brickell and V. Shmatikov, "The cost of privacy: Destruction of data-mining utility in anonymized data publishing," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 70–78.

[33] Y. Miao *et al.*, "Lightweight fine-grained search over encrypted data in fog computing," *IEEE Trans. Services Comput.*, p. 1, 2018.

[34] M. Yang, T. Zhu, B. Liu, Y. Xiang, and W. Zhou, "Machine learning differential privacy with multifunctional aggregation in a fog computing architecture," *IEEE Access*, vol. 6, pp. 17119–17129, 2018.

[35] M. Du, K. Wang, Z. Xia, and Y. Zhang, "Differential privacy preserving of training model in wireless big data with edge computing," *IEEE Trans. Big Data*, p. 1, 2018.

[36] Y. He, F. R. Yu, N. Zhao, and H. Yin, "Secure social networks in 5G systems with mobile edge computing, caching, and device-to-device communications," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 103–109, Jun. 2018.

[37] A. Narayanan and V. Shmatikov, "How to break anonymity of the Netflix prize dataset," CoRR, 2006. [Online]. Available: http://arxiv.org/abs/cs/0610105

[38] Z. Jorgensen and T. Yu, "A privacy-preserving framework for personalized, social recommendations," in *Proc. 17th Int. Conf. Extending Database Technol.*, 2014, pp. 571–582.

[39] X. Huang, R. Yu, and J. Kang, "Distributed reputation management for secure and efficient vehicular edge computing and networks," *IEEE Access*, vol. 5, pp. 25408–25420, 2017.

[40] S. Pinto *et al.*, "IIoTEED: An Enhanced, Trusted execution environment for industrial IoT edge Devices," *IEEE Internet Comp.*, vol. 21, no. 1, pp. 40–47, Jan./Feb. 2017.

[41] J. Yuan and X. Li, "A Multi-source Feedback based Trust Calculation Mechanism for Edge Computing," in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2018, pp. 819–824.

[42] G. Bachi, M. Coscia, A. Monreale, and F. Giannotti, "Classifying trust/distrust relationships in online social networks," in *Proc. SocialCom*, 2012, pp. 552–557.

[43] J. Tang, H. Gao, A. Sarma, Y. Bi, and H. Liu, "Trust evolution: Modeling and its applications," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 6, pp. 1724–1738, Jun. 2015.

[44] G. Liu *et al.*, "Opinionwalk: An efficient solution to massive trust assessment in online social networks," in *Proc. IEEE Conf. Comput. Commun.*, 2017, pp. 1–9.

[45] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[46] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Recommender Systems Handbook*, New York, NY, USA: Springer, 2011, pp. 217–253.

[47] M. G. Azar, A. Lazaric, and E. Brunskill, "Online stochastic optimization under correlated bandit feedback," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 1557–1565.

[48] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential privacy under continual observation," in *Proc. 42nd ACM Symp. Theory Comput.*, 2010, pp. 715–724.

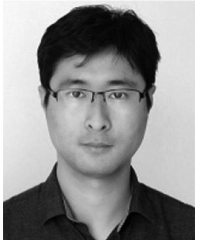
[49] T.-H. H. Chan, E. Shi, and D. Song, "Private and continual release of statistics," *ACM Trans. Inf. Syst. Secur.*, vol. 4, no. 3, pp. 1–23, 2011.

[50] B. Thomee *et al.*, "The new data and new challenges in multimedia research," arXiv preprint arXiv:1503.01817, vol. 1, no. 8, 2015.

[51] P. Zhou, K. Wang, J. Xu, and D. Wu, [Supplementary], "Differentially-Private and Trustworthy Online Social Multimedia Big Data Retrieval in Edge Computing," Oct. 2018. [Online]. Available: https://www.dropbox.com/s/e36x3dvsgicuxmv/Edge_supp.pdf?dl=0

**Pan Zhou** (S'07–M'14) received the B.S. degree in the *Advanced Class* of Huazhong University of Science and Technology (HUST), Wuhan, China, in 2006, and the Ph.D. degree from the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA, in 2011. He is currently an Associate Professor with the School of Electronic Information and Communications, HUST, Wuhan, China. He was a Senior Technical Member with Oracle, Inc., America, from 2011 to 2013, Boston, MA, USA. His current research interests include security and privacy, machine learning and big data analytics, and information networks.

**Kehao Wang** received the B.S. degree in electrical engineering, the M.S. degree in communication and information system from Wuhan University of Technology, Wuhan, China, in 2003 and 2006, respectively, and the Ph.D degree from the Department of Computer Science, the University of Paris-Sud XI, Orsay, France, in 2012. From February 2013 to August 2013, he was a Postdoc with the HongKong Polytechnic University. In 2013, he joined the School of Information Engineering, the Wuhan University of Technology, where he is currently an Associate Professor. Since December 2015, he has been a Visiting Scholar with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, USA. His research interests include stochastic optimization, operation research, scheduling, wireless network communications, and embedded operating systems.



**Jie Xu** (M'15) received the B.S. and M.S. degrees in electronic engineering from Tsinghua University, Beijing, China, in 2008 and 2010, respectively, and the Ph.D. degree in electrical engineering from University of California Los Angeles (UCLA), Los Angeles, CA, USA, in 2015. He is an Assistant Professor with the Department of Electrical and Computer Engineering, the University of Miami, Miami, FL, USA. His research interests include mobile edge computing/caching, green communications, and network security. Prof. Xu is a recipient of the distinguished Ph.D. dissertation award from UCLA and a recipient of the best paper award at APCC.

**Dapeng Wu** (S'98–M'04–SM'06–F'13) received the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2003. He is a Professor with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA. His research interests include areas of networking, communications, signal processing, computer vision, machine learning, smart grid, and information and network security.