Publishing Node Strength Distribution With Node Differential Privacy

GANGHONG LIU, XUEBIN MA, AND WUYUNGERILE LI
Inner Mongolia Key Laboratory of Wireless Networking and Mobile Computing, Inner Mongolia University, Hohhot 010000, China

Corresponding author: Xuebin Ma (csmaxuebin@imu.edu.cn)

This work was supported in part by the Inner Mongolia Natural Science Foundation under Grant 2018MS06026, and in part by the Science and Technology Program of Inner Mongolia Autonomous Region under Grant 2019GG116.

ABSTRACT

The challenge of graph data publishing under node-differential privacy mainly comes from the high sensitivity of the query. Compared with edge-differential privacy that can only protect the relationship between people, node-differential privacy can protect both the relationship between people and personal information. Therefore, Node-differential privacy must pay attention to the protection of personal information. This paper studies the release of node strength distribution under node-differential privacy by reducing sensitivity. We propose two algorithms to publish node strength distribution: Alternate-histogram (ALT-histogram) and Density-histogram (DEN-histogram). Experimental results show that compared with the existing node strength histogram publishing algorithm, our proposed algorithm has advantages in L1-error and KS-distance, make the noise node strength distribution closer to that of the original graph. We also propose an introspective analysis to understand the influence of the projection algorithm on the node strength distribution, experiments to prove that the projection algorithm plays an important role in the node strength distribution.

INDEX TERMS
Differential privacy, private graph publishing, node strength.

I. INTRODUCTION

As a general data structure, the graph can not only directly describe the relationship between things but also reflect various statistical characteristics. Many privacy data in life can be abstracted into a graph, such as social networks, recommendation systems, communication mode. However, in the case of graph anonymity [1], [2], the direct publication of graph data still has a high probability of disclosing personal sensitive information. Therefore, how to protect the privacy of sensitive personal information in the graph has gradually attracted the attention of researchers. Reference [3] is the first paper that proposes to study the distribution of node-differential privacy. Since then, the problem of publishing information on a graph under differential privacy [4], [5] has been studied by researchers. The difference from other privacy protection technologies is that differential privacy does not rely on the attacker’s specific background knowledge and can provide quantifiable privacy guarantees. An algorithm that satisfies differential privacy guarantees that the output distribution does not change significantly from one dataset to a neighboring dataset.

Node-differential privacy [10]–[14] and edge-differential privacy [3], [6]–[9], [15] are variants of differential privacy applied to the graph. In edge-differential privacy, two neighboring graphs differ by a single edge, while in node-differential privacy, two neighboring graphs differ by one node and all edges connected with this node. For a graph $G = (V, E)$ with the number of nodes $|V| = n$ (where $V$ is the set of all nodes and $E$ is the set of all edges), deleting one edge only affects the degree change of two nodes on this edge, while deleting one node will cause the $n - 1$ edge to be deleted in the worst case. Therefore, node-differential privacy in graph data is more difficult to satisfy than edge-differential privacy, but it can provide higher privacy protection.

Most of the existing differential privacy research on graphs [3], [9], [11], [14], [15] are mostly focused on the degree of research on unweight graphs [3], [11], [14] but ignore the study of weighted graphs. However, as [15] said, the importance of nodes in a weighted graph does not entirely depend on the degree distribution of the weighted graph. For example, Bitcoin OTC [20] is a graph of “who trusts who” composed of people who use Bitcoin for transactions on the platform.
In this graph, the node degree cannot represent the user’s reputation, but can only represent the number of user transactions. At this time, the node strength can be used to represent the user’s reputation, and the node strength is the sum of the weights of all edges connected to the node. This paper studies the publishing of the histogram of weighted graph node strength under node-differential privacy. Given a graph \( G = (V, E, W) \), the goal is to publish a histogram of node strength while satisfying node-differential privacy, as close as possible to the true distribution of \( G \). The main contributions of this paper are as follows: (1) as far as we know, the issue of node strength histogram on the weighted graph under node differential privacy is studied for the first time. (2) Based on the sensitivity determined by the projection algorithm, we propose two algorithms to publish node strength histogram under node-differential privacy: ALT-histogram and DEN-histogram, and prove that the two node strength distribution publishing mechanisms meet the definition of node-differential privacy. (3) A large number of experiments are carried out on four real datasets, and the proposed algorithm is compared with other existing algorithms. The experimental results show that compared with the baseline algorithm, the proposed algorithm has obvious improvement. We also conducted an introspective analysis to examine the influence of the projection algorithm.

The rest of this paper is organized as follows. In Sect. 2, we discuss the related work. In Sect. 3, we define the problem and discuss the differential privacy and its application on graphs. In Sect. 4, we introduce our algorithm. The experimental results are given in In Sect. 5, and the conclusion in Sect. 6.

II. RELATED WORK

The application of differential privacy to graph data has been extensively studied. Kasiviswanathan et al. [10] used differential privacy technology to study graph data for the first time. They showed how to estimate the number of triangles in social networks satisfying edge-differential privacy and showed how to calibrate the noise of subgraph count accordingly. Day et al. [14] proposed two algorithms to publish the degree distribution of graphs under node-differential privacy. The \( \theta \)-Cumulativehistogram algorithm first transforms the histogram into a cumulative histogram, then adds noise, and finally extracts the noisy histogram. The advantage of this algorithm is that it can significantly reduce the sensitivity, but it is only suitable for the degree distribution. If it is extended to the node strength distribution, the sensitivity is too high. The \( (\theta, \mathcal{E}) \)-histogram algorithm proposed a geometric series group algorithm according to the properties of social networks. This algorithm cannot adapt to different datasets because of its fixed grouping. Ding et al. [16] proposed a node-differential privacy algorithm for each node 3-subgraph counting is used for protection, and the algorithm was extended to the local clustering coefficient; Qian et al. [15] proposed two algorithms to publish the node strength histogram of graphs under edge-differential privacy for the first time, in which the node strength histogram publishing algorithm based on sequence awareness has high accuracy but high time overhead, and the privacy protection provided by edge-differential privacy is weaker than that of node-differential privacy.

A key technique to satisfy node-differential privacy is that of “projection”, which transforms a graph to be \( \theta \)-degree-bounded to reduce sensitivity. Kasiviswanathan et al. [10] proposed to reduce the sensitivity by “truncation” projection, which is to delete all nodes whose node degree is greater than the threshold value. Although this algorithm can reduce the sensitivity, it can delete many unnecessary edges and lose a lot of effective information in the original graph. Blocki et al. [12] proposed a projection algorithm of “edge removal” to reduce the sensitivity, that is to select any order of edges, traverse the edges in this order, and then delete the edges in the nodes whose node degree is greater than the threshold. Compared with the “truncation” algorithm, this algorithm retains more effective information, but it is still not ideal. Chen and Zhou [13] based on the exponential mechanism, used the flow graph algorithm to map the degree distribution of the graph to a histogram. The disadvantage of this algorithm is that the sensitivity is too high, which is far greater than the first two algorithms. Day et al. [14] proposed a projection algorithm of “edge adding”. Firstly, the edge set of the graph is set to be empty, and the edges are traversed according to the stable order of the edges. If the node degrees of the nodes at both ends of the edge are less than the threshold value, they are added, otherwise, they are not added. Because of the uncertainty of edge order, this algorithm still cannot keep the edge information of the original graph to the greatest extent. Qian et al. [15] proposed an algorithm to project the weight of the original graph. This algorithm traversed each edge and projects the edge whose weight is greater than the threshold value so that the weight of the edge is the threshold value. This algorithm is the first weight projection algorithm.

To solve the problem of node strength publishing under differential privacy, the existing study [10]–[12] reduces the sensitivity by projection, but it loses a lot of effective information in the original graph. Moreover, the research on node strength mainly focuses on edge-difference privacy, and there is no research on node strength publication in differential privacy. As mentioned above, node-differential privacy provides higher protection than edge-differential privacy. In the structure of the graph, a node usually represents a person, and node-differential privacy cannot only protect the relationship between people but also protect personal information. Therefore, it is necessary to realize node-differential privacy to protect user data.

III. PRELIMINARIES

A. NODE STRENGTH

A weighted and finite graph \( G = (V, E, W) \) is defined by a set of nodes \( V = \{v_1, v_2, \ldots, v_n\} \), a set of edges \( E = \{e_{ij} | i, j \in V\} \), and \( W = \{w_{ij} | w_{ij} = value(e_{ij})\} \), where \( value(e_{ij})\)
represents the connection weight of the edge $e_{ij}$. In this paper, for simplicity of presentation, we assume that $n = |V|$, the number of nodes of the input graph $G$, is publicly known.

We generally extend the degree of a node $v_i$ to the sum of weights of edges that connect $v_i$, denoted by node strength, which is given as follows:

$$s_i = \sum_{j \in \Gamma(i)} w_{ij}$$  \hspace{1cm} (1)

where $\Gamma(i)$ is the set of adjacent nodes of node $v_i$.

**B. DIFFERENTIAL PRIVACY**

The notion of $\varepsilon$-differential Privacy [4] is defined based on the concept of neighboring databases. Two datasets $G$ and $G'$ are defined as neighboring datasets if they only differ in one record.

**Definition 1 ($\varepsilon$-Differential Privacy)**. A randomized algorithm $A$ satisfies $\varepsilon$-differential privacy when for any two neighboring databases, $D$ and $D'$, denoted by $D \simeq D'$,

$$Pr[A(D) \in S] \leq \exp(\varepsilon) \times Pr[A(D') \in S]$$  \hspace{1cm} (2)

where $S \subseteq \text{range}(A)$ and $\varepsilon$ is a parameter for privacy level.

**Definition 2 (Global Sensitivity)** For a query $f : G \rightarrow \mathbb{R}^d$, and neighboring databases $G$ and $G'$, the $L_1$-global sensitivity of $f$ is defined as:

$$\Delta f = \max_{D \simeq D'} ||f(D) - f(D')||_1$$  \hspace{1cm} (3)

where the $L_1$ distance $||x||_1$ is the sum of the absolute values of each element of the vector $x$.

Laplace Mechanism [4] For a query $f : G \rightarrow \mathbb{R}^d$ over a database $G$, we use Laplace mechanism to satisfy $\varepsilon$-differential privacy, noise $\text{Lap}(\Delta f / \varepsilon)$ is added to the output of the query.

$$A(G) = f(G) + \text{Lap}(\Delta f / \varepsilon)^d$$ \hspace{1cm} (4)

and $Pr[\text{Lap}(b) = x] = \frac{1}{b} \exp(-|x|/b)$

Exponential Mechanism [17] Given a quality score $u(G, H_i)$ for outputting $H_i$ on input $G$ and a randomized algorithm $A$, we have

$$A(G, H_i) = \left\{ H_i : | Pr[H_i \in \mathcal{H}] \propto \exp \left( \frac{\varepsilon u(G, H_i)}{2\Delta u} \right) \right\}$$  \hspace{1cm} (5)

where $\Delta u = \max_{H_i \in \mathcal{H}, G'} |u(G, H_i) - u(G', H_i)|$ is the global sensitivity of the quality function. The sampling probability for each $H_i \in \mathcal{H}$ is determined based on a user specified quality function $u$.

**C. COMPOSITION PROPERTIES**

Differential privacy satisfies sequential composition and transformation invariance [18], [19]. In short, sequential composition guarantees that for any sequence of computations $A_1, A_2, \ldots, A_k$, if each $A_i$ satisfies $\varepsilon$-differential privacy, then publishing the results of all of $A_1, A_2, \ldots, A_k$ satisfies $\sum_{i=1}^{k} \varepsilon_i$-differential privacy. The transformation invariance property says that given $A_1$ that satisfies $\varepsilon$-differential privacy and any algorithm $A_2$, the new algorithm $A(\cdot) = A_2(A_1(\cdot))$ satisfies $\varepsilon$-differential privacy. In other words, performing post-processing on the output of an algorithm that satisfies differential privacy does not affect the privacy guarantee.

**D. UTILITY METRICS**

To evaluate the performance of different node strength distribution algorithms, the L1 error used in [10] and the KS-distance used in [3] are used as quantitative indicators.

The L1 error is used to calculate the cumulative error between the strength histogram distributions of two nodes. For two strength distributions $D$ and $D'$ with length $M$, the L1 error between them can be obtained by $L_1 = \sum_{i=0}^{M-1} |D_i - D'_i|$, while for the strength distribution whose length is smaller than $M$, it is filled with 0 to $M$ for convenience.

The KS-distance is used to reflect the proximity between the strength histograms of two nodes. For node strength distributions $D$ and $D'$, the KS-distance between them is defined as $KS(D, D') = \max(|CDF_D(i) - CDF_{D'}(i)|)$, where $CDF_D(i)$ is the cumulative distribution function value of node strength $i$ on distribution $D$.

**IV. PROPOSED ALGORITHM**

**A. PROPOSED PROJECTION ALGORITHM**

Although the current graph projection algorithm reduces the sensitivity, it loses a lot of important information, and the existing algorithms can only be divided into the projection of node degree and the projection of edge weights, and cannot project both at the same time. In the algorithm of node strength distribution under node-differential privacy, both node degree and edge weight are directly related to node strength. Therefore, to project node degree and edge weight at the same time, we propose a new projection algorithm, which can constrain node degree and edge weight, and solve the problems of the high sensitivity of node-differential privacy and sparse distribution of real node strength. As shown in Algorithm 1, our algorithm requires all edges in the input graph $G$ to be sorted from smallest to largest weight, represented by $\Lambda(G)$.

The projection algorithm first constructs a graph in which the nodes in the graph are all nodes in $G$, but there are no edges, and then traverses the edges according to the $\Lambda(G)$ order. If the degree of the nodes at both ends of the edge is less than the degree threshold and the edge weight is less than or equal to the weight threshold, it is added to the edge set. If the degree of the nodes at both ends of the edge is less than the degree threshold and the edge weight is greater than the weight threshold, the edge weight is set as the weight threshold and then added to the edge set. If there is a node at both ends of the edge whose degree is greater than or equal to the degree threshold, it is not added.

According to the projected graph, we can get the node strength histogram $\text{hist}(G[1.0]) = \{h_1, h_2, h_3, \ldots, h_k\}$ represents the number of nodes whose strength is $i$. According to
Algorithm 1 \text{Proj}(\theta, \lambda) Projection by Strength-Bounded

\textbf{Input}: An input graph \( G = (V, E, W) \),
and an edge ordering \( \Lambda(G) = < e_1, e_2, e_3, \ldots, e_n > \),
a degree bound \( \theta \), a weight bound \( \lambda \).

\textbf{Output}: An output \((\theta, \lambda)\) - bounded graph \( G(\theta, \lambda) \).

1: \( E^0 \leftarrow \emptyset; d(v) \leftarrow 0 \) for each \( v \in V \)
2: \textbf{for} \( e \leftarrow (u, v) \in \Lambda \) \textbf{do}
3: \textbf{if} \( d(u) < \theta \) \& \& \( d(v) < \theta \) \& \& \( w(u, v) \leq \lambda \) \textbf{then}
4: \( E^0 \leftarrow E^0 \cup \{ e \}; d(u) \leftarrow d(u) + 1; d(v) \leftarrow d(v) + 1 \)
5: \textbf{end}
6: \textbf{if} \( d(u) < \theta \) \& \& \( d(v) < \theta \) \& \& \( w(u, v) > \lambda \) \textbf{then}
7: \( E^0 \leftarrow E^0 \cup \{ e \}; d(u) \leftarrow d(u) + 1 \)
8: \( d(v) \leftarrow d(v) + 1; w(u, v) \leftarrow \lambda \)
9: \textbf{end}
10: \textbf{end}
11: \textbf{return} \( G(\theta, \lambda) = (V, E^0, W^\lambda) \)

the global sensitivity of \( \text{hist}(G(\theta, \lambda)) \) released by lemma1, \( \Delta \text{hist} \) is \( 2\theta + 1 \).

\textbf{Lemma 1:} For any \( G \simeq G \) that differ in one node, we have
\[
\left| \text{hist}(G(\theta, \lambda)) - \text{hist}(G'(\theta, \lambda)) \right| \leq 2\theta + 1
\]  

\textbf{Proof:} Assume, without loss of generality, Compared with \( G = (V, E), G' = (V', E') \) has an additional node \( v^+ \), that is, \( V' = V \cup \{ v^+ \}, E' = E \cup E^+, E^+ \) is the set of all edges connected to \( v^+ \) in \( G' \). In the worst case, \( v^+ \) is connected to \( \theta \) nodes, and the existence of \( v^+ \) and its connected edges will affect the strength of \( \theta \) nodes except \( v^+ \).

And because the strength of a node changes, it will cause two transformations in the histogram, and \( v^+ \) itself has another change, then deleting a node and its connected edges will cause \( 2\theta + 1 \) changes.

\textbf{B. PARAMETER SELECTION}

The simplest solution to meet differential privacy is to add Laplace noise directly, but this algorithm will lead to too much noise accumulated by interval queries, increase the error, and finally make the data lose its availability. One way to improve the accuracy is to apply the idea of aggregation on the histogram box to group the histogram. The smaller the grouping number \( k \), the less noise is added to the histogram, but at the same time, more aggregation errors caused by the group average instead of the real value will be introduced. Similarly, larger \( \theta \) and \( \lambda \) can retain more information, but the number of barrels in the node strength histogram will increase and more noise will be added.

Because the parameters of the algorithm will directly affect the results, choosing the appropriate parameters can reduce the error of the algorithm. In this section, it is proposed to select parameters through the exponential mechanism to weigh the projection error caused by projection and the aggregation error caused by data release. For this reason, a quality function with low sensitivity is designed to select the optimal grouping center, node degree threshold \( \theta \), and edge weight threshold \( \lambda \).

Our mass function consists of two parts. The first part captures the error caused by the projection algorithm. Because the projection algorithm changes some nodes and edges, the two buckets in the histogram will change due to the change of the node strength of one node. The node strength is determined by the degree of the node and the edge weight of the edge connected to the node. The projection error is:
\[
I_{\text{proj}}(G, \theta, \lambda, j) = 2|\{ v \in V, \text{deg}(v) > \theta \} \cup \{ e \in E(v), \text{value}(e) > \lambda \}|
\]  

where \( \text{value}(e) \) represents the weight of edge \( e \) in the graph, and \( E(v) \) contains all edges adjacent to node \( v \), and \( \text{deg}(v) \) represents the degree of node \( v \).

The second part captures the error caused by the data publishing algorithm. The aggregation error of our proposed data publishing algorithm comes from the error caused by replacing the true value of the bucket with the group average value after grouping the histogram and the error caused by adding Laplace noise. Therefore, the aggregation error can be expressed by using the formula in [14]:
\[
I_{\text{hist}}(G, \theta, \lambda, B_n) = \sum_{x \in B_n} \sum_{d \in x} \left( c_d \left( -\frac{1}{\theta} \right) + \frac{2\theta l + 1}{\epsilon \cdot |B_n|} \right)
\]  

\( c_d \) represents the true value of bin, \( B_n \) represents the result of the packet, and \( B_n \) represents the \( k \)th packet. Combining these two components, the proposed quality function is:
\[
q^\theta(G, \theta, \lambda, B_n) = -I_{\text{proj}}(G, \theta, j) - I_{\text{hist}}(G, \theta, j)
\]  

To apply the exponential mechanism to select the parameter pair \((\theta, \lambda, B)\), we need to determine the sensitivity of the mass function. lemma 2 proves that \( \Delta q \) is \( 6\theta + 4 \).

\textbf{Lemma 2:} For any \( G \simeq G \) that differ in one node, we have
\[
|q^\theta(G, \theta, \lambda, \Omega_i) - q^\theta(G', \theta, \lambda, \Omega_i)| \leq 6\theta + 4
\]  

\textbf{Proof:} Based on Lemma 1, there is a change in the strength of at most \( \theta + 1 \) nodes in both \( G(\theta, \lambda) \) and \( G'(\theta, \lambda) \), so we can get \( I_{\text{proj}}(G, \theta, \lambda) = I_{\text{proj}}(G', \theta, \lambda) \leq 2(\theta + 1) \).

According to [14], \( I_{\text{hist}}(G, \theta, \lambda, \Omega_i) = I_{\text{hist}}(G', \theta, \lambda, \Omega_i) \leq 2(2\theta + 1) \) can be obtained. Thus, we have
\[
q^\theta(G, \theta, \lambda, \Omega_i) - q^\theta(G', \theta, \lambda, \Omega_i) = \left| -I_{\text{proj}}(G, \theta, \lambda) - I_{\text{hist}}(G, \theta, \Omega_i) \right|
\]  

Finally, we can conclude that for any \( G \simeq G \) that differ in one node, we have \( \Delta q \leq 6\theta + 4 \).
C. ALT-HISTOGRAM ALGORITHM

In this part, we introduce an ALT-histogram algorithm, which randomly selects K buckets as the group center in the histogram, and then selects a bucket that is not the group center to replace the group center corresponding to the bucket in the partition result. If the error can be improved, it will be replaced; if it cannot be improved, the next bucket that is not in the group center will be selected until the group center no longer changes. Next, the real value of each bucket in the group is replaced by the average of the group. Finally, add noise to release.

Algorithm 2 ALT-Histogram Algorithm

Input: A weight graph $G = (V, E, W)$, privacy budget $\varepsilon_1$ and $\varepsilon_2$, candidates $O, T$ and $K$
Output: A histogram $H$ satisfying the differential privacy

1: for each $(\theta_i, \lambda_i, B_n) \in O \times T \times K$, compute $q^{\theta}(G, \theta_i, \lambda_i, B_n)$
2: select $(\theta_s, \lambda_s, B_s)$ with prob. $\propto \exp \left( \frac{(1 + \varepsilon^2)q^\theta(G, \theta, \lambda, B_n)}{2\Delta^2q^\theta} \right)$
3: $G^{(\lambda, \theta)} \leftarrow \text{Proj}^{(\lambda, \theta)}(G)$ by Algorithm 1
4: Choose k candidates $C \leftarrow \{c_1, \ldots, c_k\}$ randomly from the histogram $\text{hist}(G^{(\lambda, \theta)})$
5: $i \leftarrow 0$
6: while $i < k$ do
7: Associate each $h_i$ with the group center in $C$, and $\text{hist}(G^{(\lambda, \theta)})$ is divided into k set $B \leftarrow \{b_1, \ldots, b_k\}$
8: for each $h_i \in b_i$ do
9: $C' \leftarrow \{c_1, \ldots, h_i, \ldots, c_k\}$
10: if $\text{cost}(C') < \text{cost}(C)$ then
11: $C \leftarrow C'; i \leftarrow i - 1$; break
12: end if
13: end for
14: $i++$
15:end
16: Calculate average count $h_k$ for every bin in the bucket $b_k$
17: Add noise to count as $\tilde{h}_i \leftarrow h_k + \text{Lap}\left(\frac{\Delta \text{hist}_{\varepsilon_2|b_k}}{\varepsilon_2}\right)$
18: return $H$

As shown in Algorithm 2, it uses a weight graph $G$, privacy budget $\varepsilon_1$ and $\varepsilon_2$, node degree threshold candidate set $O$, edge weight threshold candidate set $T$ and group number candidate set $K$ as input. The output is a noise node strength distribution histogram satisfying differential privacy.

Firstly, k buckets are randomly selected as the group center $C = \{c_1, \ldots, c_k\}$ (line4). Then traverse the group center and use the formula in [15] to divide all the buckets into corresponding groups.

\[
\begin{align*}
\text{dist} (c_1, h_i) &= \sum_{h_{e \in \{c_1, h_i\}}} |h_j - c_1| \\
\text{dist} (h_i, c_r) &= \frac{\sum_{h_{e \in \{h_i, c_r\}}} |h_j - c_1|}{q - i}
\end{align*}
\]

where $p$ and $q$ represent the position of $c_1$ and $c_r$ in the original histogram, $c_r$ represents the group center on the right side of the bucket. If $\text{dist} (c_1, h_i) < \text{dist} (h_i, c_r)$, then we divide all buckets between $c_1$ and $h_i$ into $b_i$. Otherwise, the division between $h_i$ and $c_r$ is divided into $b_r$. The result of this step is $B = \{b_1, \ldots, b_k\}$ (line7).

 Traverse the partition result $b_i$ corresponding to the current group center $c_i$. A new group of group center $C'$ is obtained by replacing $c_i$ with a bucket $h_i$ in $b_i$, and the $\text{cost}(C')$ caused by $C'$ is calculated. If it is less than $\text{cost}(C), C'$ is used instead of $C$ (lines 8-11). Because this algorithm uses the average value of the group to replace the true value of each bucket in the group, so the error caused by grouping is represented by

\[
\text{cost} (C) = \sum_{h_{e \in b_i}} (|h_j - b_i|)_{\text{average}}
\]

where $b_i|\text{average}$ represents the average value of $b_i$. Then return to line6 to start traversing the group center again until the group center does not change (lines 12-15). Next, the real value of each bucket in the group is replaced by the average of the group. Finally, add noise to release (lines 16-18).

Lemma 3 ALT-histogram of Algorithm 2 satisfies $(\varepsilon_1 + \varepsilon_2)$-node-differential privacy

Proof: In algorithm 2, Line2 uses the exponential mechanism with privacy budget of $\varepsilon_1$. Line17 uses the Laplace mechanism with a privacy budget of $\varepsilon_2$. All other steps are independent and performed on the noisy output. By the composition theorem, the algorithm satisfies $(\varepsilon_1 + \varepsilon_2)$-differential privacy.

D. DEN-HISTOGRAM ALGORITHM

In this section, we will discuss another algorithm, which is based on the observation that the fluctuation of the group center in the adjacent bucket is small and the distance between the group center is large. The central idea is to find the high-density bucket separated by the low-density bucket. In short, if the density of the other buckets around one bucket is lower than his, and the bucket is far away from the other group centers, then this bucket is suitable for the group center.

As shown in Algorithm 3, it uses a weight graph $G$, privacy budget $\varepsilon_1$, $\varepsilon_2$ and $\varepsilon_3$, node degree threshold candidate set $O$, edge weight threshold candidate set $T$ and group number candidate set $K$ as input. The output is a noise node strength distribution histogram satisfying differential privacy.

First, calculate the distance between each bucket in the original histogram and add noise (lines 4-7).

\[
distance (i, j) = |h_i - h_j|
\]

Secondly, the local density of each bucket in the original histogram is calculated,

\[
\rho_i = \sum_{j \neq i} X((\text{distance}(i, j) - \text{disThreshold}) < 0)
\]

where $\text{disThreshold}$ represents the truncation threshold and $j$ represents other buckets, $X(x) = \frac{1}{\exp(x)}$. The meaning of this formula is to find the number of buckets whose distance from the $i$th bucket is less than the truncation distance and regard it as the local density of the $i$th bucket. (lines 8-10)
Algorithm 3 DEN-Histogram Algorithm

Input: A weight graph $G = (V, E, W)$, privacy budget $\varepsilon_1, \varepsilon_2$ and $\varepsilon_3$, candidates $O, T$ and $K$

Output: A histogram $\hat{H}$ satisfying the differential privacy

1: for each $(\theta_i, \lambda_j, B_n) \in O \times T \times K$, compute $g^\ell(G, \theta_i, \lambda_j, B_n)$
2: select $(\theta_s, \lambda_s, B_s)$ with prob. $\propto \exp \left( \frac{\varepsilon_1 g^\ell(G, \theta_s, \lambda_s, B_s)}{\varepsilon_2} \right)$
3: $G^{(\lambda, \theta)} \leftarrow \text{Proj}^{(\lambda, \theta)}(G)$ by Algorithm 1
4: for each $h_i, h_j \in H_{\text{proj}}$
5: compute the distance between $h_i$ and $h_j$ as $\text{distance}(i, j)$
6: $\text{distance} = \text{distance}(i, j) + \text{Lap}(\frac{\Delta \text{hist}}{\varepsilon_2})$
7: end for
8: for each $h_i \in H_{\text{proj}}$
9: compute the local density $\rho_i$ of $h_i$
10: end for
11: for each $h_i \in H_{\text{proj}}$
12: compute the distance from high Local density $\delta_i$
13: compute $\gamma_i = \rho_i \delta_i$
14: end for
15: select the top $k$ bin with $\gamma_i$ as the cluster center
16: Associate each $h_i$ with the group center in $C$, and $\text{hist}(G^{(\lambda, \theta)})$ is divided into $k$ set $B = \{b_1, \ldots, b_k\}$
17: Calculate average count $h_k$ for every bin in the bucket $b_k$
18: Add noise to count as $\hat{h}_i \leftarrow h_k + \text{Lap}(\frac{\Delta \text{hist}}{\varepsilon_3 \|b_k\})$
19: return $\hat{H}$

Next, calculate the distance between the bucket and the bucket with higher local density,

$$\delta_i = \min_{i, j} (\text{distance}(i, j)) \quad (15)$$

Set the points with higher local density and larger distance from the higher-density bucket as the group center, we sort them according to the size of $\gamma_i = \rho_i \delta_i$. Select the top $k$ as center groups and then divide them according to formula (14) (lines 11-16).

Finally, the real value of each bucket in the group is replaced by the average of the group, then add noise to release (lines 17-19).

It should be noted that the local density of two buckets cannot be calculated based on the real distance between them, otherwise, it will violate differential privacy, so we add Laplace noise to distance. Lemma 4 below proves that the sensitivity of the computation of $\text{distance}(i, j)$ is at most $4\theta + 2$.

Lemma 4. For any $G \simeq G'$ that differ in one node, $\text{distance}(i, j)$ represents the distance from the $i$th bucket to the $j$th bucket in the $\text{hist}(G^{(\lambda, \theta)})$, $\text{distance}'(i, j)$ represents the distance from the $i$th bucket to the $j$th bucket in the $\text{hist}(G^{(\lambda, \theta')})$, and we can get the

$$||\text{distance}(i, j) - \text{distance}'(i, j)||_1 \leq 4\theta + 2 \quad (16)$$

Proof: Assume, without loss of generality, According to Lemma 1, the existence of a node will cause at most $2\theta + 1$ changes in the histogram, so we can conclude that $|\text{hist}(i) - \text{hist}'(i)| \leq 2\theta + 1$, similarly, $|\text{hist}(j) - \text{hist}'(j)| \leq 2\theta + 1$. So, we can get

$$||\text{distance}(i, j) - \text{distance}'(i, j)||_1$$

$$= ||\text{hist}(i) - \text{hist}(j)\| - ||\text{hist}'(i) - \text{hist}'(j)||_1$$

$$\leq \left( ||\text{hist}(i) - \text{hist}(j)\| - \text{hist}'(i) - \text{hist}'(j)\| \right)$$

$$\leq \left( ||\text{hist}(i) - \text{hist}'(i)\| - \text{hist}'(j) - \text{hist}'(j)\| \right)$$

$$\leq ||\text{hist}(i) - \text{hist}'(i)\| + ||\text{hist}(j) - \text{hist}'(j)||_1$$

$$\leq 2\theta + 1 + 2\theta + 1 = 4\theta + 2$$

Lemma 5: DEN-histogram of Algorithm 3 satisfies $(\varepsilon_1 + \varepsilon_2 + \varepsilon_3)$-node-differential privacy.

Proof: In algorithm 3, Line 2 uses the exponential mechanism with privacy budget of $\varepsilon_1$. Line 6 uses the Laplace mechanism with a privacy budget of $\varepsilon_2$ to noise the distance between buckets. Line 18 uses the Laplace mechanism with a privacy budget of $\varepsilon_3$. All other steps are independent and performed on the noisy output. By the composition theorem, the algorithm satisfies $(\varepsilon_1 + \varepsilon_2 + \varepsilon_3)$-node-differential privacy.

V. EXPERIMENT

In this section, we compare the experimental results of the proposed algorithm with $(\theta, \Omega)$-histogram algorithm [14] and $\theta$-Cumulative histogram algorithm [14], and analyze how different aspects of our proposed algorithm affect the effectiveness.

A. DATASETS AND SETTINGS

Our experiment is based on four real datasets downloaded from [20]. These datasets come from different fields: (1) Wiki-vote is a Wikipedia voting network. (2) Facebook is a social network dataset where we add random weight values to each edge of an unweighted graph. (3) Bitcoin OTC trust network is a “who trusts who” network composed of people who use bitcoin to trade on the bitcoin OTC platform. This is the first explicit weighted signed directional network that can be used for research. (4) Enron dataset is an e-mail network obtained from about 1 million e-mail datasets. Table 1 describes the attributes of the datasets, such as maximum edge Weightmax and maximum node strength $n_{\text{max}}$.

| Graph      | $|V|$ | $|E|$ | Weightmax | $n_{\text{max}}$ |
|------------|------|------|-----------|------------------|
| Facebook   | 4039 | 88234| 21        | 11441            |
| Bitcoin OTC| 5881 | 35592| 21        | 3545             |
| Enron      | 36692| 183831| 10       | 7629             |
| Wiki-Vote  | 7066 | 100736| 10       | 4528             |

According to [14], [15], all experiments are performed under the condition of privacy budget $\varepsilon \in [0.1, 2.0]$. 

VOLUME 8, 2020
B. EVALUATING ALT-HISTOGRAM AND DEN-HISTOGRAM

We compare DEN-histogram and ALT-histogram with \((\theta, \Omega)\)-histogram and \(\theta\)-Cumulativehistogram, which is used to answer a given range query. And \((\theta, \Omega)\)-histogram and \(\theta\)-Cumulativehistogram are improved to publish node strength histogram under node-differential privacy.

In this section, the privacy budget of DEN-histogram is divided into \(\varepsilon_1 = 0.8\varepsilon\), \(\varepsilon_2 = 0.1\varepsilon\) and \(\varepsilon_3 = 0.1\varepsilon\), and disThreshold = 30. The privacy budget of ALT-histogram is divided into \(\varepsilon_1 = 0.8\varepsilon\), \(\varepsilon_2 = 0.2\varepsilon\). All results are the average values of 10 operations.

In Fig.1, we present two histogram publishing algorithms and \((\theta, \Omega)\)-histogram and \(\theta\)-Cumulativehistogram. The upper part is KS-distance and the lower part is an L1 error. It can be seen that: (1) in KS-distance, the effect of the \(\theta\)-Cumulativehistogram algorithm is the worst in all four data sets, followed by \((\theta, \Omega)\)-histogram. The two algorithms proposed by us are better than the comparison algorithm in KS distance.

(2) In terms of L1 error, the worst performance of the Bitcoin OTC dataset is \((\theta, \Omega)\)-histogram, followed by \(\theta\)-Cumulativehistogram. In the other three datasets, the worst performance was found in \(\theta\)-Cumulativehistogram, followed by \((\theta, \Omega)\)-histogram. On each dataset, our two algorithms are better than the comparison algorithm in the L1 error. (3) With the increase in the value of \(\varepsilon\), we can see that the L1 error and KS-distance tend to decrease.

Although the added noise is random, to a large extent, \(\varepsilon\) can ensure the magnitude of injected histogram noise. The smaller the \(\varepsilon\) is, the more noise is injected, and the more effective the data security protection is, but at the same time, the accuracy and availability of data are lost. (4) Compared with the DEN-histogram, ALT-histogram can get a relatively accurate node strength histogram, which is mainly because the DEN-histogram adds noise when calculating the distance between buckets, which leads to the selection of group center is not the most accurate. Although the initial group center of the ALT-histogram is generated by random selection, we reduce the randomness by iteration and improve the accuracy of the ALT-histogram. In general, both ALT-histogram and DEN-histogram can publish high-precision node strength histogram under node-differential privacy.

Next, we show the complementary cumulative distribution function (CCDF) of the node strength distribution and the real node strength distribution issued by the four algorithms in the case of privacy budget \(\varepsilon = 1\). It can be seen from Fig.2 that: (1) compared with the comparison algorithm, the distribution of ALT-histogram and DEN-histogram follows the real distribution, which shows that the distribution of ALT-histogram and DRN-based histogram is more similar to the real distribution.

(2) Among the two algorithms of ALT-histogram and DEN-histogram, ALT-histogram performs best. In contrast, the fluctuation of the DEN-histogram is more obvious. This is mainly due to the noise added to the distance between buckets when calculating density, which reduces the accuracy of the select group center. (3) all four algorithms have obvious
critical points because all four algorithms carry out the projection process, which restricts the maximum strength of nodes. (4) the performance of $\theta$-CumulativeHistogram is the worst and the distance from the original distribution is the furthest. This is mainly because the sensitivity of this algorithm is so high that it is impossible to select more suitable parameters when select parameters through the exponential mechanism.
C. INTRUSIVE ANALYSIS

In this section, we perform an introspective analysis to see how our proposed projection algorithm affects the performance of ALT-histogram and DEN-histogram. Noproj-ALT-histogram is the ALT-histogram algorithm without projection, and Noproj-DEN-histogram is the DEN-histogram algorithm without projection. Fig. 3 shows the results. The upper part is KS distance and the lower part is an L1 error. It can be seen that, in terms of L1 error or KS distance, compared with Noproj-ALT-histogram and Noproj-DEN-histogram, the proposed algorithm has advantages in L1-error and KS-distance, which makes the noisy node strength distribution closer to the node strength distribution of the original graph. In the future, we plan to extend the algorithm to serve the information of the graph as much as possible by limiting the weight and degree and restricting the sensitivity of privacy. First, we propose a projection algorithm that preserves the information of the graph as much as possible by limiting the weight and degree and restricts the sensitivity of the node intensity histogram to be published on the projection graph. Then, we proposed two algorithms: ALT-histogram and DEN-histogram and proved that they satisfy differential privacy. The experimental results show that, compared with the existing node strength histogram publishing algorithms, the proposed algorithm has advantages in L1-error and KS-distance, which makes the noisy node strength distribution closer to the node strength distribution of the original graph. In the future, we plan to extend the algorithm to triangle coefficients and local clustering coefficients.

VI. CONCLUSION

This paper focuses on the histogram of the node strength distribution of the published graph under node-differential privacy. First, we propose a projection algorithm that preserves the information of the graph as much as possible by limiting the weight and degree and restricts the sensitivity of the node intensity histogram to be published on the projection graph. Then, we propose two algorithms: ALT-histogram and DEN-histogram and prove that they satisfy differential privacy. The experimental results show that, compared with the existing node strength histogram publishing algorithms, the proposed algorithm has advantages in L1-error and KS-distance, which makes the noisy node strength distribution closer to the node strength distribution of the original graph. In the future, we plan to extend the algorithm to triangle coefficients and local clustering coefficients.

REFERENCES

[1] L. Backstrom, C. Dwork, and J. Kleinberg, “Wherefore art thou R3579X? Anonymized social networks, hidden patterns, and structural steganography,” in Proc. 16th Int. Conf. World Wide Web, vol. 2007, pp. 181–190.

[2] A. Narayanan and V. Shmatikov, “De-anonymizing social networks,” in Proc. 50th IEEE Symp. Secur. Privacy, May 2009, pp. 173–187.

[3] M. Hay, C. Li, G. Miklau, and D. Jensen, “Accurate estimation of the degree distribution of private networks,” in Proc. 9th IEEE Int. Conf. Data Mining, Dec. 2009, pp. 169–178.

[4] C. Dwork, “Differential privacy: A survey of results,” in Proc. Int. Conf. Theory Appl. Models Comput., 2008, pp. 1–19.

[5] C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating noise to sensitivity in private data analysis,” in Proc. Theory Cryptogr. Conf., 2006, pp. 265–284.

[6] V. K. S. Raskhodnikova, A. Smith, and G. Yaroslavtsev, “Privacy analysis of graph structure,” Proc. VLDB Endowment, vol. 4, no. 11, pp. 1146–1157, Aug. 2011.

[7] V. K. S. Raskhodnikova and A. Smith, “Differentially private graphical degree sequences and synthetic graphs,” in Proc. Int. Conf. Privacy Stat. Databases, 2012, pp. 273–285.

[8] 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.

[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] S. P. Kasiviswanathan, K. Nissim, S. Raskhodnikova, and A. Smith, “Analyzing graphs with node differential privacy,” in Proc. Theory Cryptogr. Conf., Mar. 2013, pp. 457–476.

[11] S. Raskhodnikova and A. Smith, “Efficient lipschitz extensions for high-dimensional graph statistics and node private degree distributions,” 2015, arXiv:1504.07912. [Online]. Available: https://arxiv.org/abs/1504.07912

[12] J. Blocki, A. Blum, A. Datta, and O. Sheffet, “Differentially private data analysis of social networks via restricted sensitivity,” in Proc. 4th Conf. Innov. Theor. Comput. Sci. - ITCS, 2013, pp. 87–96.

[13] S. Chen and S. Zhou, “Recursive mechanism: Towards node differential privacy and unrestricted joins,” in Proc. Int. Conf. Manage. Data - SIGMOD, 2013, pp. 653–664.

[14] W.-Y. Day, N. Li, and M. Lyu, “Publishing graph degree distribution with node differential privacy,” in Proc. Int. Conf. Manage. Data - SIGMOD, 2016, pp. 123–138.

[15] Q. Qian, Z. Li, P. Zhao, W. Chen, H. Yin, and L. Zhao, “Publishing graph node strength histogram with edge differential privacy,” in Proc. Int. Conf. Database Syst. Adv. Appl., 2018, pp. 75–91.

[16] 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.

[17] F. McSherry and K. Talwar, “Mechanism design via differential privacy,” in Proc. 48th Annu. IEEE Symp. Found. Comput. Sci. (FOCS), Oct. 2007, pp. 94–103.

[18] F. D. McSherry, “Privacy integrated queries: An extensible platform for privacy-preserving data analysis,” in Proc. 35th SIGMOD Int. Conf. Manage. Data - SIGMOD, 2009, pp. 19–30.

[19] D. Kifer and B.-R. Lin, “Towards an axiomatization of statistical privacy and utility,” in Proc. 29th ACM SIGMOD-SIGACT-SIGART Princ. Database Syst. Data - PODS, 2010, pp. 147–158.

[20] Stanford Large Network Dataset Collection. [Online]. Available: http://snap.stanford.edu/data/