applied sciences

MDPI

Article

Graph Convolutional Networks for Privacy Metrics in Online Social Networks

Xuefeng Li [1,2], Yang Xin [1,2,*], Chensu Zhao [1,2,3], Yixian Yang [1,2] and Yuling Chen [2]

[1] National Engineering Laboratory for Disaster Backup and Recovery, Information Security Center, School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China; lxf3710@bupt.edu.cn (X.L.); zhao-cs@bupt.edu.cn (C.Z.); yxyang@bupt.edu.cn (Y.Y.)

[2] Guizhou Provincial Key Laboratory of Public Big Data, Guizhou University, Guizhou 550025, China; ylchen3@gzu.edu.cn

[3] School of Information and Engineering, Shandong Yingcai University, Jinan 250104, China

* Correspondence: yangxin@bupt.edu.cn

Received: 24 December 2019; Accepted: 13 February 2020; Published: 15 February 2020

Abstract: In recent years, privacy leakage events in large-scale social networks have become increasingly frequent. Traditional methods relying on operators have been unable to effectively curb this problem. Researchers must turn their attention to the privacy protection of users themselves. Privacy metrics are undoubtedly the most effective method. However, social networks have a substantial number of users and a complex network structure and feature set. Previous studies either considered a single aspect or measured multiple aspects separately and then artificially integrated them. The measurement procedures are complex and cannot effectively be integrated. To solve the above problems, we first propose using a deep neural network to measure the privacy status of social network users. Through a graph convolution network, we can easily and efficiently combine the user features and graph structure, determine the hidden relationships between these features, and obtain more accurate privacy scores. Given the restriction of the deep learning framework, which requires a large number of labelled samples, we incorporate a few-shot learning method, which greatly reduces the dependence on labelled data and human intervention. Our method is applicable to online social networks, such as Sina Weibo, Twitter, and Facebook, that can extract profile information, graph structure information of users' friends, and behavioural characteristics. The experiments show that our model can quickly and accurately obtain privacy scores in a whole network and eliminate traditional tedious numerical calculations and human intervention.

Keywords: Online Social Networks; privacy; graph convolutional networks; metrics; few-shot learning

1. Introduction

With the rapid development of the Internet, the cost for ordinary users to use the network is decreasing. The Internet has entered an era in which the whole populace are netizens. The Internet has become the most important way for people to obtain and release information, for which purpose, social networks are the most popular websites. However, in recent years, the privacy literacy of users has been unable to support their privacy protection behaviour [1], and an increasing number of privacy disclosure events have occurred; Facebook provided 50 million pieces of user data to the Cambridge company, which affected the American election in March 2018, and in November, due to the opening of the API (application programming interface) to third-party applications, 68 million users' mobile phone photo information was leaked. In 2017, Twitter announced that it was giving up its DNT (do not track) privacy protection policy. Tumblr disclosed 65 million users' email accounts and

passwords in May 2016. In addition, a hacker organisation called "peace" sold 167 million LinkedIn users' login information on the black market at the price of five bitcoins (approximately $2200). On 19 October 2016, the NetEase user database was leaked, and more than 100 million users' 163 and 126 mailbox information was leaked, including usernames, passwords, password security information, login IPs, and user birthdays. These privacy leakages may lead to a series of malicious acts [2–5], including but not limited to tracking, defamation, spam, phishing, identity theft, personal data cloning, Sybil attack, etc. However, academia has always focused on large-scale industrial networks, such as the Internet of Things, smart grids, and cloud storage networks, ignoring the privacy protection of individual users because the privacy leakage of large-scale networks causes substantial visible economic losses, but the impact of personal information leakage is scattered and small. Moreover, users' use of social networks is a process of sharing public information. Applying privacy protection methods to traditional industrial networks will damage user experience.

The existing privacy protection methods in social networks include anonymity, decentralisation, encryption, information security regulations, fine-grained privacy settings and access control, and improving user privacy awareness and privacy behaviour [6]. The first four methods need to rely on operators, which has proven to be unreliable. With the rapid development of attribute inference, link inference, personal identification, and community discovery, the role of anonymisation is becoming weaker and weaker. As long as users use social networks, they are disclosing information, so the adaptability of encryption is poor. Fine-grained privacy settings and access control rely on the user's own privacy literacy, but according to the research, less than one-third of users have changed these settings, and most of the settings are not reasonable. Therefore, the most fundamental solution is to help users cultivate their privacy awareness and enhance their privacy protection behaviour. Privacy metrics are the best way to achieve this goal. These metrics quantify all aspects of the ways that users may disclose private information on social networks and transform the virtual concept of privacy into specific values in the physical space so that users can intuitively understand their privacy status. If they are not satisfied with their current privacy status, they can continuously adjust it according to the privacy score, enhance their privacy awareness in the process of adjustment, and cultivate behavioural habits of privacy protection.

The method of traditional privacy metrics basically uses mathematical calculations to obtain quantitative statistics on all the aspects that affect users' privacy disclosure, including but not limited to attribute information, network environment information, trust between users, and publishing information content. However, these approaches face two problems. First, these approaches are inefficient. Most of these approaches first extract features, then measure them separately, and finally integrate them into a numerical value. In addition, the calculation method also faces various doubts because privacy is a virtual concept with no unifying principle, and any calculation is considered to be subjective and unconvincing. Second, this method relies too strongly on artificial feature extraction. In previous research on privacy metrics, feature extraction is a difficulty. Which features can be used for privacy measurement? Which features are more important? What associations exist between these features? These problems urgently need to be solved but have not been solved. Meanwhile, when considering the network environment of users [7], there may be tens of millions of links around a user. Previous research methods only obtained one user's privacy score after analysing the whole network environment, which is undoubtedly inefficient and a waste of resources.

For the above problems, deep neural networks, which are currently a popular research topic, have inspired us. A deep neural network can effectively extract the hidden relationship between features without human intervention. A large number of studies show that the hidden relationship between these features has a very large effect on the final results. Among the many neural network models, graph convolution networks (GCNs) [8] can extract features from graph structures and combine the features of nodes, which is suitable for scenarios in social networks. However, deep learning frameworks require a large amount of labelled data for training, and the vast majority of people cannot

accurately quantify privacy as a virtual concept, so data annotation is very difficult. Therefore, we label a small number of samples with the help of experts and use few-shot learning to solve this problem.

The goal of this paper is to use GCNs to build a framework to measure the privacy of online social network users. Compared with the traditional method, this framework has the following advantages. First, this framework can eliminate the interference of human subjective consciousness to the greatest extent, it only needs to select features, and it does not participate in the calculation and measurement process. Second, due to the difficulty of sample labelling in privacy measurement, our method only needs a small number of labelled samples. Finally, our method can obtain the privacy measurement scores of all the users in the whole network at the same time.

2. Related Work

For a long time, risk assessment in industrial networks, which is similar to personal privacy metrics, has been developing rapidly, but privacy metrics have been developing slowly. With the increase in the personal privacy disclosure events in recent years, some researchers have begun to focus on personal privacy.

In the early stages of privacy metrics, researchers measured a single aspect of privacy leakage. Dee et al. studied the harm caused by a large number of users' age information being leaked on social networks [9]. Srivastava et al. conducted an in-depth study on privacy leakage caused by character information being published by users [10]. Liang et al. found that after deleting published pictures on various social networks, they could still be accessed for a certain period of time, and such deletion delays could cause unnecessary privacy disclosure [11].

With the development of research, an increasing number of factors are being considered in privacy metrics. Maximilien et al. measured the sensitivity and visibility of multiple attribute information in user profiles and then calculated a unified score through Bayesian statistics [12]. Liu et al. proposed an IRT model to integrate the sensitivity and visibility of attributes, which more reasonably calculates the privacy score [13]. Aghasian et al. combined attribute sensitivity and visibility and used a statistical fuzzy system to solve the problem of calculating a privacy score on multiple social network platforms [14]. Fang et al. used the above methods to provide users with a guiding template for privacy settings on their profiles [1].

With the continuous in-depth study of social networks, the strong interactivity in social networks has attracted much attention. In addition, attribute inference, link inference, identity links, and other research topics have been greatly developed, which means researchers must extend the privacy problem to the whole network environment. The more friends around a user who pay attention to his life, the greater the risk of privacy leakage he faces [15]. Pensa et al. use the community discovery method to group users, measure the privacy in each group, and help users make reasonable changes to privacy settings through online learning. Zeng et al. believe that a single user's privacy disclosure depends on his trust in the friends around him and proposes a framework based on trust awareness to evaluate a user's privacy leakage [16]. Alsakal et al. [17] introduced information metrics for users in a whole social network by using information entropy theory and discussed the impact of individual identifying information and combinations of different pieces of information on user information disclosure.

With the development and influence of interdisciplinary subjects, such as social science, in recent years, researchers have been committed to analysing the graph structure in social networks and found that it is highly consistent with the small-world and homogeneity principles, thus triggering a series of studies. Serfontein et al. used the method of self-organising networks to study security threats to users in a network graph [18]. Djoudi et al. proposed an infectious communication framework that uses graph structure analysis, discussed the impact of high-trust and low-trust users in social networks, and proposed a calculation method for the trust index [19]. Due to a large number of users and low computational efficiency in the social network graph structure, some researchers have shifted their research objectives from users to social networks and performed privacy metrics on an entire social networking site, but this measurement has little significance for users. Oukemeni et al. conducted

detailed research on access control of social networking sites and various settings of users and then measured the privacy leakage of these social networking sites in different situations [6]. Yu et al. analysed possible aspects of privacy leakage in social networking sites and constructed a framework based on user groups, information control and posts, information deletion and reply, and other aspects to measure the privacy scores of multiple social networking sites [20]. De Salve et al. propose a Privacy Policies Recommended System (PPRS) that assists the users in defining their own privacy policies. The proposed system is able to exploits a certain set of properties (or attributes) of the users to define permissions on the shared contents [21].

However, in recent years, the application of graph neural networks in other fields has been far ahead of the direction of privacy protection, which often causes privacy threats. In [22], the author applies GNNs to social recommendation, which can perform social recommendation in heterogeneous networks, but it can also infer some characteristics of the target users from the recommended users. [23] used GNNs to link identities of users in multiple social networks. Although GNNs can provide more services for users, it also increases the malicious behaviour of identity inference. Mei et al. [24] proposed a framework for using a convolutional neural network (CNN) to carry out inference attacks on image and attribute information. Experiments showed that it can effectively and accurately infer attribute information that users do not disclose. At the same time, Li et al. [25] proposed a method of using a CNN to infer missing information in social network personal data; the effect is very excellent, but the missing information may be deliberately hidden by users. We hope that we can also use neural networks to help users improve their privacy awareness, cultivate their privacy behaviour, and have the ability to fight against these methods that may cause privacy disclosure.

To date, the privacy metrics of social networks have remained at the stage of using basic mathematical calculations to quantify multiple aspects of privacy, but the hidden relationship between these aspects is ignored. Meanwhile, these methods are inefficient, which means that in most studies, the whole network where users are located is analysed and quantified, but only the privacy score of the target user is obtained, which is undoubtedly a waste of resources. Related fields have entered the area of artificial intelligence, but the development of artificial intelligence in privacy measurement is slow.

To address the above problems, we innovatively introduce deep learning model GCNs to obtain users' privacy metrics on social networks. Graph convolution networks (GCNs) [8] are an efficient variant of convolutional neural networks (CNNs) on graphs. A GCN stacks layers of learned first-order spectral filters followed by a nonlinear activation function to learn graph representations. Recently, GCNs and their subsequent variants have made state-of-the-art achievements in various application fields, including but not limited to citation networks [8], social networks [26], applied chemistry [27], natural language processing [28–30], and computer vision [31,32].

Few-shot learning is a method to learn a unique set of new classes by using a few labelled data in a set of base classes without overfitting. A group of mature applications have emerged: Liu et al. [33] follow a semi-supervised learning setup but use graph-based label propagation (LP) [34] for classification and jointly use all test images. Douze et al. [35] extended this method to a wider range by using a large number of unlabelled images in the set of images for learning without using additional text information. In the research of Iscen et al. [36], a method of using GCNs and few-shot learning to identify a new class of image is proposed, which inspires us. In the field of privacy security, labelled data are scarce, and experts with relevant specialties are required to finish this work, which greatly limits research in deep learning. Therefore, we introduce few-shot learning to solve this problem.

In this paper, our contributions are as follows:

(1) We combine a user's attribute information, behaviour characteristics, friend relationships, and graph structure information to obtain the user's comprehensive privacy scores.
(2) We innovatively introduce the deep learning framework into the field of privacy measurement, which addresses the shortcomings of previous studies that can only calculate privacy metrics for a single user each time, extracts the hidden relationship between different features, and accurately and efficiently measures the privacy of users in the whole network.

(3) We introduce the few-shot learning method and SGCs (simplifying graph revolutionary networks), which can alleviate the difficulties of labelled data in the security field and long, time-consuming training in deep learning.
(4) We crawl real datasets on social networks, perform statistical analysis, extract features, and conduct an experimental demonstration of our model.

3. Datasets and Notation

3.1. Datasets

To use a GCN, we need to collect users' relationship graphs and features to build sample datasets. In addition, with the introduction of few-shot learning, we score a small number of samples in each category under the guidance of private security experts; these scores measure and grade the privacy status of users based on their characteristics. Then, we label the samples with the privacy scores. In this section, we will give a detailed description of the datasets and the extracted features we collect, which will provide the basis for subsequent method design and validation.

Due to the lag in the research of privacy measurement in social networks and datasets involving user privacy, there are few public datasets. The existing datasets in privacy metrics research are mainly referenced from similar fields, such as data mining, sentiment analysis, recommendation systems, link inference, community discovery, and attribute inference. However, these datasets are collected for their specific areas and do not contain all the content we need. Even for social network datasets, their content is not comprehensive, either containing only user information and graph information or containing published text information. For example, Blogcatalog is a dataset of social networks whose graph is composed of bloggers and their social relations (such as friends); labels indicate the interests of the blogger. Reddit is composed of posts from the Reddit forum. Epicions is a multigraph dataset collected from an online product review website that contains the attitudes (trust/distrust) of reviewers towards other reviewers and the rating of products by reviewers. In our study, we combine a user's profile information, the relationship information in the user's network environment, the user's behaviour characteristics, and other extracted features. No public datasets contain all of this information. Therefore, we build our dataset according to our needs.

Our data were collected on Sina Weibo, which is China's largest online social network, with 486 million monthly active users and 211 million daily active users. A user's friend relationships, profile, and publishing content are all public. We investigated nearly 500 students, teachers, and employees at our school. Approximately one-third of these individuals are users of Sina Weibo. In a further follow-up survey, we selected 16 users with certain differences as the seed of the crawling dataset according to the following criteria: (1) These users are active users of Sina Weibo. (2) These users have as many friend relationships as possible with other seed users. The reason for our requirements is to ensure that the final dataset conforms to the small-world and homogeneity principles and enables us to extract enough behaviour characteristics that relate users. If users are randomly selected from social networks, the resulting structure graphs may be disjoint.

We use the crawled Sina Weibo user data to build two datasets and extract the features we need. The first dataset contains these 16 users and all their friends, including Follows and Followers. The second dataset contains the friends of these 16 users and the friends of their friends. Specific information and selected feature information are shown in Tables 1 and 2. The node structure of Dataset 1 is shown in Figure 1.

Table 1. Datasets.

Dataset	Nodes	Edges
Dataset 1	3244	6452
Dataset 2	704,903	1,527,106

Table 2. Features.

Features		
Attribute extraction difficulty	Attribute sensitivity	The number of likes
The number of comments	The number of @	The number of reposts
The number of topics involved	The number of follower	The number of follow
The number of published Weibo	The number of published pictures	The number of published videos
Account usage time	Authenticated user or not	The time of last Weibo

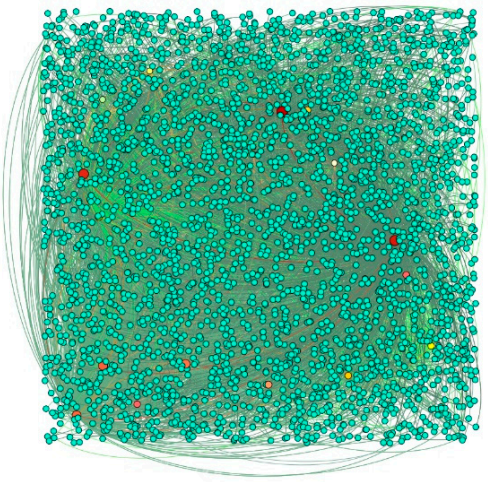

Figure 1. Dataset 1.

It should be noted that when we crawled the dataset, we processed users with large numbers of followers differently because most of these users are public figures and official institutions, and studying them would have little relevance to the privacy disclosure of ordinary users. Without processing, the number of followers in only one public figure account will reach hundreds of millions. Therefore, we only crawl some of the followers of these users, but the feature content is still calculated according to the actual number of followers.

The difficulty of extracting the features in Table 2 is the average extraction difficulty of the nine attributes in Table 3. The extraction difficulty of an attribute is set to 0 or 1. An attribute that can be directly obtained from the profile has a difficulty of 1; otherwise, it is 0. Sensitivity is the sum of the sensitivity of the exposed attributes of the user, and the sensitivity of each attribute is shown in Table 3. Account usage time is the number of months from the registration time to the time when we crawled the data. The registration time can be directly obtained from the profile. The time of last Weibo is the number of months from the last published Weibo to the time when we crawled the data. The number of likes, comments, reposts, @-mentions, topics involved, and Weibo are obtained from the Weibo content published by users. The number of pictures and videos sent can be obtained from social networking sites.

Table 3. Attribute sensitivity.

Attribute	Sensitivity
Phone Number	0.5669
Email	0.3260
Hometown	0.2253
Birthdate	0.2748
Address	0.4212
Job Details	0.2024
Relationship Status	0.1731
Interests	0.1255
Education	0.1575

The sensitivity of attributes is obtained through statistical analysis of an online questionnaire (https://www.wjx.cn/report/1647730.aspx). We have the degree of anxiety of users after they disclose these attributes as an option, let users choose it according to their own ideas, and then objectively and accurately count the privacy sensitivity of attributes. We designed five options: *L1*: very worried, *L2*: worried, *L3*: not clear, *L4*: not worried, and *L5*: not worried at all. The final result represented for each option is the percentage of the number of people who selected the option. We generate statistics on the data obtained from the questionnaire, take L3 as the benchmark, and use adjustment parameters and formulas to calculate the sensitivity of each attribute. The larger the value is, the more sensitive the attribute is, and the more worried the user is about the disclosure of this attribute content. The final results are shown in Table 3.

$$\theta = \frac{0.5 * L3 + L4 + 1.5 * L5}{1.5} \tag{1}$$

3.2. Problem Description and Notation

In social networks, privacy metrics evaluate various factors that may cause privacy disclosure and finally obtain a number that can map the user's privacy status, which represents the degree of the user's privacy disclosure. In this paper, we aim to integrate the attribute information, friend relationships, and behaviour characteristics of users on social networks through the introduction of GCNs to obtain the degree of privacy exposure and achieve the purpose of privacy metrics. In addition, our method can reduce subjective intervention and the difficulty of labelling samples.

We define the sample space as $X = \{x_1, x_2, x_3 \cdots\cdots x_N\}^T$. The privacy scores are defined as numbers from 1 to 7, that is, the ultimate goal is to classify unlabelled samples into $K = \{k_1, k_2, k_3, k_4, k_5, k_6, k_7\}$. The privacy score of a sample is the corresponding category number $i = |k_i|$ and is ultimately classified into a certain category. In each category k, there are few labelled samples X_k^l, $|X_k^l| = C \in \{1, 5, 10, 20, 50, 100\}$ and an unlimited number of unlabelled samples X^z. The feature set of the samples is $V_N = [v_1, v_2, v_3, \cdots\cdots, v_d]$, and finally, all the samples can be expressed as a matrix $V \in \mathbb{R}^{N \times d}$.

To combine the graph structure information, we establish the adjacency matrix $A \in \mathbb{R}^{N \times N}$ of nodes, where a_{ij} indicates that the link of node i points to node j, and in a social network, it indicates that user i follows user j. It is worth noting that the adjacency matrix we establish is asymmetric—a_{ij} is not necessarily equal to a_{ji}—which means that the relationship between nodes is bilateral. Since the elements on the diagonal of the matrix A are all 0, to be able to back-propagate in the training of the deep neural network, we normalise the adjacency matrix A to obtain $\widetilde{A} = A + I$, where $D = diag((A + I)1)$ is the degree matrix of $\widetilde{A}$ and **1** is a vector for which the elements are all 1.

4. Framework Design

4.1. GCNs

Our framework is based on GCNs, so we first give a brief overview of GCNs. Similar to deep neural networks, such as CNNs (convolutional neural networks) and RNNs (recurrent neural networks), GCNs are also multilayer stack structures. Each hidden layer consists of three steps: feature propagation, linear transformation, and nonlinear transformation. For ease of description, we express the input of the nth layer as $H^{(n-1)}$. The input of the first layer is the initial feature matrix:

$$H^{(0)} = V \tag{2}$$

Feature Propagation In brief, the purpose of feature propagation is to aggregate the information of other nodes within n hops of each node into the node itself so that each node can learn the information of the surrounding nodes in the network structure. The specific method is to use the adjacency matrix of nodes to construct a standardised correlation matrix P:

$$P = D^{-\frac{1}{2}}\widetilde{A}D^{-\frac{1}{2}} \tag{3}$$

Feature propagation can be expressed as:

$$\overline{H}^{(n-1)} \leftarrow PH^{(n-1)} \tag{4}$$

By stacking an additional layer of feature propagation, a node can aggregate information from nodes that are one more hop away.

Linear Transformation To ensure that the model achieves the purpose of learning, there is a weight matrix in each layer of linear transformation. The weight matrix is accompanied by the learning of samples, and the value is adjusted continuously by reducing the loss function:

$$\overline{\overline{H}}^{(n-1)} \leftarrow \overline{H}^{(n-1)}\Theta^{(n-1)} \tag{5}$$

Nonlinear Transformation Nonlinear transformation uses nonlinear functions to make the model learn the nonlinear characteristics of samples. The most commonly used function in GCNs is the RELU function. The final output of the hidden layer is:

$$H^{(n)} \leftarrow \mathrm{ReLU}(\overline{\overline{H}}^{(n-1)}) \tag{6}$$

Finally, the output layer can choose different functions according to specific application scenarios, such as Sigmoid and SoftMax.

4.2. SGC

In the model training of GCNs, we find that the training process is slow and time-consuming when the number of nodes in the network structure graph is large. However, in social networks, the number of nodes within two hops may be on the order of millions. As the number of hops increases, the number of hidden layers should increase accordingly, which increases the difficulty of model training. Therefore, we refer to the improvements made by Wu et al. [37] on GCNs; they proposed an SGC (simplifying graph convolutional network) model that can greatly reduce training time. The experiments show that after removing the nonlinear transformation of the hidden layer, the continuous power multiplication of the adjacency matrix can be pre-calculated, thereby greatly reducing the

training time of the model, and the results are still significant. The improved overall model can be expressed as Formula (7):

$$H^{(n)} = P \cdot \cdots \cdot PV\Theta^{(1)} \cdots \Theta^{(n-1)} = P^n V\Theta \tag{7}$$

In $P^n V$, there is no weight matrix involved in the calculation, which can be input into the network as a whole matrix after pre-calculation, effectively reducing the training time.

4.3. Framework Structure

To solve the problem that we put forward above, we use the SGC model to build a two hop SGC neural network. The network model can be represented by F_Θ:

$$F_\Theta = Sigmoid(PPV\Theta_1\Theta_2) \tag{8}$$

where $P \in \mathbb{R}^{N\times N}$ is the adjacency matrix, $V \in \mathbb{R}^{N\times d}$ is the feature matrix, and $\Theta_1 \in \mathbb{R}^{d\times l}$ and $\Theta_2 \in \mathbb{R}^{l\times 1}$ are the weight matrixes. Finally, the model outputs a matrix of $\mathbb{R}^{N\times 1}$ through the sigmoid function, which represents the belonging degree $r^k \in [0, 1]$ of each sample to the specified class k. The closer the belonging degree is to 1, the more similar the sample is to this class, and we assume that the user's privacy score is the value represented by this class. When the model is trained, we set the label of all the labelled samples as 1 for input and the other unlabelled samples as 0. The weight matrix Θ is optimised by minimising the loss function $L(V, P; \Theta)$, which is shown in Formula (9).

$$L(V, P; \Theta) = -\frac{1}{C}\sum_{i=1}^{|X^l|} \log(F^i_\Theta) - \frac{\lambda}{(N-C)}\sum_{j=1}^{|X^z|} \log(1 - F^j_\Theta) \tag{9}$$

λ represents the weight of unlabelled samples in the loss function, which can make unlabelled samples with high similarity obtain a higher belonging degree, while those with low similarity tend to 0. The value of λ is discussed in the section on the experiment.

Based on the above model, we design a framework to measure the privacy score of users in the whole network. For example, we put a sample in which the privacy score is labelled as 1 and other unlabelled samples into the model for training. The belonging degree r^1 of all the unlabelled samples is output. With the same step, we can obtain the belonging degree of each sample with other classes r^2, r^3, r^4, r^5, r^6, and r^7. Finally, the privacy scores of all the unlabelled samples are calculated as in Formula (10); the architecture of the whole framework is shown in Figure 2.

$$S_N = i \leftarrow \max(r^i_N) \tag{10}$$

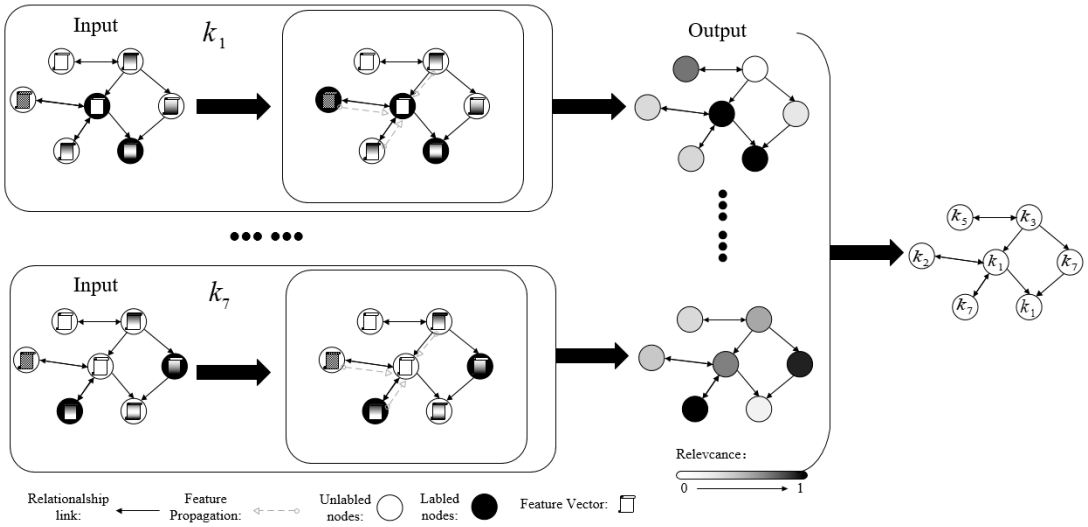

Figure 2. Framework architecture.

Through our method, we can also obtain the belonging degree of the labelled samples. In contrast to the unlabelled samples, the belonging degree of the corresponding class of the labelled samples is 1, but for incorrect categories, a sample will have a new belonging degree. For example, the final belonging degree of a labelled sample with a privacy score of 1 may be [1,0.13,0.1,0.07,0.11,0.08,0.05]. In the selection of λ, we will combine different C values to calculate the belonging degree deviation E of all the labelled samples in incorrect categories. The smaller the deviation is, the more accurate our model will be. The calculation method is as follows:

$$E = \frac{(\sum_{i=1}^{|X^l|} \sum_{j=1}^{|K|} r_{\mathrm{i}}^{j}) - |X^l|}{|X^l|} \tag{11}$$

E represents the mean value of the sum of belonging degrees of all labelled samples in incorrect classes.

5. Experimental Evaluation

5.1. Parameter Selection

We use Formula (11) and multiple experiments to verify the combination of parameters λ and C of Formula (9) in Dataset 1 and Dataset 2, respectively. The specific results are shown in Figures 3 and 4:

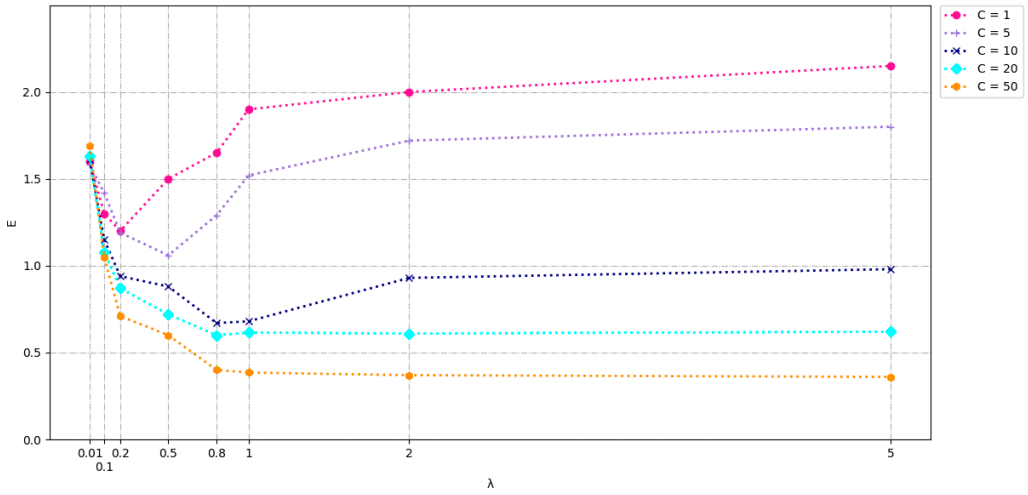

Figure 3. λ and C in Dataset 1.

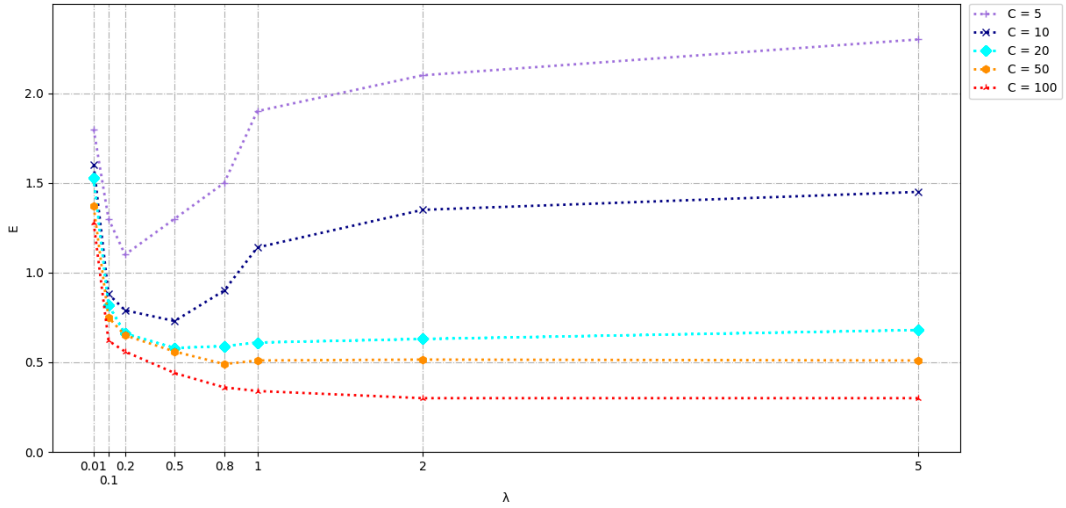

Figure 4. λ and C in Dataset 2.

It can be seen from Figures 3 and 4 that with the increase in C, the larger λ is, and the smaller the deviation E is. However, due to the limitation of the value of C, there is a certain limit to the reduction of deviation. This result shows that the model still needs a certain number of labelled samples, from which it can obtain more features for learning. However, it will cost more effort to label samples. The specific choice can be made according to the actual scenario.

5.2. Experiment 1

Currently, in the study of privacy metrics on social networks, researchers are committed to numerically quantifying multiple relevant factors of privacy leakage and ultimately combining these quantitative values to comprehensively calculate the privacy score. However, the procedures are cumbersome, and the selection of factors is subjective and limited. Meanwhile, the deep connections between these factors and features cannot be used.

We are the first to propose a privacy metric method based on a deep neural network. To verify its accuracy, we select 50 unlabelled samples in each category of model output results for manual labelling and compare them with the model output results to calculate accuracy. It is worth noting that there is considerable subjectivity in manual labelling. It is unfair to use our label to verify other researchers' methods, so we only verify the accuracy of models under different parameters in our methods. Due to space limitations, we choose the combination of parameters in Figures 3 and 4 for verification.

From Table 4, it can be seen that the performance of the model can be greatly improved with the optimisation of parameters, whether in a small dataset or a large dataset. In addition, we can see that the model has the best measurement effect when the privacy score has the highest or lowest value. Since these two types of users are mostly public figure accounts and abandoned accounts, they are very different from normal users.

Table 4. Accuracy in different class.

	k_1	k_2	k_3	k_4	k_5	k_6	k_7
Dataset 1							
$C=1$, $\lambda=0.2$	52%	40%	42%	52%	36%	38%	54%
$C=5$, $\lambda=0.5$	70%	52%	50%	62%	48%	46%	74%
$C=10$, $\lambda=0.8$	80%	62%	58%	82%	58%	54%	86%
$C=20$, $\lambda=0.8$	84%	74%	70%	86%	68%	62%	92%
$C=50$, $\lambda=1$	92%	88%	82%	92%	84%	84%	98%
Dataset 2							
$C=5$, $\lambda=0.2$	64%	44%	54%	62%	50%	42%	70%
$C=10$, $\lambda=0.5$	76%	60%	64%	72%	58%	56%	80%
$C=20$, $\lambda=0.5$	80%	62%	70%	78%	58%	58%	86%
$C=50$, $\lambda=0.8$	84%	78%	82%	84%	80%	78%	96%
$C=100$, $\lambda=2$	96%	92%	86%	94%	88%	84%	100%

5.3. Experiment 2

To demonstrate the superiority of choosing SGCs over GCNs and to compare our model with others, we add the improved FastGCN model of Chen et al. [26] to compare the performance of three different models with the same datasets and parameters. We compare the training time and accuracy of the three models. The results are the average of ten runs, which are shown in Tables 5 and 6.

Table 5. Training time.

Models	Dataset 1 (Seconds)	Dataset 2 (Hours, Minutes)
GCNs	0.57 s	23 h 36 m
FastGCN	3.88 s	14 h 22 m
SGC	0.14 s	9 h 17 m

Table 6. Test accuracy.

	GCNs	FastGCN	SGC
Dataset 1			
$C=1$, $\lambda=0.2$	47.14%	46%	46.57%
$C=5$, $\lambda=0.5$	55.71%	54.86%	56%
$C=10$, $\lambda=0.8$	69.71%	67.71%	68.57%
$C=20$, $\lambda=0.8$	76.57%	75.14%	76.57%
$C=50$, $\lambda=1$	83.86%	84.86%	85.57%
Dataset 2			
$C=5$, $\lambda=0.2$	54.57%	51.71%	54.86%
$C=10$, $\lambda=0.5$	64.86%	60%	66%
$C=20$, $\lambda=0.5$	72.57%	69.57%	72.29%
$C=50$, $\lambda=0.8$	81.43%	76.86%	82.57%
$C=100$, $\lambda=2$	91.86%	88.57%	91.43%

The training time of GCNs and the improved model (FastGCN and SGC) on Dataset 1 and Dataset 2, respectively, are listed in Table 5. SGC greatly reduces the time consumption of training, which is due to the pre-calculation of the continuous power multiplication of the adjacency matrix and the feature matrix. The effect is very significant regardless of whether the dataset is small or large. FastGCN is less effective in small datasets because it improves computational complexity by sampling to reduce neighbourhood size, and it works better for large datasets. In the comparison of accuracy, it can be seen from Table 6 that the differences among the three models are very small because SGC and FastGCN are improvements on GCNs for model simplification and time efficiency. In actual social network sites, the relationship between user nodes is complex, and there are a large number of surrounding nodes. SGC is undoubtedly a better choice.

6. Discussions and Conclusions

In the field of social network privacy metrics, the traditional method of calculating privacy metrics is to measure individual users' privacy disclosure factors separately and then integrate them to obtain an overall privacy score. The present study was designed to innovatively introduce a deep learning model that integrates multiple factors, such as graph structure, attribute information, and behaviour characteristics but avoids tedious calculation procedures. The superior experimental results show that it is necessary to consider the deep relationship between these different factors to obtain more accurate results. In addition, our model can measure the privacy of all users in a whole network at one time, which greatly improves efficiency. Through the introduction of few-shot learning, the problem of an insufficient number of labelled samples is also alleviated.

The present study was designed to provide a new direction for future research, but there are still some problems. Our model needs to be trained many times for different categories. Although the use of SGC has greatly reduced the training time, it still requires substantial time for large datasets. In future research, we will work on improving the model so that it can obtain a user's privacy score through a one-time training model. In addition, we hope that more researchers will be able to apply more deep learning methods so that we can more comprehensively compare the advantages and disadvantages of these methods.

Author Contributions: Funding acquisition, Y.Y.; methodology, X.L.; software, C.Z.; visualisation, X.L. and Y.C.; writing—original draft, X.L.; writing—review and editing, Y.X. and Y.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by BUPT Excellent Ph.D. Students Foundation CX2019319, Foundation of the National Key R&D Program of China under Grant 2017YFB0802300, Foundation of Major Scientific and Technological Special Project of Guizhou Province under Grant 20183001, and in part by Foundation of Guizhou Provincial Key Laboratory of Public Big Data under Grant 2018BDKFJJ008, 2018BDKFJJ020, 2018BDKFJJ021.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Fang, L.; LeFevre, K. Privacy wizards for social networking sites. In Proceedings of the 19th International Conference on World Wide Web, Raleigh, NC, USA, 26–30 April 2010; ACM: New York, NY, USA, 2010; pp. 351–360.
2. Qiu, M.; Gai, K.; Xiong, Z. Privacy-preserving wireless communications using bipartite matching in social big data. *Futur. Gener. Comput. Syst.* **2018**, *87*, 772–781. [CrossRef]
3. Xu, L.; Jiang, C.; Chen, Y.; Wang, J.; Ren, Y. A framework for categorizing and applying privacy-preservation techniques in big data mining. *Computer* **2016**, *49*, 54–62. [CrossRef]
4. Lam, I.F.; Chen, K.T.; Chen, L.J. Involuntary information leakage in social network services. In *International Workshop on Security*; Springer: Berlin, Germany, 2008; pp. 167–183.
5. Patsakis, C.; Zigomitros, A.; Papageorgiou, A.; Galván-López, E. Distributing privacy policies over multimedia content across multiple online social networks. *Comput. Netw.* **2014**, *75*, 531–543. [CrossRef]

6. Oukemeni, S.; Rifà-Pous, H.; Puig, J.M.M. Privacy Analysis on Microblogging Online Social Networks: A Survey. *ACM Comput. Surv. CSUR* **2019**, *52*, 60. [CrossRef]
7. Altenburger, K.M.; Ugander, J. Monophily in social networks introduces similarity among friends-of-friends. *Nat. Hum. Behav.* **2018**, *2*, 284. [CrossRef] [PubMed]
8. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. In Proceedings of the International Conference on Learning Representations (ICLR'2017), Toulon, France, 24–26 April 2017.
9. Dey, R.; Tang, C.; Ross, K.; Saxena, N. Estimating age privacy leakage in online social networks. In Proceedings of the INFOCOM, 2012 Proceedings IEEE, Orlando, FL, USA, 25–30 March 2012; IEEE: New York, NY, USA, 2012; pp. 2836–2840.
10. Srivastava, A.; Geethakumari, G. Measuring privacy leaks in online social networks. In Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Mysore, India, 22–25 August 2013; IEEE: New York, NY, USA, 2013; pp. 2095–2100.
11. Liang, K.; Liu, J.K.; Lu, R.; Wong, D.S. Privacy concerns for photo sharing in online social networks. *IEEE Internet Comput.* **2015**, *19*, 58–63. [CrossRef]
12. Maximilien, E.M.; Grandison, T.; Sun, T.; Richardson, D.; Guo, S.; Liu, K. Privacy-as-a-service: Models, algorithms, and results on the Facebook platform. In *Web 2.0 Security and Privacy Workshop*; IEEE: New York, NY, USA, 2009; Volume 2.
13. Liu, K.; Terzi, E. A framework for computing the privacy scores of users in online social networks. *ACM Trans. Knowl. Discov. Data* **2010**, *5*, 6. [CrossRef]
14. Aghasian, E.; Garg, S.; Gao, L.; Yu, S.; Montgomery, J. Scoring Users' Privacy Disclosure Across Multiple Online Social Networks. *IEEE Access* **2017**, *5*, 13118–13130. [CrossRef]
15. Pensa, R.G.; Di Blasi, G.; Bioglio, L. Network-aware privacy risk estimation in online social networks. *Social Netw. Anal. Min.* **2019**, *9*, 15. [CrossRef]
16. Pensa, R.G.; Di Blasi, G. A privacy self-assessment framework for online social networks. *Expert Syst. Appl.* **2017**, *86*, 18–31. [CrossRef]
17. Alsarkal, Y.; Zhang, N.; Xu, H. Your Privacy Is Your Friend's Privacy: Examining Interdependent Information Disclosure on Online Social Networks. In Proceedings of the 51st Hawaii International Conference on System Sciences, Waikoloa Village, HI, USA, 2–6 January 2018.
18. Serfontein, R.; Kruger, H.; Drevin, L. Identifying Information Security Risks in a Social Network Using Self-organising Maps. In Proceedings of the IFIP World Conference on Information Security Education, Lisbon, Portugal, 25–27 June 2019; Springer: Cham, Germany; pp. 114–126.
19. Djoudi, A.; Pujolle, G. Social Privacy Score through Vulnerability Contagion Process. In Proceedings of the 2019 Fifth Conference on Mobile and Secure Services (MobiSecServ), Miami Beach, FL, USA, 2–3 March 2019; IEEE: New York, NY, USA, 2019; pp. 1–6.
20. Yu, L.; Motipalli, S.M.; Lee, D.; Liu, P.; Xu, H.; Liu, Q.; Tan, J.; Luo, B. My Friend Leaks My Privacy: Modeling and Analyzing Privacy in Social Networks. In Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies, Indianapolis, IN, USA, 13–15 June 2018; ACM: New York, NY, USA, 2018; pp. 93–104.
21. De Salve, A.; Guidi, B.; Michienzi, A. Exploiting Community Detection to Recommend Privacy Policies in Decentralized Online Social Networks. In Proceedings of the European Conference on Parallel Processing, Turin, Italy, 27–31 August 2018; Springer: Cham, Germany; pp. 573–584.
22. Fan, W.; Ma, Y.; Li, Q.; He, Y.; Zhao, E.; Tang, J.; Yin, D. Graph Neural Networks for Social Recommendation. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; ACM: New York, NY, USA, 2019; pp. 417–426.
23. Zhang, W.; Shu, K.; Liu, H.; Wang, Y. Graph Neural Networks for User Identity Linkage. *arXiv* **2019**, arXiv:1903.02174.
24. Mei, B.; Xiao, Y.; Li, R.; Li, H.; Cheng, X.; Sun, Y. Image and attribute based convolutional neural network inference attacks in social networks. *IEEE Trans. Netw. Sci. Eng.* **2018**, 1. [CrossRef]
25. Li, X.; Cao, Y.; Shang, Y.; Liu, Y.; Tan, J.; Guo, L. Inferring user profiles in online social networks based on convolutional neural network. In Proceedings of the International Conference on Knowledge Science, Engineering and Management, Melbourne, Victoria, Australia, 19–20 August 2017; Springer: Cham, Germany, 2017; pp. 274–286.

26. Chen, J.; Ma, T.; Xiao, C. FastGCN: Fast Learning with Graph Convolutional Networks via Importance Sampling. In Proceedings of the International Conference on Learning Representations (ICLR'2018), Vancouver, BC, Canada, 30 April–3 May 2018.
27. Liao, R.; Zhao, Z.; Urtasun, R.; Zemel, R. Lanczosnet: Multi-scale deep graph convolutional networks. *arXiv* **2019**, arXiv:1901.01484.
28. Yao, L.; Mao, C.; Luo, Y. Graph convolutional networks for text classification. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI'19), Hilton Hawaiian Village, Honolulu, HI, USA, January 27–February 1 2019.
29. Zhang, Y.; Qi, P.; Manning, C.D. Graph convolution over pruned dependency trees improves relation extraction. *arXiv* **2018**, arXiv:1809.10185.
30. Han, B.; Cook, P.; Baldwin, T. Geolocation prediction in social media data by finding location indicative words. In Proceedings of the 24th International Conference on Computational Linguistics, Mumbai, India, 8–15 December 2012; Dublin City University and Association for Computational Linguistics: Dublin, German, 2012; pp. 1045–1062.
31. Kampffmeyer, M.; Chen, Y.; Liang, X.; Wang, H.; Zhang, Y.; Xing, E.P. Rethinking knowledge graph propagation for zero-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 11487–11496.
32. Wang, X.; Ye, Y.; Gupta, A. Zero-shot recognition via semantic embeddings and knowledge graphs. In *Computer Vision and Pattern Recognition (CVPR)*; IEEE Computer Society: San Francisco, CA, USA, 2018; pp. 6857–6866.
33. Liu, Y.; Lee, J.; Park, M.; Kim, S.; Yang, E.; Hwang, S.J.; Yang, Y. Learning to propagate labels: Transductive propagation network for few-shot learning. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
34. Zhou, D.; Bousquet, O.; Lal, T.N.; Weston, J.; Schölkopf, B. Learning with local and global consistency. In Proceedings of the NeurIPS, Vancouver, BC, Canada, 8–13 December 2003.
35. Douze, M.; Szlam, A.; Hariharan, B.; Jégou, H. Low-shot learning with large-scale diffusion. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 19–24 June 2018.
36. Iscen, A.; Tolias, G.; Avrithis, Y.; Chum, O.; Schmid, C. Graph convolutional networks for learning with few clean and many noisy labels. *arXiv* **2019**, arXiv:1910.00324.
37. Wu, F.; Zhang, T.; Souza, A.H.D., Jr.; Fifty, C.; Yu, T.; Weinberger, K.Q. Simplifying graph convolutional networks. *arXiv* **2019**, arXiv:1902.07153.