IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 1, 2020, accepted April 17, 2020, date of publication April 28, 2020, date of current version May 13, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2991062

Differential Privacy for Weighted Network Based on Probability Model

YUYE WANG, JING YANG, AND JIANPEI ZHANG
College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China

Corresponding author: Jing Yang (yangjing@hrbeu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61672179, Grant 61370083, and Grant 61402126, in part by the Natural Science Foundation of Heilongjiang Province under Grant F2015030, in part by the Science Foundation for Youths of Heilongjiang under Grant QC2016083, and in part by the Postdoctoral Foundation of Heilongjiang Province under Grant LBH-Z14071.

ABSTRACT Weighted network contains a lot of sensitive information and may seriously jeopardize individual privacy. In this paper, we study the problem of differential privacy for weighted network. We found most existing methods add noise to edge weights directly and neglect the structural role of node. These methods perform with low accuracy. To address the above issue, we propose two approaches. One approach describes a differential privacy method for Stochastic Block Model. This private SBM reveals and the structural role of node and respects the privacy of it. Another approach develops a differential privacy method for weighted network through structuring a private probability model. We use Variational Bayes to learn the private model parameters. It adds noise to the parameters of the probability model instead of edge weights, and achieve high data utility. Experiments on real datasets illustrate that our algorithm privately releases weighted network and achieves high accuracy.

INDEX TERMS Weighted network, differential privacy, stochastic block model, variational bayes.

I. INTRODUCTION

Social network is a form of dataset consisting of interactions between pairs of individuals. Network data is represented by graph structure where vertices represent individuals and edges represent interactions. Recently, social network is studied by sociologists, economics and informatics. Many networks contain highly sensitive personal information, and releasing them would pose seriously threats to individual's privacy. To respect the privacy of personal information, network data should be released for public with ''sanitization''. Anonymization techniques(e.g., k-anonymity [1] and l-diversity [2]) are traditional methods to ensure network data privacy. Recently, differential privacy has been proposed as a way to address such privacy problem. Unlike the anonymization methods, differential privacy provides strong theoretical guarantees against adversaries with prior knowledge. The standard technique of ensuring differential privacy is to ''sanitization'' the presence or absence of an edge.

The associate editor coordinating the review of this manuscript and approving it for publication was Longxiang Gao.

Unlike the relational data, vertices of network data are pairwise related and play latent structural roles in generating the network's structure. Community is a common representation of such structural role, and identifies a network partition which groups together vertices with similar structural roles. Such structural role is important to the structure of network. It is necessary to respect the privacy of structural role. However, current locus of differential privacy of network is around topological structure and neglect the structural role. These methods cannot generate synthetic network which has similar structural features with origin network. It is difficult to analyze the cluster features using these private synthetic network. In this paper, we first propose a differentially private SBM algorithm called SSN(Sufficient Statistic Noisy). The stochastic block model (SBM) [3], [4] is a popular generative model for learning community structure of unweighted networks and it presents the connecting probability of pairwise interactions among n vertices. Each vertex belongs to one of K latent groups and the probability of each edge only depends on the group memberships of the vertices. Vertices in the same group play similar structural roles and are equivalent in generating the network's structure. Thus, we could respect the

privacy of structural roles through the differentially private SBM algorithm.

For unweighted networks, the presence or absence of edge is represented as a binary variable. However, most real-world networks are weighted networks which network edges have weights. Christopher Aicher *et al.* [5] introduced the Weighted Stochastic Block Model (WSBM), a generalization of the SBM for weighted networks. WSBM uses an efficient variational Bayes approach to learn the parameters and it handles one technical difficulty in fitting in with weight distributions of edges, such as the degeneracy in the likelihood calculation. What's more, WSBM is important to our differentially private weighted network releasing method.

Perturbing the edge weights directly is a common approach to release differential privacy weighted network. However, this kind of approaches incur excessive noise. As the sensitivity of direct perturbation is the maximum edge. When most of edge weights are much less than the maximum edge, it results that the sensitivity to be prohibitively high and suffers poor performance. In this paper, we propose a differential privacy method for weighted networks, called VB-WNDP(i.e.,Variational Bayes-Weighted Network Differential Privacy). Firstly, we use the idea of SSN algorithm to protect the privacy of structural role. Then, we construct a probability model of weighted network and use Variational Bayes to learn the model parameters. We propose a method to add noise to the model parameters in order to make the model satisfies differential privacy. At last, we generate the sanitized weighted network through synthetic network generation. VB-WNDP not only offer better data utility, but also protect the the privacy of structural roles.

In summary, we present several contributions: (1) We introduce a differentially private SBM algorithm named SSN. This technique makes SBM satisfying differential privacy and protects the privacy of structural role. (2) We develop a differential privacy method for weighted network named VB-WNDP. This method uses the idea of SSN to protect the privacy of structural role and constructs a private weighted network probability model to release the private weighted network. (3) Through formal privacy analysis, we prove that SSN and VB-WNDP both satisfy ϵ-differential privacy. We experimentally study over real datasets, and the results demonstrate that SSN and VB-WNDP perform with high accuracy.

Our paper is organized as follows: Section II provides a literature review on differential privacy for networks. Section III presents necessary background on differential privacy and SBM. Section IV describes the differentially private SBM algorithm SSN. Section V presents the weighted networks differential privacy method VB-WNDP. Section VI reports the comprehensive experimental results. Section VII concludes the paper.

II. RELATED WORK

Many existing works about social network differential privacy focus on social network analysis. These methods output some network statistics under differential privacy such as degree distribution, subgraph number and clustering coefficient. Dwork *et al.* [6] added noise to outcome directly and answered the queries under differential privacy. Hay *et al.* [7] proposed a differentially private method in a post-processing phase to compute the consistent input most likely to have produced the noisy output. They used this to estimate the private degree distribution. Karwa *et al.* [8] expanded this concept to calculate the k-star count of network. Zhang *et al.* [9] analysed the statistics through a ladder function and reduced the sensitivity effectively. Cheng *et al.* [10] presented a two-phase differentially private frequent subgraph mining algorithm called DFG. In DFG, frequent subgraphs are privately identified in the first phase, and the noisy support of each identified frequent subgraph is calculated in the second phase. Ding *et al.* [11] published the triangle counts satisfying the node-differential privacy with two kinds of histograms: the triangle count distribution and the cumulative distribution. Sun *et al.* [12] studied fundamental problems related to extended local view. They formulated a decentralized differential privacy scheme named DDP, which requires that each participant consider not only her own privacy, but also that of her neighbors involved in her ELV. They also designed a multi-phase framework under DDP that enables an analyst to accurately estimate subgraph counts.

Differentially private social network releasing also draws attention. Sala *et al.* [13] introduced a differentially private graph model called Pygmalion for publishing social network. Pygmalion extracts a graph structure into private dK-graph and generates a synthetic graph. Mir and Wright [14] used maximum likelihood estimation to privately estimating the parameters of stochastic Kronecker graph model. Xiao *et al.* [15] proposed a differentially private network publishing method HRG-MCMC. They computed an estimator of graph in the hierarchical random graph(HRG)model under differential privacy, and sampled possible HRG structures in the model space via Markov chain Monte Carlo (MCMC) witch satisfies the exponential mechanism. Qin *et al.* [16] investigated techniques to ensure local differential privacy of individuals while collecting structural information and generating representative synthetic social graphs. They proposed LDPGen which incrementally clusters users based on their connections to different partitions of the whole population and adapted existing social graph generation models to construct a synthetic social graph. Chen *et al.* [17] presented a method for publishing differentially private synthetic attributed graphs, which is able to preserve the community structure of the original graph without sacrificing the ability to capture global structural properties.

Many existing works also focus on weighted network privacy. Liu *et al.* [18] identified weighted $1*$-neighborhood attacks and defined probabilistic indistinguishability to resist this attack. They proposed a HIGA scheme to generate a probabilistically indistinguishable social network. Maria Skarkala *et al.* [19] presented a clustering-based

k-anonymization technique for weighted network. This method groups nodes with similar sets of neighbors and their connections into supernodes and superedges, respectively. Chen *et al.* [20] proposed k-histogram-inverse-l diversity to investigate the sensitive label privacy disclosure problem in weighted graph. Liu *et al.* [21] proposed privacy preserving methods using the centrality based on complex network theory to protect privacy of virtual assets.

Several works focus on weighted network differential privacy. Li *et al.* [22] proposed the Merging Barrels and Consistency Inference strategy to protect weighted social graphs. They merged the barrels with the same count into one group to reduce the noise required. They also did consistency inference according to original order of the sequence as an important postprocessing step to keep most of the shortest paths unchanged. Wang and Long [23] proposed a modified algorithm LMBCI to reduce the more substantial error MBCI generated. Qian *et al.* [24] investigated the problem of publishing the topological information with the weight distribution of the weighted graph. They proposed two clustering approaches based on sequence-aware and local density to aggregate histogram.

According to the above mentioned works, we find that only a few works focus on weighted network differential privacy. These works neglect the structural role and incur excessive noise on account of adding noise to weights. Hence, we introduce a differential privacy method for weighted network which could solve above problems.

III. BACKGROUND

A. DIFFERENTIAL PRIVACY

We model an input network dataset as a graph $G = (V, E)$, where V is the set of vertices and E is the set of edges. Given a graph G, differential privacy [6] insures the outputs to be approximately same even if any edge is arbitrarily added or deleted in the graph. Thus, the presence or absence of any edge has a negligible effect on the outputs. We define two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ to be neighbors if they satisfy $V_1 \subset V_2$ where $|V_2| = |V_1| + 1$, $E_1 \subset E_2$ where $|E_2| = |E_1| + d_{(V_2 - V_1)}$. The degree of the node is denoted as $d(\cdot)$. ϵ-differential privacy is defined as follows:

Definition 1 (ϵ-differential privacy): A randomized algorithm $\mathcal{A}$ is ϵ-differential privacy if for any two neighboring graphs G_1 and G_2, and for any output $O \in Range(\mathcal{A})$,

$$Pr[\mathcal{A}(G_1) \in O] \leqslant e^{\epsilon} Pr[\mathcal{A}(G_2) \in O]. \quad (1)$$

Differential privacy are based on the concept of *global sensitivity* of a function f. It is used to measure the maximum change in the outputs of f when any edge in the graph is changed. The global sensitivity of f is defined as $\Delta f = max_{G_1, G_2} \|f(G_1) - f(G_2)\|_1$.

Differential privacy can be achieved by Laplace mechanism and exponential mechanism. The Laplace mechanism is mainly used for functions whose outputs are real values. Differential privacy can be achieved by adding properly noise drawn randomly from Laplace distribution to the true answer.

Theorem 1 (Laplace Mechanism): [6] For any function $f : G \to \mathbb{R}^d$ with sensitivity Δf, the algorithm

$$\mathcal{A}(G) = f(G) + \left\langle Lap_1\left(\frac{\Delta f}{\epsilon}\right), \ldots, Lap_d\left(\frac{\Delta f}{\epsilon}\right)\right\rangle. \quad (2)$$

satisfies ϵ-differential privacy, where $Lap_i\left(\frac{\Delta f}{\epsilon}\right)$ are i.i.d Laplace variables with scale parameter $\frac{\Delta f}{\epsilon}$.

The exponential mechanism is mainly used for functions whose outputs are not real numbers. The main idea is to sample the output data O from the output space $\mathcal{O}$ according to the utility function u. The global sensitivity of u is $\Delta u = max_{O, G_1, G_2} |u(G_1, O) - u(G_2, O)|$.

Theorem 2 (Exponential Mechanism): [25] Given a graph G and a utility function $u : (G \times \mathcal{O}) \to \mathbb{R}$, the arithmetic $\mathbb{A}$ whose output is with probability proportional to $exp\left(\frac{\epsilon \cdot u(G,O)}{2\Delta u}\right)$ satisfies ϵ-differential privacy.

Theorem 3 (Sequential Composition 1): [26] If each arithmetic $\mathcal{A}_i$ provides ϵ_i-differential privacy, a sequence of $(\mathcal{A}_1(D), \mathcal{A}_2(D), \ldots, \mathcal{A}_n(D))$ over the same database D provides $\sum_{i=1}^{n} \epsilon_i$-differential privacy.

Theorem 4 (Sequential Composition 2): [27] Any subset D_{iter} sampled from D satisfies each data point is included independently with probability p. If algorithm $\mathcal{A}(D_{iter})$ satisfies ϵ_{iter}-differential privacy, $\mathcal{A}(D)$ satisfies $log(1 + p((e^{\epsilon_{iter}}) - 1))$-differential privacy.

B. STOCHASTIC BLOCK MODEL

The adjacency matrix of social network contains binary values A_{ij} which represents edge existences, i.e., $A_{ij} \in \{0, 1\}$. K denotes a fixed number of latent groups and each vertex belongs to one of the K groups. The vector $\mathbf{z}$ represents the group label of each vertex, i.e., $z_i \in \{0, 1, 2, \ldots, K\}$. Vertices in the same group play similar structural roles and connect with vertices from other groups in the same distribution. The variable l_r represents the probability of vertices belongs to group r, and the element $\theta_{z_i z_j}$ in K-by-K matrix $\boldsymbol{\theta}$ represents the connection probability of groups which vertex i and vertex j belongs to respectively. π_r represents the prior distribution of l_r and satisfies $\sum_{r=1}^{K} \pi_r = 1$.

In SBM, $\{A_{ij}\}$ represents the observed data, $\{z_i\}$ represents the latent data which can not be observed directly, $\Theta = \{\theta, \pi\}$ represents the parameters of the model. The likelihood function of SBM is

$$\begin{aligned}
&Pr(A, z \mid \theta, \pi) \\
&= Pr(A, z, \theta \mid \pi) \cdot Pr(z \mid \theta, \pi) \\
&= \prod_j \pi_{z_i} \cdot \prod_{i,j} \theta_{z_i z_j}^{A_{ij}} (1 - \theta_{z_i z_j})^{1 - A_{ij}} \\
&= \prod_j \pi_{z_i} \cdot \prod_{i,j} exp\left(A_{ij} \cdot log\left(\frac{\theta_{z_i z_j}}{1 - \theta_{z_i z_j}}\right) + log\left(1 - \theta_{z_i z_j}\right)\right) \\
&= \prod_j \left(\sum_i \pi_{z_i} \cdot exp\left(s\left(A_{ij}\right) \cdot n\left(\theta_{z_i z_j}\right)\right)\right). \quad (3)
\end{aligned}$$

where $s(\cdot)$ are sufficient statistics and $s(x) = \{x, 1\}$; $n(\cdot)$ are natural parameters and $n(x) = \left\{log\left(\frac{x}{1-x}\right), log\,(1\text{-}x)\right\}$.

IV. DIFFERENTIALLY PRIVATE SBM

A. DIFFERENTIALLY PRIVATE SBM DESCRIPTION

This section shows the details of differentially private SBM algorithm. During the parameter learning process, maximum likelihood estimation is used normally. EM algorithm introduces a probability distribution over latent variables to give rise to a lower bound on log likelihood. EM algorithm iteratively alternates between the parameters and the probability distribution over the latent variables, and its iterative process is

E-step: Given the parameters $\{\theta, \pi\}$, output the latent variables $\{z\}$.

M-step: Given the latent variables $\{z\}$, output the parameters $\{\theta, \pi\}$.

To satisfies the differential privacy, a straightforward approach is to add perturbation noise to both the parameters and the latent variables directly in each iteration. However, this approach may produce much cumulative noise and suffer poor performance. The root cause is: (1) The parameters of each vertex in each iteration produce noise respectively. (2) Latent variables in each iteration produce noise. Thus, we propose a differentially private SBM algorithm called SSN, which uses Variational Bayesian EM(VBEM) to compute the model parameters and obtain the differentially private SBM.

In VBEM, latent variables and model parameters are both treated as random variables. Their posterior distributions $Pr(z, \theta, \pi \mid A)$ are learned. However, the posterior distribution is generally difficult to calculate. Instead, we use a factorizable distribution $q(z, \theta, \pi) = q(\theta, \pi) \cdot \prod_N q(z)$ to approximate the posterior distribution. As SBM falls in the conjugate-exponential (CE) family, the iterative process of VBEM is composed by updating the parameters of CE family:

VBE:

$$q(z) = \prod_N q(z_i). \tag{4}$$

where CE family is expressed as $q(z_i) \propto exp\left(\bar{n}^\top \cdot s\left(A_{ij}, z_i\right)\right)$, and the expected natural parameters is expressed as $\bar{n} = \langle n(\theta, \pi)\rangle_{q(\theta,\pi)}$.

VBM:As the prior over the parameters are conjugate to $q(\theta, \pi)$, $q(\theta, \pi)$ can be expressed as:

$$q(\theta, \pi) = h\left(\eta', \nu'\right) \cdot g(\theta, \pi)^{\eta'} \cdot exp\left(\nu'^\top \cdot n(\theta, \pi)\right). \tag{5}$$

where $\eta' = \eta + N$ and $\nu' = \nu + \sum_N \bar{s}\left(A_{ij}\right)$ are the hyperparameters of prior; $g(\theta, \pi) = \left(\pi_{z_i}\right)^i$; $h\left(\eta', \nu'\right)$ is a normalizing constant. The expected sufficient statistics is expressed as $\bar{s}\left(A_{ij}\right) = \left\langle s\left(A_{ij}, z_i\right)\right\rangle_{q(z_i)}$.

To satisfies the differential privacy in each iteration, we need to add noise to $q(z_i)$ and $q(\theta, \pi)$. For $q(\theta, \pi)$, we update ν' by calculating expected sufficient statistics $\bar{s}\left(A_{ij}\right)$, and then compute $q(\theta, \pi)$. As the algorithm needs to look at the original data A_{ij} when computing $\bar{s}\left(A_{ij}\right)$, we need to add noise to $\bar{s}\left(A_{ij}\right)$. When we compute $q(z_i)$, the algorithm needs to look at the original data A_{ij} directly. So we also need to add noise to $q(z_i)$ directly. However, it produces a excessive amounts of additive noise when adding noise to $q(z_i)$ directly. The reason is that it is necessary to add noise to the latent variables of each vertex and we do not need to output the latent variables during the process of iteration.

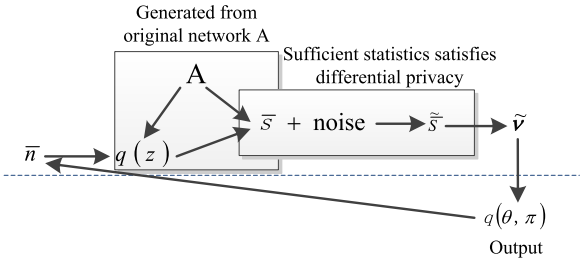


FIGURE 1. The framework of SSN.

To this end, we introduce our SSN algorithm. During the iteration process, we only need to add noise to $\bar{s}\left(A_{ij}\right)$. The output of computing $q(z_i)$ is only treated as the input of computing $\bar{s}\left(A_{ij}\right)$, not other variables. And it is not necessary to look at the original data A_{ij} when computing the natural parameters $n(\theta, \pi)$. Thus, it is not necessary to add noise to $n(\theta, \pi)$. In Fig. 1, we show the framework of SSN.

B. THE GLOBAL SENSITIVITY OF SSN ALGORITHM

As discussed above, SSN algorithm satisfies the differential privacy by adding noise to the sufficient statistics during each iteration. The global sensitivity of the sufficient statistics is the maximum difference of sufficient statistics when any vertex and its adjacent edges both change. More specifically, it equals to $max_{A,A'}\left|\bar{s}(A) - \bar{s}\left(A'\right)\right|$, where A' is the neighbor network. The $K \times K$ expected sufficient statistics of SBM set corresponds to the $K \times K$ bundles.

We assume that P is the edge set between arbitrary groups a and b, and the size of P is p. Q is the number of changing edges of P when any vertex changes, and the size of Q is q. We know that q is less than the number of vertices in group a and b. The global sensitivity of the sufficient statistic between group a and group b is expressed as:

$$\begin{aligned} &max_{A,A'}\left|\bar{s}(A) - \bar{s}\left(A'\right)\right| \\ &= max_{A,A'}\left|\sum_{ij}\sum_{(z_i,z_j)=u} s(A) - \sum_{ij}\sum_{(z_i,z_j)=u} s\left(A'\right)\right| \\ &= \left|\frac{1}{p}\sum_{A_{ij}\in P} s\left(A_{ij}\right)\cdot l_{i,z_i}\cdot l_{j,z_j}\right. \\ &\quad \left. - \frac{1}{p-q}\sum_{A_{ij}\in(\text{P-Q})} s\left(A_{ij}\right)\cdot l_{i,z_i}\cdot l_{j,z_j}\right| \end{aligned}$$

$$
\begin{aligned}
&= \left| \frac{1}{p} \sum_{A_{ij} \in (P-Q)} s\left(A_{ij}\right) \cdot l_{i,z_i} \cdot l_{j,z_j} \right. \\
&\quad - \frac{1}{p-q} \sum_{A_{ij} \in (\mathrm{P\text{-}Q})} s\left(A_{ij}\right) \cdot l_{i,z_i} \cdot l_{j,z_j} \\
&\quad \left. + \frac{1}{p} \sum_{A_{ij} \in Q} s\left(A_{ij}\right) \cdot l_{i,z_i} \cdot l_{j,z_j} \right| \\
&\leqslant \left| \frac{q}{p(p-q)} \sum_{A_{ij} \in (\mathrm{P\text{-}Q})} s\left(A_{ij}\right) \cdot l_{i,z_i} \cdot l_{j,z_j} \right| \\
&\quad + \left| \frac{1}{p} \sum_{A_{ij} \in Q} s\left(A_{ij}\right) \cdot l_{i,z_i} \cdot l_{j,z_j} \right|. \qquad (6)
\end{aligned}
$$

where l_{i,z_i} is the probability that vertex i belongs to group z_i, and satisfies $l_{i,z_i} \leqslant 1$. As a result, the global sensitivity $max_{A,A'} \left|\bar{s}\left(A\right) - \bar{s}\left(A'\right)\right| \leqslant \frac{2q}{p}$.

V. DIFFERENTIAL PRIVACY FOR WEIGHTED NETWORK

We now formally describe the differential privacy weighted networks publishing method VB-WNDP. Unlike the methods which adding noise directly to the weights, VB-WNDP uses the idea of partition in WSBM. The weights in the same group-group relationship obey the same distribution and have the same parameters. In this paper, we model the edge weights with the normal distribution and the edge weights are real-valued. The parameter $\theta_{z_i z_j}$ represents the existence probability of an edge between group z_i and group z_j. It only depends on the group memberships of vertices i and j. It is parameterized by a mean and variance $\theta_{z_i z_j} = \left(\mu_{z_i z_j}, \sigma^2_{z_i z_j}\right)$, and the likelihood is

$$
\begin{aligned}
&Pr\left(A, z \mid \mu, \sigma^2, \pi\right) \\
&\quad = \prod_j \pi_{z_i} \cdot \prod_{ij} \mathcal{N}\left(A_{ij} \mid \mu_{z_i z_j}, \sigma^2_{z_i z_j}\right) \\
&\quad = \prod_j \pi_{z_i} \cdot \prod_{ij} exp\left(A_{ij} \cdot \frac{\mu_{z_i z_j}}{\sigma^2_{z_i z_j}} - A^2_{ij} \cdot \frac{1}{2\sigma^2_{z_i z_j}} - 1 \cdot \frac{\mu_{z_i z_j}}{\sigma^2_{z_i z_j}}\right). \qquad (7)
\end{aligned}
$$

where the sufficient statistic $s = \left(x, x^2, 1\right)$, and the natural parameter $n = \left(\frac{\mu}{\sigma^2}, -\frac{1}{2\sigma^2}, -\frac{\mu^2}{2\sigma^2}\right)$.

We define r as the $K \times K$ indexes between groups. Thus, the parameters $\Theta = \{\theta, \pi\}$ can be represented as $\Theta = \{\Theta_1, \Theta_2, \ldots, \Theta_r\}$. The sufficient statistic and the natural parameter are s_r and n_r respectively. The expected sufficient statistic can be represented as $\bar{s}_r = \frac{1}{p_r} \sum_{ij} \sum_{(z_i,z_j)=r} s\left(A_{ij} \cdot l_{i,z_i} \cdot l_{j,z_j}\right)$, where p_r is the number of edges in r.

We add noise to satisfy the differential privacy by using the idea of SSN algorithm. The variables should be perturbed are merely the expected sufficient statistics. We add Laplace noise to the expected sufficient statistics as

$$
\tilde{\bar{s}}_r = \bar{s}_r + Y_r. \qquad (8)
$$

where $Y_r \sim Lap\left(\frac{\Delta \bar{s}}{\epsilon}\right)$.

A. SENSITIVITY COMPUTATION OF VB-WNDP

We separate the edge into the edge existence and the edge weight respectively. For the edge existence, the number of edge represents the edge existence, specifically 1 represents the existence of edge and 0 represents the inexistence. The linear summation of egde weights represents the number of edges. When we use the node differential privacy, we could use the linear summation of egde weights to compute the sensitivity. However, for the edge weights with integer, the linear summation of edge weights neither represents the number of edges nor the maximum change of the number of edges which the existence or the inexistence of any vertex leads to. What's more, it also do not represents the maximum change of a single edge weight. As a result, we cannot use the linear summation of edge weights to compute the sensitivity when the node differential privacy is used. So we use the edge differential privacy and the neighbour network only changes an edge. We could use the change of an edge weight to compute the global sensitivity.

The global sensitivity of $\bar{s}_r$ is represented as $\Delta \bar{s}_r = max_{A,A'} \left|\bar{s}_r\left(A\right) - \bar{s}_r\left(A'\right)\right|$. We assume that the neighbour network A' changes the maximum edge weight A^r_0 in r. The global sensitivity of $\bar{s}_r$ is

$$
\begin{aligned}
&\Delta \bar{s}_r \\
&= max_{A,A'} \left|\bar{s}_r\left(A\right) - \bar{s}_r\left(A'\right)\right| \\
&= \left| \frac{1}{p_r} \sum_{ij} \sum_{(z_i,z_j)=r} s\left(A_{ij}\right) \cdot l_{i,z_{z_i}} \cdot l_{j,z_j} \right. \\
&\quad \left. - \frac{1}{p_r - 1} \sum_{ij} \sum_{(z_i,z_j)=r \ excluding \ A^r_0} s\left(A_{ij}\right) \cdot l_{i,z_{z_i}} \cdot l_{j,z_j} \right| \\
&\leqslant \left| \frac{1}{p_r} \sum_{ij} \sum_{(z_i,z_j)=r \ excluding \ A^r_0} s\left(A_{ij}\right) \right. \\
&\quad \left. - \frac{1}{p_r - 1} \sum_{ij} \sum_{(z_i,z_j)=r \ excluding \ A^r_0} s\left(A_{ij}\right) + \frac{1}{p_r} \cdot s\left(A^r_0\right) \right| \\
&= \left| \frac{1}{p_r\left(1 - p_r\right)} \sum_{ij} \sum_{(z_i,z_j)=r \ excluding \ A^r_0} s\left(A_{ij}\right) + \frac{1}{p_r} \cdot s\left(A^r_0\right) \right| \\
&= \frac{1}{p_r\left(1 - p_r\right)} \left| \sum_{ij} \sum_{(z_i,z_j)=r \ excluding \ A^r_0} s\left(A_{ij}\right) \right. \\
&\quad \left. + \left(p_r - 1\right) \cdot s\left(A^r_0\right) \right|. \qquad (9)
\end{aligned}
$$

As $s = \left(x, x^2, 1\right)$ is a decreasing function, we could get $s\left(A^r_0\right) \geqslant s\left(A_{ij}\right)_{(z_i,z_j)=r \ excluding \ A^r_0}$. So the global

sensitivity is

$$\Delta\bar{s}_r \leqslant \frac{1}{p_r\left(p_r-1\right)} \cdot 2\left(p_r-1\right) \cdot s\left(A_0^r\right) = \frac{2}{p_r} s\left(A_0^r\right). \quad (10)$$

B. VB-WNDP ALGORITHM DESCRIPTION

The weighted network differential privacy algorithm VB-WNDP is shown in Algorithm 1.

The likelihood has the form of an exponential family and we could compute the sufficient statistics and the natural parameters. We use the idea of SSN algorithm to satisfy the differential privacy. In VBEM, we aim to compute an approximation to the posterior distribution. In conjugate-exponential (CE) class, we could update the hyperparameters (η, ν) during the iteration process to get the approximation to the posterior distribution. During the iteration process, we add noise to the expected sufficient statistics and use them to update the hyperparameter ν. The iteration process operates until the hyperparameters converge. At last for each pair of vertices, we could sample the edge weight through the noisy parameters of exponential family, and then get a sanitized synthetic network.

Algorithm 1 VB-WNDP Algorithm

Input: Input network A, group number K, privacy parameter ϵ

Output:Sanitized network $\tilde{A}$

1. Turn the likelihood of the model to CE class.
2. Update the model parameters (π, θ) in a differentially private way. See Algorithm 2.
3. For each pair of vertices $(i.j)$, place an edge weight $A_{ij} \in \mathbb{R}$.
4. Return sanitized network $\tilde{A}$.

The process of parameter learning is shown in Algorithm 2. We first initialize the latent variables l. Vertices are divided in each group with the same probability and we set the initial value of the latent variables as $l = \frac{1}{k}$ (line 1). As the model parameters are divided by $K \times K$ bundles, the privacy parameter ϵ is divided into $\epsilon_r = log\left(K^2 \cdot (e^{\epsilon} - 1) + 1\right)$ by the sequential composition property of differential privacy (line 2). For computing the expected sufficient statistics of each edge bundles,we need to compute the sufficient statistic of each edge in r (line5-7). To satisfy the differential privacy, we need to add Laplacian noise to $\bar{s}_r$. The global sensitivity of $\bar{s}_r$ is denoted as $\Delta\bar{s}_r = \frac{2}{p_r} s\left(A_0^r\right)$ by using (10) (line8-9). Based on the conjugate property of exponential family, we update the hyperparameter ν_r' by $\tilde{\bar{s}}_r$ and then compute the posterior distribution of the parameters (θ, π) (line10-11). To compute the latent variables, we need to compute the expected natural parameters $\bar{n}_r$ (line 12). As it is unnecessary to look at the data A directly when computing ν_r' and $\bar{n}_r$, we only need to add noise to the result of computing $\tilde{\bar{s}}_r$. Suppose we update ν until convergence takes $O(J)$ time, all the iterative process takes $O\left(JK^2\right)$. When computing the latent variable l_{i,z_i} of each vertex, vertex i must sum over its connected vertices in r. Then we get l_{i,z_i} in exponential family(line14-16). Moreover, we do not need to add noise to l_{i,z_i} directly. The reason is that it is unnecessary to output l_{i,z_i} most of the time. It will take $O(JN)$ time for all N vertices for all the iterative process.

Algorithm 2 The Process of Parameter Learning

Input: Input network A, group number K, privacy parameter ϵ

Output:Sanitized model parameters (θ, π)

1. Initialize $l = \frac{1}{K}$.
2. Divide privacy parameter ϵ into $\epsilon_1, \epsilon_2, \ldots, \epsilon_{K^2}$, where $\epsilon_r = log\left(K^2 \cdot (e^{\epsilon} - 1) + 1\right)$.
3. **repeat**
4. **for** $r = 1, 2, \ldots, K^2$ **do**
5. Compute the edge number p_r in bundle r.
6. Compute the sufficient statistic $S\left(A_{ij}\right)$ of each edge in r.
7. Compute the expected sufficient statistic $\bar{s}_r = \frac{1}{p_r} \sum_{ij} \sum_{(z_i,z_j)=r} s\left(A_{ij}\right) \cdot l_{i,z_i} \cdot l_{j,z_j}$.
8. $A_0^r \leftarrow$ maximum edge in r, $\Delta\bar{s}_r = \frac{2}{p_r} s\left(A_0^r\right)$.
9. Add Laplacian noise to $\bar{s}_r$ and get $\tilde{\bar{s}}_r = \bar{s}_r + Y_r$, where $Y_r \sim Lap\left(\frac{\Delta\bar{s}}{\epsilon}\right)$.
10. Update the hyperparameter $\nu_r' = \nu_r + \tilde{\bar{s}}_r$.
11. Output the posterior distribution $q(\theta, \pi)$.
12. Compute the expected natural parameter $\bar{n}_r = \left\langle n_{r(\theta,\pi)} \right\rangle_{q(\theta,\pi)}$.
13. **end for**
14. **for** $i = 1, 2, \ldots, N$ **do**
15. Compute $l_{i,z_i} \propto exp\left(\sum_r \sum_{(z_i,z_j)=r} \sum_j s\left(A_{ij}\right) \cdot l_{i,z_i} \cdot \bar{n}_r\right)$.
16. **end for**
17. **until** l, ν converge.
18. **return** θ, π.

C. PRIVACY ANALYSIS OF VB-WNDP

Taken with the sequential composition theorem of differential privacy, we can prove that VB-WNDP ensures ϵ-differential privacy.

Theorem 5: VB-WNDP satisfies ϵ-differential privacy.

Proof: Suppose the hyperparameter ν and the latent variable l converge taking J times. In Algorithm 2, the iterative process of each bundle satisfies ϵ_{iter} -differential privacy. Based on Theorem 4, each iterative process satisfies $log\left(1 + \frac{1}{K \times K}\left((e^{\epsilon_{iter}}) - 1\right)\right)$-differential privacy, all the iterative process satisfies $J \cdot log\left(1 + \frac{1}{K \times K}\left((e^{\epsilon_{iter}}) - 1\right)\right)$-differential privacy. After getting the model parameters, it does not consume any privacy budget when we output the synthetic network. Hence, VB-WNDP satisfies ϵ-differential privacy, and $\epsilon = J \cdot log\left(1 + \frac{1}{K \times K}\left((e^{\epsilon_{iter}}) - 1\right)\right)$. □

TABLE 1. The statistics of adjnoun, football and karate.

Dataset	Adjnoun	Football	Karate
N	112	115	34
E	425	613	78
C	2	2	2

N, the number of nodes; E, the number of edges; C, the number of communities.

TABLE 2. The statistics of bison, macaque and residence.

Dataset	Bison	Macaque	Residence
N	26	62	217
E	314	1187	2672
W	1-20	1-10	1-5
F	Directed	Directed	Directed

N, the number of nodes; E, the number of edges; W, the weights range; F, the network format.

VI. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of the algorithm we propose on several real-world networks. As differential privacy need to produce random noise, we measure the accuracy of the result by the median relative error where we run the Laplace mechanism for 10 times.

A. DATASETS

(1) We evaluate the utility of SSN algorithm over three real-life datasets, namely Adjnoun, Football and Karate. Adjnoun contains the network of common adjective and noun adjacencies for the novel "David Copperfield" by Charles Dickens. Nodes represent the most commonly occurring adjectives and nouns in the book and edges connect any pair of words that occur in adjacent position in the text of the book. Football contains the network of American football games between Division IA colleges during regular season Fall 2000. The nodes have values that indicate to which conferences they belong. Karate contains the network of friendships between the 34 members of a karate club at a US university.The statistics of these data is shown in Table 1.

(2) We evaluate the utility of VB-WNDP algorithm over three real-life weighted network datasets, namely Bison [28], Macaque [29] and Residence hall [30]. Bison describes the usual aggressive behaviors (fighting, nod-threats, broadside threats, head-on threats, rush threats and supplanting) were recorded among 26 males in a herd of American bison. Observations were recorded for 12 hours per day from July 25 through August 14, 1972 on the National Bison Range in Moiese, Montana. A node represents a bison and an edge represents dominance of the left bison over the right bison. Macaques records dominance of the row animal over the column animal in a colony of 62 adult female Japanese macaques (Macaca fuscata fuscata). They are known as the "Arashiyama B group". Records were made during the non-mating season, April to early October, 1976. A node represents a macaque and a directed edge A to B represents dominance of macaque A over macaque B. Residence hall collects friendship data among the 217 residents living at a residence hall located on the Australian National University campus. A node represents a person and an edge represents the friendship. The statistics of these data is shown in Table 2.

B. EVALUATION OF SSN

To show the utility of SSN algorithm,we compare the Normalized Mutual Information (NMI). NMI is a kind of measure to score the accuracy of community detection. The NMI is

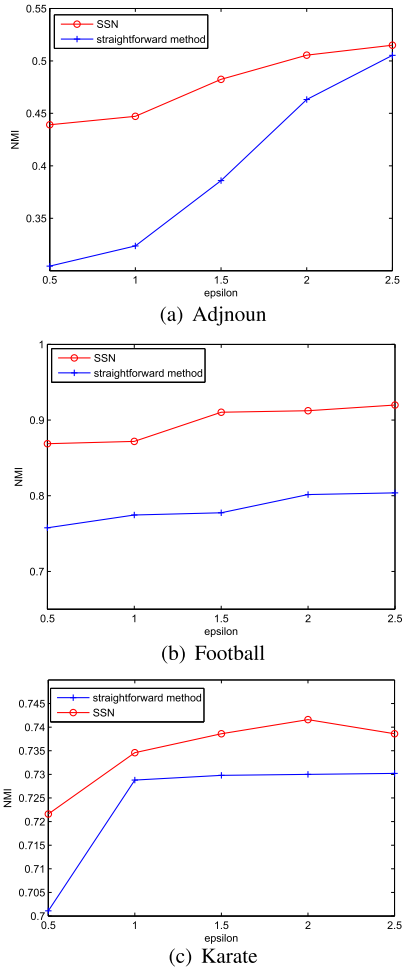


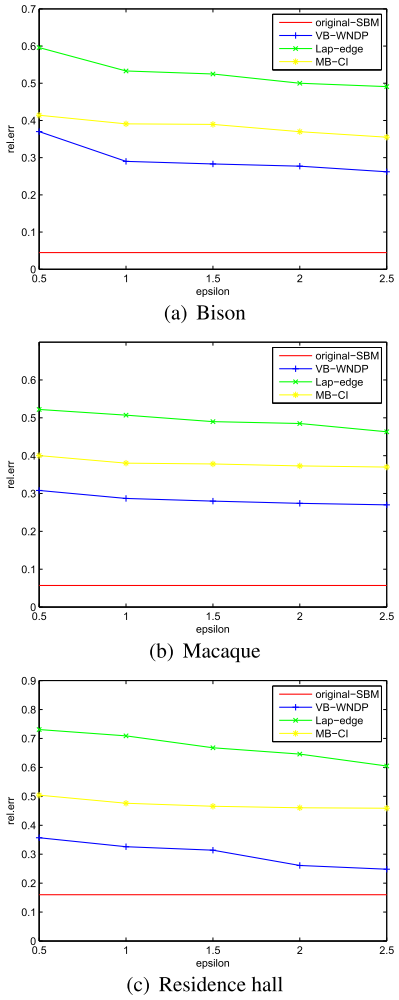
FIGURE 2. Normalized mutual information.

represented as:

$$NMI\ (X, Y) = \frac{2I\ (X, Y)}{H\ (X) + H\ (Y)} \tag{11}$$

where $H\ (X)$ is the Shannon entropy of X, and $I\ (X, Y) = H\ (X) - H\ (X \mid Y)$. It takes a value close to one if the assignments are identical and zero if they are uncorrelated. We compare with a straightforward method which adding noise to the model parameters $\{\theta, \pi\}$ directly during the iterative process of SBM. We allocate the privacy budget ϵ as follows: 0.5,1.0,1.5,2.0 and 2.5.

From Fig. 2, we can see that SSN outperforms the straightforward method. When privacy budget is relatively large, its NMI always stays high. With the increase of ϵ, the NMI rises.

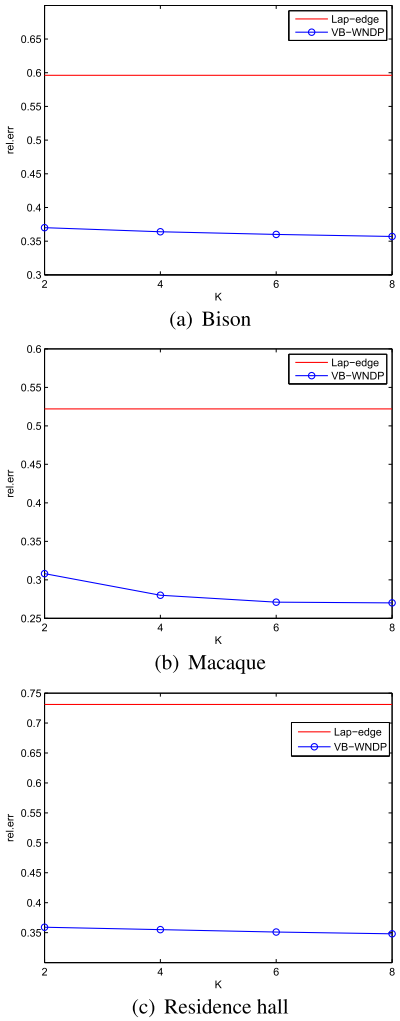


(a) Bison

(b) Macaque

(c) Residence hall

FIGURE 3. Relative error of edge weights on different ϵ.



(a) Bison

(b) Macaque

(c) Residence hall

FIGURE 4. Relative error of edge weights on different K.

As ϵ increases, the scale of noise reduces. From Figs. 2(a) and 2(b), we can also see NMI increases gradually with increases. However in Figs. 2(c), we can see NMI increases massively with increases. The reason is that the scale of Karate is smaller than Adjnoun and Football and it leads to a greater influence on NMI when we add more noise.

C. EVALUATION OF VB-WNDP

To show the utility of VB-WNDP algorithm, we compare the relative error of edge weights and evaluate the relative error for VB-WNDP under the group number K.. We represent the relative error of edge weights as:

$$rel.err = \frac{1}{m}\sum_{ij}\left|A_{ij} - \tilde{A}_{ij}\right|. \tag{12}$$

where $\tilde{A}_{ij}$ is the differentially private output and m is the number of edges.

In Fig. 3, we evaluate the relative error of edge weights, comparing with two methods. The main idea of the first one method is adding noise to the edge weights directly, we named it Lap-edge. Another method is MB-CI. What's more, we use the method which generating synthesis networks by SBM without differential privacy as a base. We allocate the privacy budget ϵ as follows: 0.5,1.0,1.5,2.0 and 2.5.

From Fig. 3, we can see that VB-WNDP outperforms both Lap-edge and MB-CI. When privacy budget is relatively large, its relative error always stays low. With the increase of ϵ, the relative error increases. As ϵ increases, the scale of noise reduces.

In Fig. 4, we show how the group number K affects the output of the relative error of edge weights. In this experiment, we set the privacy budget ϵ to be 0.5, and we use the method which adding noise to the edge weights directly as a base. We set the group number K as follows: 2,4,6 and 8. Form Fig. 4, we can see, adding noise to the edge weights directly generates poor results. With the increase of K, the relative error decreases. When K increases, the level we divide the group is higher. As each group-group partition has independent parameters, the noise we need to add is independent to each other. By dividing the vertices into a greater group number, the parameters are more accurate.

VII. CONCLUSION

In this paper, we investigate the problem of differential privacy for weighted network. We observe that the structural role of the node affects the topological structure of the network and it is necessary to take it into account in differential privacy for weighted network. We introduce a differential privacy algorithm for stochastic block model named SSN to solve the problem. By leveraging such technique,we also design a differential privacy method for weighted network named VB-WNDP. It can improve the utility of the method which adding noise to edges directly. In particular, VB-WNDP establishes a probability model of weighted network through Variational Bayes. To improve the accuracy, we add noise to sufficient statistic instead of the model parameters during iteration process. Privacy analysis and the results of extensive experiments on real datasets show that our algorithm can achieve a high data utility.

REFERENCES

[1] L. Sweeney, "k -anonymity: A model for protecting privacy," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, 2002.
[2] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity," in *Proc. 22nd Int. Conf. Data Eng. (ICDE)*, 2006, p. 24.
[3] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Social Netw.*, vol. 5, no. 2, pp. 109–137, Jun. 1983.
[4] Y. J. Wang and G. Y. Wong, "Stochastic blockmodels for directed graphs," *J. Amer. Stat. Assoc.*, vol. 82, no. 397, pp. 8–19, Mar. 1987.
[5] C. Aicher, A. Z. Jacobs, and A. Clauset, "Learning latent block structure in weighted networks," *J. Complex Netw.*, vol. 3, no. 2, pp. 221–248, Jun. 2015.
[6] C. Dwork, F. Mcsherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography* (Lecture Notes in Computer Science book), vol. 3876. 2006, pp. 265–284.
[7] M. Hay, V. Rastogi, G. Miklau, and D. Suciu, "Boosting the accuracy of differentially private histograms through consistency," *Proc. VLDB Endowment*, vol. 3, nos. 1–2, pp. 1021–1032, Sep. 2010.
[8] V. Karwa, S. Raskhodnikova, A. Smith, and G. Yaroslavtsev, "Private analysis of graph structure," *Proc. VLDB Endowment*, vol. 4, pp. 1146–1157, Aug. 2011.
[9] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, "Private release of graph statistics using ladder functions," in *Proc. ACM SIGMOD Int. Conf. Manage. Data SIGMOD*, 2015, pp. 731–745.
[10] X. Cheng, S. Su, S. Xu, L. Xiong, K. Xiao, and M. Zhao, "A two-phase algorithm for differentially private frequent subgraph mining," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 8, pp. 1411–1425, Aug. 2018.
[11] X. Ding, X. Zhang, Z. Bao, and H. Jin, "Privacy-preserving triangle counting in large graphs," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 1283–1292.
[12] H. Sun, X. Xiao, I. Khalil, Y. Yang, Z. Qin, H. Wang, and T. Yu, "Analyzing subgraph statistics from extended local views with decentralized differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2019, pp. 703–717.
[13] A. Sala, X. Zhao, C. Wilson, H. Zheng, and B. Y. Zhao, "Sharing graphs using differentially private graph models," in *Proc. ACM SIGCOMM Conf. Internet Meas. Conf. IMC*, 2011, pp. 81–98.
[14] D. Mir and R. N. Wright, "A differentially private estimator for the stochastic kronecker graph model," in *Proc. Joint EDBT/ICDT Workshops EDBT-ICDT*, 2012, pp. 167–176.
[15] Q. Xiao, R. Chen, and K.-L. Tan, "Differentially private network data release via structural inference," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining KDD*, 2014, pp. 911–920.
[16] Z. Qin, T. Yu, Y. Yang, I. Khalil, X. Xiao, and K. Ren, "Generating synthetic decentralized social graphs with local differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 425–438.
[17] X. Chen, S. Mauw, and Y. Ramírez-Cruz, "Publishing community-preserving attributed social graphs with a differential privacy guarantee," 2019, *arXiv:1909.00280*. [Online]. Available: http://arxiv.org/abs/1909.00280
[18] Q. Liu, G. Wang, F. Li, S. Yang, and J. Wu, "Preserving privacy with probabilistic indistinguishability in weighted social networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 5, pp. 1417–1429, May 2017.
[19] M. E. Skarkala, M. Maragoudakis, S. Gritzalis, L. Mitrou, H. Toivonen, and P. Moen, "Privacy preservation by k-Anonymization of weighted social networks," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Aug. 2012, pp. 423–428.
[20] K. Chen, H. Zhang, B. Wang, and X. Yang, "Protecting sensitive labels in weighted social networks," in *Proc. 10th Web Inf. Syst. Appl. Conf.*, Nov. 2013, pp. 221–226.
[21] M. Liu, Y. Zeng, Z. Jiang, Z. Liu, and J. Ma, "Centrality based privacy preserving for weighted social networks," in *Proc. 13th Int. Conf. Comput. Intell. Secur. (CIS)*, Dec. 2017, pp. 574–577.
[22] X. Li, J. Yang, Z. Sun, and J. Zhang, "Differential privacy for edge weights in social networks," *Secur. Commun. Netw.*, vol. 2017, pp. 1–10, Mar. 2017.
[23] D. Wang and S. Long, "Boosting the accuracy of differentially private in weighted social networks," *Multimedia Tools Appl.*, vol. 78, no. 24, pp. 34801–34817, Dec. 2019.
[24] Q. Qian, Z. Li, P. Zhao, W. Chen, H. Yin, and L. Zhao, "Publishing graph node strength histogram with edge differential privacy," in *Database Systems for Advanced Applications* (Lecture Notes in Computer Science), vol. 10828. 2018, pp. 75–91.
[25] M.-F. Balcan and A. Blum, "Mechanism design via machine learning," in *Proc. 46th Annu. IEEE Symp. Found. Comput. Sci. (FOCS)*, Oct. 2005, pp. 605–614.
[26] F. Mcsherry, "Privacy integrated queries: An extensible platform for privacy-preserving data analysis," *Commun. ACM*, vol. 53, no. 9, pp. 89–97, 2010.
[27] N. Li, W. Qardaji, and D. Su, "On sampling, anonymization, and differential privacy or,k-anonymization meets differential privacy," in *Proc. 7th ACM Symp. Inf., Comput. Commun. Secur. ASIACCS*, 2012, pp. 32–33.
[28] D. F. Lott, "Dominance relations and breeding rate in mature male American bison," *Zeitschrift Für Tierpsychologie*, vol. 49, no. 4, pp. 418–432, 1979.
[29] Y. Takahata, "Diachronic changes in the dominance relations of adult female Japanese monkeys of the Arashiyama B group," *The Monkeys of Arashiyama. State University of New York Press, Albany*, 1991, pp. 123–139.
[30] L. C. Freeman, C. M. Webster, and D. M. Kirke, "Exploring social structure using dynamic three-dimensional color images," *Social Netw.*, vol. 20, no. 2, pp. 109–118, 1998.

YUYE WANG is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Harbin Engineering University, China. His research interests include social networks, machine learning, and privacy preservation.

JING YANG received the Ph.D. degree from the College of Computer Science and Technology, Harbin Engineering University, China. She is currently a Professor with the College of Computer Science and Technology, Harbin Engineering University. Her main research interests include database, data mining, privacy preservation, sentiment analysis, and social computing.

JIANPEI ZHANG is currently a Professor and a Ph.D. supervisor with Harbin Engineering University. His main research interests include data mining and artificial intelligence.