Hindawi
Security and Communication Networks
Volume 2019, Article ID 3053418, 14 pages
https://doi.org/10.1155/2019/3053418

Research Article

Stopping the Cyberattack in the Early Stage: Assessing the Security Risks of Social Network Users

Bo Feng,[1] Qiang Li,[1,2] Yuede Ji,[3] Dong Guo,[1,2] and Xiangyu Meng[1,2]

[1]*College of Computer Science and Technology, Jilin University, Changchun 130012, China*
[2]*Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, China*
[3]*Department of Electrical and Computer Engineering, DC George Washington University, Washington, D.C., USA*

Correspondence should be addressed to Xiangyu Meng; xymeng512@jlu.edu.cn

Received 1 December 2018; Accepted 16 June 2019; Published 11 July 2019

Academic Editor: Bela Genge

Online social networks have become an essential part of our daily life. While we are enjoying the benefits from the social networks, we are inevitably exposed to the security threats, especially the serious Advanced Persistent Threat (APT) attack. The attackers can launch targeted cyberattacks on a user by analyzing its personal information and social behaviors. Due to the wide variety of social engineering techniques and undetectable zero-day exploits being used by attackers, the detection techniques of intrusion are increasingly difficult. Motivated by the fact that the attackers usually penetrate the social network to either propagate malwares or collect sensitive information, we propose a method to assess the security risk of the user being attacked so that we can take defensive measures such as security education, training, and awareness before users are attacked. In this paper, we propose a novel user analysis model to find potential victims by analyzing a large number of users' personal information and social behaviors in social networks. For each user, we extract three kinds of features, i.e., statistical features, social-graph features, and semantic features. These features will become the input of our user analysis model, and the security risk score will be calculated. The users with high security risk score will be alarmed so that the risk of being attacked can be reduced. We have implemented an effective user analysis model and evaluated it on a real-world dataset collected from a social network, namely, Sina Weibo (Weibo). The results show that our model can effectively assess the risk of users' activities in social networks with a high area under the ROC curve of 0.9607.

1. Introduction

With the development of smart terminals, social networks have become part of people's private and business communication. For example, the daily active user of Sina Weibo reached 165 million, an increase of 25% over the last year according to 2017 Weibo User Development Report [1]. While people enjoy the convenience brought by social networks, they also face security threats from social networks, such as phishing, drive-by download, and malicious code injection. These malicious behaviors threaten the users' security of information and property. Recently, South Korean media wrote about North Korean refugees and journalists being targeted by unknown actors using KakaoTalk (a popular chat app in South Korea) and other social network services (such as Facebook) to send links to install malware on victims' devices. This method shows that attackers are always looking for different ways to deliver malware [2].

Nowadays, more and more attackers launch cyberattack through social networks [3]. People's social networking behaviour, whether accidental or intentional, provides an opportunity for attackers to launch targeted attacks. There are two kinds of ways to exploit social networks to launch attacks: (1) The attacker uses the mutual trust relationship with the user to directly send the disguised malicious URLs, which could be hidden in the short links or pictures [4–7]. (2) The attacker launches spear-phishing attacks or water hole attacks against the user according to the user's preference [8, 9]. After gathering sufficient information of the members from a target organization, they can leverage the members to penetrate the target network.

In order to protect the users away from cyberattacks and provide users with a secure social network environment, many researchers make a great effort for it. Existing detection methods are mainly divided into two categories. The first category is the detection algorithms based on the relation

graph of social networks [10–12]. Many kinds of relations exist in social networks, so researchers use relations in social networks to build a social graph. Through analyzing the characteristics of the user's location in the graph, a detection algorithm can be designed based on the graph to identify suspicious messages or users. However, the accuracy of the detection algorithm based on graph is relatively low, and different social networks have their own graph structure. The second category is the detection methods based on machine learning algorithms [7, 13–15]. Researchers extract features from social network data such as users' personal information, social behavior, relationship with friends, and message content and then use machine learning algorithms to train classifiers to identify malicious messages or users. However, once the attacker modifies key features, this detection method can be obsolete easily.

In social networks, users usually trust their communication partners although they are only verified by an email address or a virtual profile [16]. According to the survey, more and more people believe that the use of social networks has increased the probability of launching a successful APT attack (95 percent in 2015, up from 92 percent in 2014) [17]. Attackers use a spear-phishing method that targeted key employees of victim organisations through social networks in order to conduct reconnaissance and theft of confidential proprietary information. To penetrate a company's highly protected network, the attackers usually start attack from the employees when they are out of the company. Therefore, the social network becomes a major penetration source for the attackers. According to the survey [18], 67% of companies report that they have not increased awareness training relative to cyberattacks. In order to stop the attack in the early stage, we try to identify the members who are most likely to be attacked by analyzing their social network activities; then we can take defensive measures such as security education, training, and awareness before users being attacked.

In this paper, we propose a novel user analysis model to assess the users' security risk. Firstly, we collect the social network information of 4,536 users in 134 days from Weibo, including their social relations, social behaviors, and microblogs. Among these users, we mark 107 normal users who had abnormal behaviors as positive samples and the rest are marked as negative samples. By analyzing the positive and negative data from the perspective of the attacker, we extract three kinds of features, i.e., statistical features, social-graph features, and semantic features. For some of the more complex features in these three kinds of features, we described in detail how to extract these features. For the feature complexity of social circle, we get it by establishing a user follow graph and using hierarchical clustering method and for the feature obviousness of preference, we build a microblog topic classification system based on Google's word2vec tool and TensorFlow framework. After classifying the topics of users' microblog, we utilize the properties of normal distribution curve to calculate the obviousness of preference of each user. Finally, we build a feature vector for each user as input to our user analysis model to get the corresponding security risk score. According to the confusion matrix and other evaluation parameters, we compare the performance of using different machine learning algorithms as the classifier and plot ROC curves for each classifier to prove the validity of them. The experimental results show that our user analysis model can effectively quantify the user's security level based on the user's social information.

In summary, this paper makes the following contributions:

(1) We propose a novel user analysis model to identify potential victims by analyzing the users' personal information and social behaviors. By giving each user a security score, we are able to alarm the users with high risks.

(2) We extract five features and classify these features into three kinds. To extract these features, we design an algorithm to obtain the complexity of users' social circle and build a microblog topic classification system to get the obviousness of users' preference. After all the features have been extracted, we set up a feature vector for each user.

(3) We use different evaluation parameters to demonstrate the validity of our user analysis model based on experiments results on the Weibo dataset. Using the GDBT classifier, the accuracy of our approach is about 89.78% and the value of AUC is about 96.07%.

The remainder of our paper is organized as follows. Related work is reviewed in Section 2. Overview is described in Section 3. Section 4 proposes two statistical features, Section 5 proposes two social-graph features, and Section 6 proposes a semantic feature. Section 7 introduces the implementation of user analysis model and the effectiveness of it is evaluated in Section 8. The paper concludes in Section 9.

2. Related Work

The detection methods for attackers are mainly divided into two categories, the algorithms based on the social graph and machine learning classifiers based on the social features. Cao et al. introduce SybilRank, through O(log n) power iteration, trust flows from known non-Sybil nodes spreads over the entire network, and then rank nodes based on their degree-normalized trust, and non-Sybil nodes are ranked higher than Sybil nodes [10]. Gong et al. introduce SybilBelief to detect Sybil nodes. SybilBelief first labels each user as non-Sybil or Sybil by a binary random variable. Then Markov Random Field is used to compute the probability of a user being benign [11]. Mulamba et al. introduce SybikRadar which improves the SybilRank and use Louvain Method to find the different communities of their social graph and compute the similarity between any pair of nodes. Then those similarity values are put as weights on the social graph to get actual attack edges. Finally, they use Supervised Random Walk to rank the nodes [12]. Lee et al. find that some malicious URLs in Twitter would jump to different pages if there are detection tools in the current environment. They analyze the URL redirection path and find the access point of condition redirection, then extract features from the jump path and tweet, and then train a classifier to detect malicious URLs [13]. Cao et al. propose a forwarding message tree and extract features from it to train a classifier to find hidden suspicious accounts which forward certain

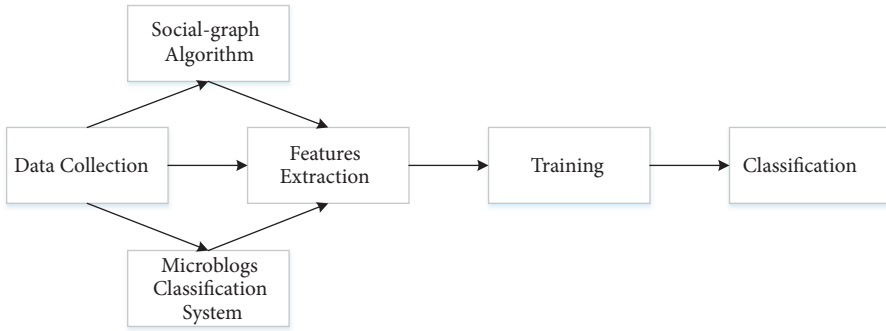


Figure 1: The structure of user analysis model.

suspicious messages together [14]. Fu et al. propose a dynamic model to measure the changes of users activities; they extract features from the temporal evolution patterns of users and then combine unsupervised clustering and supervised classification to detect the evolving spammers [15]. Cao et al., through analyzing the connection between forwarding behavior and the propagation of malicious URLs, propose three forwarding-based features. They combine these features with other social features to train a classifier to identify malicious URLs [7].

Human factors in information security management cannot be ignored, and improving employees' security awareness is very important for information security. Coronges et al. pointed out that attackers implemented more powerful phishing attacks by extracting information from users on different social networks. Experimental results show that this phishing attack has a higher success rate than traditional phishing attacks. In addition, the attacker may establish a mutual trust relationship with the victim on the social network and then penetrate the target network through the social network [19]. Egele et al. proposed a novel method to detect hijacking accounts in social networks by identifying sudden changes in user behavior. The attacker uses the mutual trust relationship on the social network to send messages to the victim in the form of URLs, pictures, etc. [20]. Alghamdi et al. analyzed and evaluated the detection of malicious URLs in existing social networks and explained that the future detection work needs to combine the characteristics of URLs with other aspects [21]. Han et al. designed a honeypot system that for the first time analyzed the entire life cycle of a phishing attack and clearly identified victims from the attackers and other third-party visitors [22]. Some attackers utilize user profile and social relationships in a collective manner to predict sensitive information of related victims in a released social network dataset. To protect against such attacks, Cai et al. propose a data sanitization method collectively manipulating user profile and friendship relations. Besides sanitizing friendship relations, the proposed method can take advantage of various data manipulating methods [23].

Existing detection methods mainly aim at detecting attacks in social network by a series of abnormal states of accounts, the content of messages, or the social relationship of users. There are little related works in the early stages of attacks in social network [20, 23–25], and our approach is primarily aimed at the reconnaissance stage of the attack. At this stage, the attacker collects and analyzes users' information and selects the appropriate attack target and method. We analyze the personal information and social behaviors of users in social networks from the perspective of attackers and extract relevant features. Based on these features, we have established a user analysis model to perform security scoring for each user and provide guidance for preventing attacks.

3. Overview

3.1. Problem Definition. A successful cyberattack usually collects and analyzes the information of target before the target is attacked. The process of the attacker collecting the information of targets is called the reconnaissance stage. Social network is one of the main ways for attackers to collect the information of target. Most of existing detection methods detect attack by analyzing abnormal activities in the host or network. Our work aims at finding the potential victims by analyzing the users behaviors in social networks and provides guidance for the prevention of cyberattacks. From the perspective of the attacker, we analyze users' behaviors and then build a user analysis model to assess the security risks of users in social networks. Through analyzing by the model, each user can get a security risk score; the higher the score is, the more likely it is to be attack.

3.2. User Analysis Model. To get a security score of each user in the social network, we design a model based on machine learning algorithms. As Figure 1 shows, this model consists of six submodules.

Data Collection. We use crawlers to collect data from Sina Weibo API, including their social relations, social behaviors, and microblogs. (More details are discussed in Section 8.)

Social-Graph Algorithm. To get the complexity of users' social circle, we establish a social-graph based follow relationship. We use the hierarchical clustering to get the number of clusters formed by the user's follow and then calculate the average cluster density of each user.

Microblogs Classification System. In order to get the obviousness of users' preference, we build a microblogs classification

system. We classify the user's microblogs topic based on the topic classification of microblogs and get a topics list for each user. According to the users' topic list, we can obtain the obviousness of users' preference.

Features Extraction. We extract three kinds of features, i.e., statistical features, social-graph features, and semantic features. After obtaining these features, we standardize them and prepare for training.

Training. We use the features extracted previously as input and train classifiers by different machine learning algorithms. By comparing the classification effects of each classifier, we choose a classifier with best performance as the classifier of our model.

Classification. We use the classifier trained by training model to classify the data in test dataset and use the probability that the sample is a positive sample as the user's security risk score.

4. Statistical Features

In this section, we analyze the users with what kind of features are more likely to attract the attention of attackers in the dataset from the perspective of attackers. Through statistics, we found two features that can distinguish positive samples from negative samples.

4.1. Activity Level. We make an observation that positive samples are more active in the social network and based on this observation, we design a feature to show the activity level of samples.

From the perspective of the attacker, we will select a user who is often active in social networks as the attack target, rather than a user who rarely uses social networks. Active users generate more information and behavior in social networks, making it easier for attackers to launch target attacks. If a user posts, forwards, comments on, or likes a microblog in Weibo, we record it as one times active_behavior, and we gather statistics about the total number of active behaviors for each user in total days (for our dataset, the value of total_days is 134) and use total activity_times to store the statistical results. We use variable avg_active_times to represent the average number of active_behaviors for a user in total days, which can be calculated by formula (1); the bigger the value of avg_active_time of a user is, the higher the activity level of it is.

$$\text{avg\_active\_times} = \frac{total\_active\_times}{\text{total\_days}} \quad (1)$$

Through statistics of the avg_active_times of each user, we plot Figure 2; the x-axis represents the value of avg_active_times and the y-axis represents the value of CDF corresponding to the value of x-axis. From the figure, we can see that the curve of avg_active_times of positive samples is in the right of negative samples; that is, the avg_active_times of positive samples are overall larger.

4.2. Frequency of Interactive Behaviors. We make an observation that positive samples have more interactive behaviors

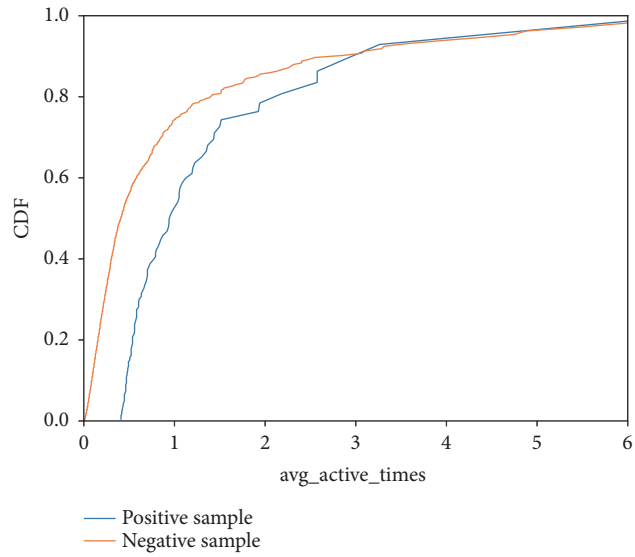


Figure 2: The active level of users.

with other users and based on this observation, we design a feature to show samples' frequency of interactive behaviors.

When the attacker establishes a relationship with the victim, the attacker can send microblogs based on the user's preferences. Once the user browses and clicks on the microblogs, it will be attacked. If the user interacts with other users very frequently, it is very likely to click the microblogs sent by attackers. We define the average number of interactions for each user as avg _interaction_times, which describes the possibility of the user making interactive behaviors with other users. Through statistics on the number of users' forwarding, comment, and like behaviors in the dataset during 134 days, we can get the avg _interaction_times for each user according to formula (2).

$$\text{avg\_interaction\_times} = \frac{\text{total\_interaction\_times}}{total\_days} \quad (2)$$

Based on statistical results, we plot the CDF figure of avg _interaction_times. As shown in Figure 3, the x-axis represents the value of avg _interaction_times and the y-axis represents the CDF value corresponding to the value of x-axis. From the figure, we can see that the average number of interactions per day of the positive samples is bigger than that of the negative samples overall.

5. Social-Graph Features

In this section, we analyze the social relationship of users in our dataset and use the follow relationships between users to establish a follow graph G. Then we get two features that can distinguish the positive samples from negative samples through analyzing the graph.

5.1. Probability of Following Back. We make an observation that positive samples have a higher probability of following back the user who follows him and based on this observation,

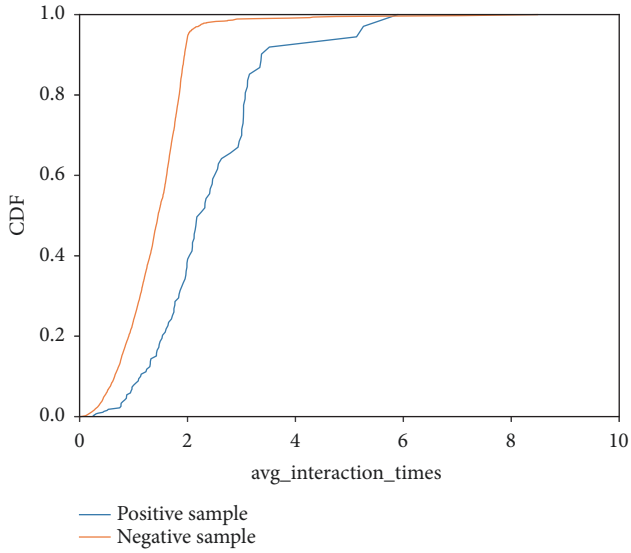


Figure 3: Average number of interactions of users.

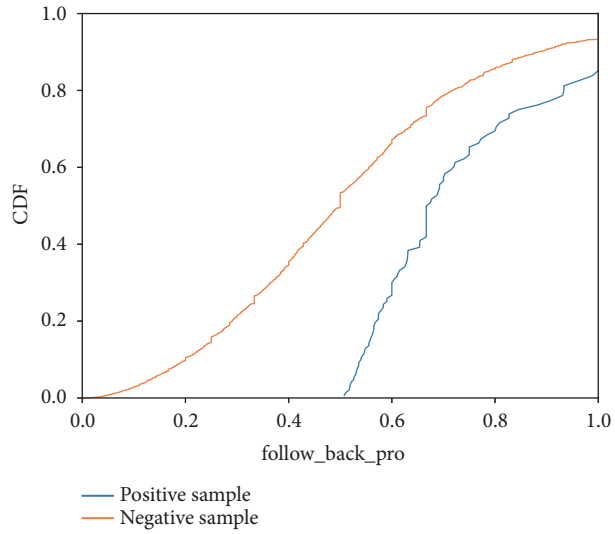


Figure 4: The probability of following back of users.

we design a feature to show samples' probability of following back.

Some attackers wish to establish mutual trust relationships with users. The process of establishing relationships is usually the attacker follows the target user first, and then the user may follow the attacker in reverse. Once the bidirectional relationship is established, the attacker can push the malicious information to target user and then launch attack and penetration. We use follow_back_pro to represent the probability of following back of a user. A user with higher follow_back_pro means that the attacker is under a higher probability of successfully establishing mutual trust relationship with him/her. The value of follow back pro can be computed by formula (3), that is, the ratio of the number of bidirectional edges (number_of_bi_edges) to the in_degree of the point in the social follow relationship graph.

$$\text{follow\_back\_pro} = \frac{number\_of\_bi\_edges}{in\_degree} \tag{3}$$

Based on the statistical result, we plot Figure 4, in which the x-axis represents the value of follow_back_pro,and the y-axis represents the value of CDF corresponding to the value of x-axis. We can discover that the CDF curve of follow_back_pro of the positive samples is over larger than negative samples'.

5.2. Complexity of Social Circle. We make an observation that the social circle of positive samples is more complex than negative samples' and based on this observation, we design a feature to show the complexity of social circle of samples.

The members of the user follow list often come from different social circles. We analyze each member and divide the members with common follows into the same social circle. In the follow graph G, for a user U_i (i is a user number), we use the BFS algorithm to find all its neighbors and store these nodes in a set C_i. Then we traverse all the nodes in C_i, find all the neighbors of each node, and put them in the corresponding set. We define the similarity between two users as the coincidence degree of their follow list. The higher the coincidence degree is, the closer the user is in the graph G. For each user U_i, we compute the similarity of any two nodes in all its neighbors. For N_j, $N_k \in C_i$, we can get corresponding sets C_j and C_k. We utilize Jaccard similarity coefficient (it can be computed by formula (4)) to calculate the similarity of two sets as the similarity of N_j and N_k. Finally, for each node, we obtain an $n*n$ symmetric matrix M_i with all elements in main diagonal being equal to 1, where n is the number of neighbors of the U_i, satisfying $J_{pq} = J_{qp}.(1 \leqslant p, q \leqslant n)$.

$$\text{Jaccard}_{C_j,C_k} = \frac{C_j \cap C_k}{C_j \cup C_k} \tag{4}$$

$$M_i = \begin{bmatrix} 1 & J_{12} & \cdots & J_{1n} \\ J_{21} & 1 & \cdots & J_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ J_{n1} & J_{n2} & \cdots & 1 \end{bmatrix} \tag{5}$$

For each node, we use the corresponding similarity matrix as input and cluster each node using the agglomerative hierarchical clustering algorithm. The process first takes each node as a cluster and then calculates the Euclidean distance between any two clusters, merging the nearest cluster each time. For the distance between two clusters, we use the complete-linkage algorithm which takes the distance between the farthest nodes of two clusters as the distance between two clusters. We define a threshold β and terminate clustering when the distance between any two clusters is greater than β.

β: The termination condition of the algorithm. If the Jaccard similarity coefficient of a node with any other nodes is 0, that is, there is no identical user in the follow list, we say that it is an isolated node. We calculate the distance between

two isolated points as the value of β. When the clustering algorithm terminates, there are common follow nodes of any two nodes in each cluster.

Assume there are two isolated nodes:

$$C_A = [1, 0, 0, \ldots, 0]$$
$$C_B = [0, 1, 0, \ldots, 0] \quad (6)$$

we use the Euclidean distance formula to calculate the value of β as follows:

$$\beta = \sqrt{(1-0)^2 + (0-1)^2 + (0-0)^2 + \cdots + (0-0)^2} = \sqrt{2} \quad (7)$$

After getting the number of clusters for each node, we define an average cluster density avg_clu_density. After clustering, the users from the same social circle in the user's follow list will eventually be in the same cluster. If the user's social circles are very simple, such as a person's social scope is limited to classmates, colleagues, relatives, etc., then only a few clusters will eventually be obtained; each cluster contains a large number of nodes; that is, the average cluster density is very large. On the contrary, the more complex the user's social circle is, the more the clusters are formed, and the smaller the avg_clu_density of the node is.

We calculate avg_clu_density of each user by formula (8) and plot the CDF figure of avg_clu_density based on the experimental results. As shown in Figure 5, the x-axis is the value of average cluster density and the y-axis represents the CDF value corresponding to the value of x-axis. From the figure, it can be clearly seen that the avg_clu_density of the positive samples is smaller than negative samples overall.

$$\text{avg\_clu\_density} = \frac{total\_neibornodes}{number\_of\_cluser} \quad (8)$$

6. The Semantic Feature

In this section, we analyze the semantic of microblogs of positive and negative samples. Through analyzing, we get a feature that can distinguish the positive samples from negative samples.

6.1. Problem Statement. Whether an attacker establishes a trust relationship with a user or launches a water hole or spear-phishing attack, the attacker is required to collect user information, analyze the user's preferences, and then increase the attack success probability through targeted attacks. Therefore, we assume that the more obvious the user's preferences are, the easier it will be a target of the attacker. To prove this assumption, we first establish a microblogs classification system to classify microblogs of samples in dataset. Then, we make statistics on the results and get a topic list for each user. Finally, we get a value that can measure the obviousness of preference of users.

Based on the existing classification of Weibo [26], we divide all microblogs into 23 categories according to the different topic, as shown in Table 1. For these 23 categories of microblogs, we collected a total of 311,731 microblogs and used them as corpus. We establish a microblog classification system, as shown in Figure 6. After formatting the microblogs in the corpus, we take them as input to the system, and then use Google's word2vec tool to get a word vector conversion model, Word2vec Model. Then we convert all the microblogs in the corpus into vectors and used Google's deep learning framework TensorFlow to build a convolutional neural network (CNN) for training and classification. Finally, we got a microblog classification system with an accuracy of 0.87. We used the visualization tool TensorBoard to visualize the accuracy and loss function in the training process, as shown in Figures 7(a) and 7(b).

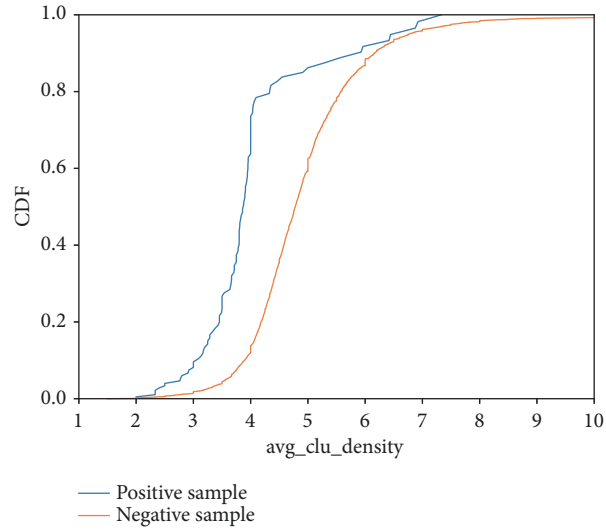


Figure 5: The average cluster density of users.

6.2. Obviousness of Preference. After the microblogs classification model was obtained, we used this model to classify 88,064 microblogs sent in 134 days by the 4,536 users in the dataset. For each user U_i, we put n microblogs sent by U_i into a Weibo list L_i. After classification, we get a topic list T_i with a length of 23, where t_i represents the number of occurrences of the topic with the number i in the user's Weibo list L_i. Then we use formula (9) to count the proportion of microblogs on each topic sent by the user to the total number of microblogs sent to get the list of F_i.

$$L_i: [W_1, W_2, W_3, \ldots, W_n] \xrightarrow{\text{classification}}$$
$$T_i: [t_1, t_2, t_3, \ldots, t_n] \xrightarrow{\text{statistics}}$$
$$F_i: [f_1, f_2, f_3, \ldots, f_n] \quad (9)$$
$$f_i = \frac{t_i}{n}$$

Obviously, if user U_i sends a much larger number of microblogs with the topic t_i than other topics, f_i is large, then we think that the user has an obvious preference. Since the

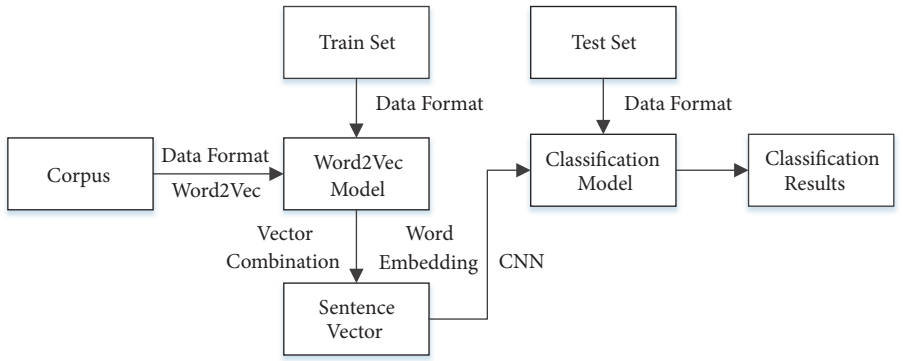
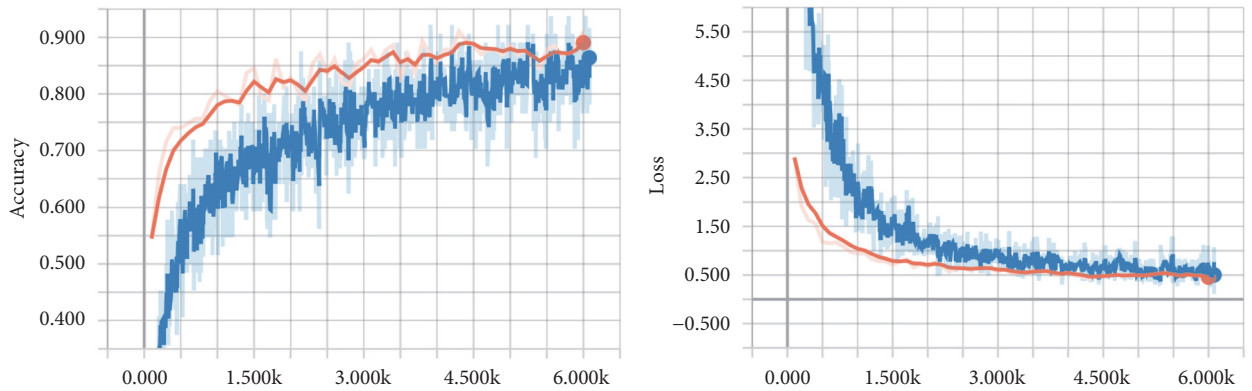


Figure 6: Microblogs classification system.



(a) The accuracy of microblogs classification system

(b) The loss function of microblogs classification system

Figure 7: The accuracy and loss function of microblogs classification system.

user may have multiple preferences at the same time, it is not suitable to simply compare the obviousness of preference of users by taking the maximum value. We sort the list F_i in the order from large to small, and we will leave out the 0 elements in the list. We use two users U_1 and U_2 as an example, and corresponding sorted F_1 and F_2 are as follows:

$$F_1 = [0.27, 0.10, 0.09, 0.08, 0.07, 0.07, 0.07, 0.06, 0.04, 0.03, 0.02, 0.02, 0.02, 0.02, 0.01, 0.01, 0.01, 0.01]$$
$$F_2 = [0.23, 0.18, 0.14, 0.14, 0.09, 0.09, 0.05, 0.05, 0.05] \quad (10)$$

We plot the experimental results as a histogram figure, where the x-axis indicates the order in which the number of microblogs topics appears from high to low and the y-axis indicates the percentage of microblogs sent with the corresponding topic by the user. Figure 8(a) shows the distribution of the topics of microblogs of user U_1, and Figure 8(b) shows the distribution of the topics of microblogs of user U_2. From the figure, we can see that the more obvious the user's preferences, the steeper the figure. In order to quantify this steepness, we make an axisymmetric mapping of the original figure with respect to x=0, resulting in Figures 9(a) and 9(b). Observing the figures, we found that the distribution is similar to a normal distribution. Normal distribution curve formula is as follows:

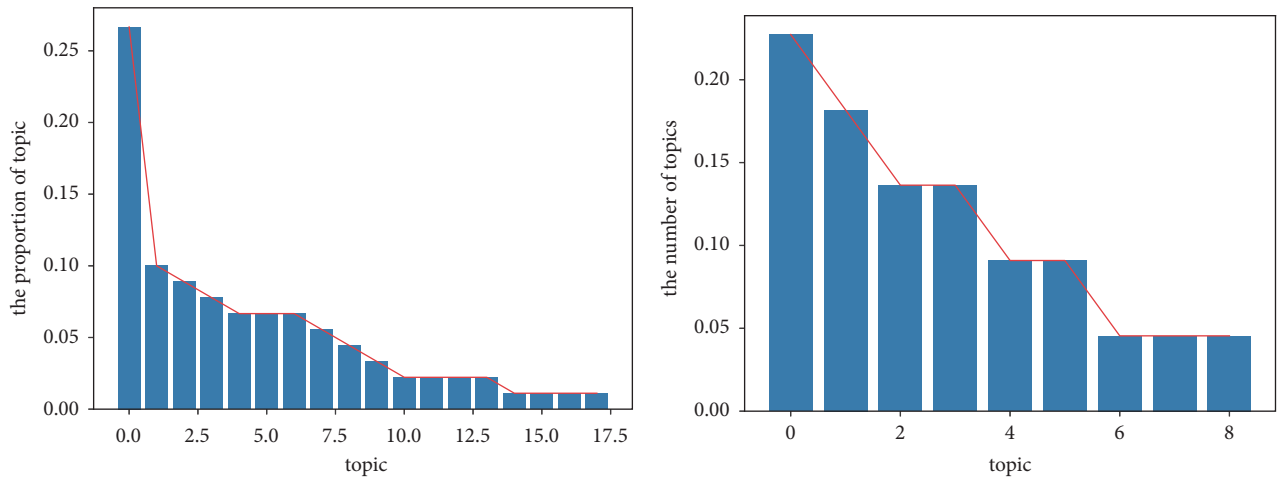
$$\varphi_{\mu,\delta(x)} = \frac{1}{\sqrt{2\pi}\delta} e^{-(x-\mu)^2/2\delta^2} \quad (11)$$

μ is the average of overall samples, which reflects the average level of the overall random variable.

δ is the standard deviation of overall samples, which reflects the degree of concentration and dispersion of the overall random variable.

Since we do an axisymmetric mapping for $X = 0$, according to the nature of the normal distribution function, when $\mu = 0$, the formula is

$$\varphi_{\delta(x)} = \frac{1}{\sqrt{2\pi}\delta} e^{-x^2/2\delta^2} \quad (12)$$



(a) The distribution of the topics of microblogs of user U_1

(b) The distribution of the topics of microblogs of user U_2

Figure 8: The distribution of topics.

Table 1: The topics of Weibo.

#	Topic
1	Science and technology
2	Art
3	Sports
4	Finance and economics
5	Film and television
6	Emotion
7	Tourism
8	Current affairs
9	Music
10	Game
11	Health
12	Stars
13	Fashion
14	Home
15	Fun
16	Campus
17	Pet
18	Lucky draw
19	Constellation
20	History
21	Food
22	Military
23	Anime

According to the properties of the normal distribution function, when μ is constant, the smaller the value of δ, it means that the distribution of random variable is concentrated near μ, and the curve is higher and narrower. The larger the value of δ is, the more scattered the random variable distribution is, and the lower and the wider the curve is. We use the 1stopt tool to fit the image to a normal distribution curve. To prevent the maximum point from being ignored as a noise point, we specify that the fitted curve passes through the maximum point. As shown in the figure, δ of Figure 10(a) is 0.016, and δ of Figure 11(a) is 0.079. We obtain Figures 10(b) and 11(b) by partially amplifying the coordinate axes, respectively. From the figure, we can clearly see that the smaller the value of δ, the higher and the narrower the curve. Therefore, we use δ to measure the obvious degree of user preference. The smaller the δ, the more obvious the user's preferences.

Through experiments, we draw the CDF figure of standard deviation. As shown in Figure 12, the x-axis is the standard deviation, and the y-axis is the cumulative distribution function value of the corresponding standard deviation. As can be seen from the figure, the standard deviation of the negative sample is relatively small. We believe that users with more obvious preferences are most likely to be targeted by attackers.

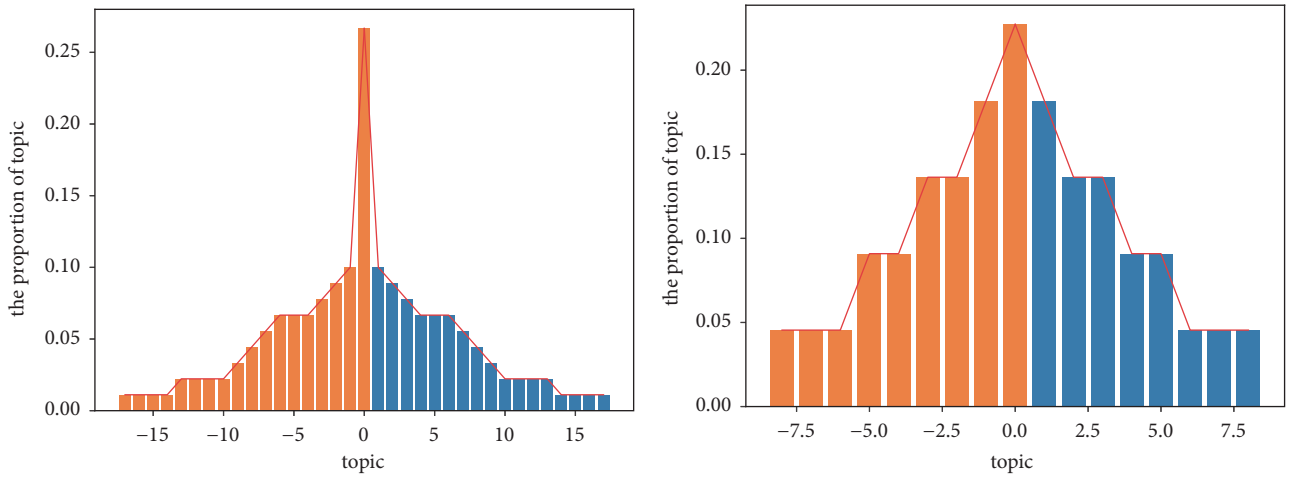
7. The Semantic Feature

In this section, we first standardize the extracted features.

After that, we will explain how we train the model and obtain the security risk score.

7.1. Standardization of Features. Data normalization processing is a basic work of data mining. Different evaluation indexes often have different range of values, which will affect the results of data analysis. To eliminate the effect of different range of values between features, data standardization is needed to resolve the comparability of data indicators. After the original data has been standardized by data normalization, each indicator is in the same level of magnitude, which is suitable for comprehensive comparative evaluation.

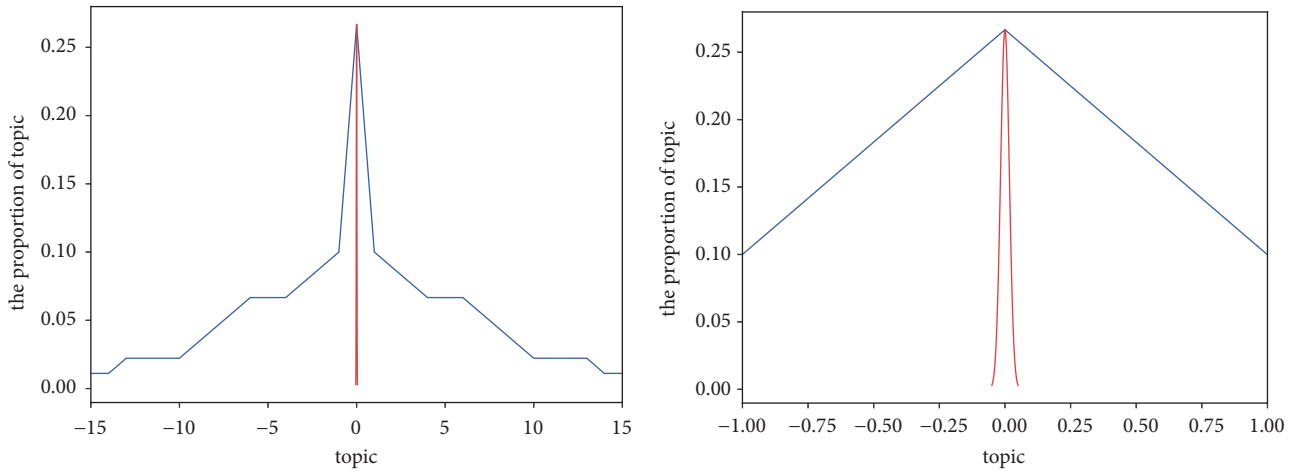
We use the machine learning method in the detection model. Since zero-mean normalization performs better because distances are used to measure similarity in classification and clustering algorithms, we use zero-mean normalization to normalize all features. The results can be



(a) The axisymmetric mapping of Figure 8(a)

(b) The axisymmetric mapping of Figure 8(b)

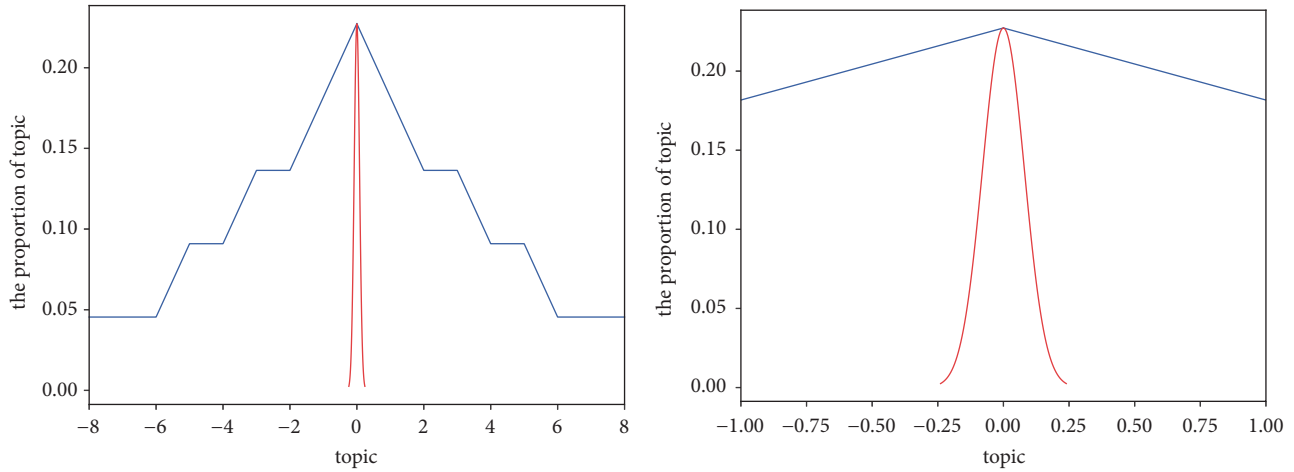
Figure 9: The axisymmetric mapping of figures.



(a) The normal distribution curve with $\delta = 0.016$

(b) Partially amplifying the coordinate axes of Figure 10(a)

Figure 10: Normal distribution curve of F_1.



(a) The normal distribution curve with $\delta = 0.079$

(b) Partially amplifying the coordinate axes of Figure 11(a)

Figure 11: Normal distribution curve of F_2.

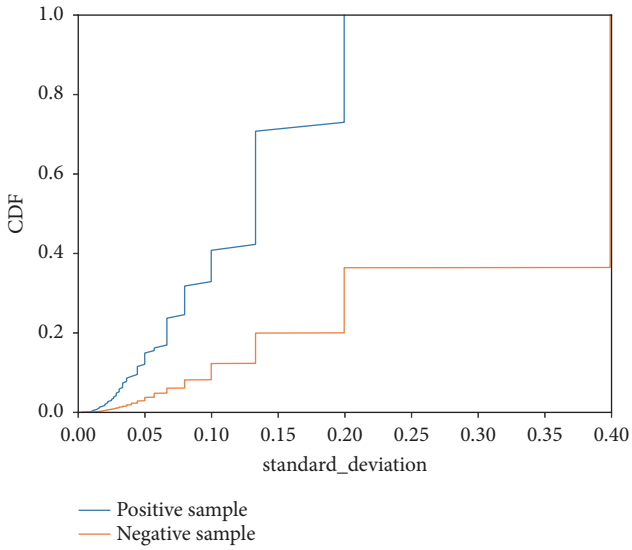


Figure 12: The standard deviation of users.

calculated by formula (13) and the normalized results are shown in Table 2.

$$z = \frac{x_i - \mu}{\delta} \tag{13}$$

μ is the average of all sample data.

δ is the standard deviation of all sample data.

z is the normalized feature value.

7.2. Training and Prediction. We select Gradient Boosting Decision Tree (GBDT) algorithm to train our classification model. GBDT is a method to make joint decisions by iterating multiple decision trees and the core of this method is that each tree learns the residual of sum of all the previous tree results, which are the sum of the real values after adding the predicted value. The biggest advantage of using this method is that each step of calculation of residual actually increases the weight of misclassified samples, and the weight of correctly classified samples tends to 0. In this way, the current tree can focus more on the samples that are misclassified by the previous method. By this method, we prioritize the users that are correctly classified and then iterate again for the users that are misclassified, so as to achieve the correct classification of all users as far as possible.

After training the model, we test our model on the test dataset. For each test sample, we obtain a two-tuple (p_0, p_1) through our model, where p_1 is the probability that the sample is a positive sample and p1 is the probability that the sample is a negative sample. Finally, we take the value of p_0 as the security risk score of the user.

8. Evaluation

In this section, we will introduce the structures of user analysis model and our dataset which can be collected from Weibo firstly. Then we compare the accuracy of different machine learning algorithms and verify the validity of our model. Finally, we rank the features using model based ranking.

Table 2: Standardization of features.

Feature	Feature	Feature
F1: Activity level	[−1, 1]	The closer the value is to 1, the more activities the user has in social networks.
F2: Reverse follow probability	[−1, 1]	The closer the value is to 1, the more the probability of following back the user is in social networks.
F3: Social circle complexity	[−1, 1]	The closer the value is to 1, the more complex the user's social circle is in social networks.
F4: Interaction behavior frequency	[−1, 1]	The closer the value is to 1, the more the frequency of interactive behaviors of the user is in social networks.
F5: Obvious degree of preference	[−1, 1]	The closer the value is to 1, the more the frequency of interactive behaviors of the user is in social networks.

8.1. Standardization of Features. Due to the limitation of the official privacy policy, collecting data from OSNs is still a challenge to researchers. OSNs protect their API carefully; for example, Twitter's API restrict methods depending on the type of requests. The limitation of crawling Twitter users basic information is 180 times every 15 min, but the users' followers can only get 15 times every 15 min. We are trying to collect more data for our future research of OSNs by the official API as far as possible.

Our data source is Weibo, which is one of the most popular social media platforms in China. In previous work [15], they have collected 12,941 Weibo user pieces of information and manually labeled 2,404 abnormal users who had sent at least one microblog containing malicious URLs. We get this dataset and recollect information based on the user IDs of these abnormal accounts. We eliminate the accounts that have been closed by official and have not been active for a long time. After manual judgment, we get 107 users' information. These users are all normal users who had one time malicious behaviors at least. We use these 107 accounts as positive samples. We collect information from normal users and obtain 4,412 negative samples with the same processing.

Our dataset contains 4,536 pieces of user information, including 4,429 negative samples and 107 positive samples. For each user, we collected his/her follow list and once again collected the follow list of the follow list, forming a follow relationship graph containing 5,931,527 points. In addition, we have collected follower list and 134 days' microblogs (2018.01.01-2018.05.14), a total of 88,064 microblogs, of the 4,536 users. In order to classify the user's microblog content,

Table 3: Confusion matrix.

	Predicted	
	Positive	Negative
Actual		
Positive	TP (True Positive)	FN (False Negative)
Negative	FP (False Positive)	TN (True Negative)

we have collected a total of 311,731 microblogs in 23 categories based on Weibo's classification system. We use these microblogs as a corpus for text classification training.

8.2. Performance of Different Classifiers. Our evaluation environment is a Dell OptiPlex 7040 computer. This computer is populated with Inter(R) Core(TM) i7-6700 CPU @ 3.40GHz 3.41GHz, 16 GB memory, a 2237.0-GB hard-disk and connected by a 1000-Mbit Ethernet.

Oversampling. Our dataset has 4,536 users, including 4,429 negative and 107 positive samples. Due to the imbalance of positive and negative samples distribution, we transform the training set from an unbalanced dataset into a balanced dataset. We can see that the number of positive samples is significantly less than the number of negative samples, so we copy multiple positive samples and add a slight random disturbance each time when we generate a new sample. Finally, we get a dataset containing 4429 positive samples and 4429 negative samples and use it as the experimental dataset.

We use different machine learning algorithms (MLAs) and 5-fold cross validation method to train and test the model. According to the confusion matrix (Table 3) and other evaluation parameters as follows, we compare the results of different machine learning algorithms.

Ture Positive Rate (TPR)

$$\text{TPR} = \frac{TP}{TP + FN} \tag{14}$$

False Positive Rate (FPR)

$$\text{FPR} = \frac{FP}{FP + TN} \tag{15}$$

Precision

$$\text{Precision} = \frac{TP}{TP + FP} \tag{16}$$

$F_1 Score$

$$F_1 Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{17}$$

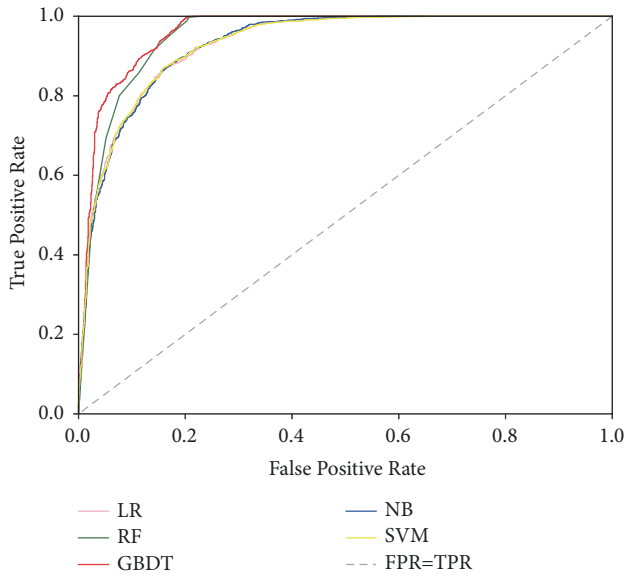


Figure 13: The ROC curves of different classifiers.

Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{18}$$

The machine learning algorithms we used include Logistic Regression (LR), Random Forest (RF), Gradient Boosting Decision Tree (GBDT), Naive Bayesian (NB), and Support Vector Machines (SVM). According to the confusion matrix and other evaluation metrics, we have drawn Table 4 to show the experimental results.

From Table 4, we can see that the classifiers based on our proposed features have higher TPR, which means the classifiers can identify positive samples well. By observing the Precision, Recall, and F-measure of five classifiers, their values are higher, indicating that our classifiers are very robust.

FPR. The FPR of the classifiers is higher, indicating that our classifiers determine many negative samples as positive samples. We analyze the cause of high FPR and make the following assumptions. (1) Some users corresponding to these FP samples have high awareness of network security and they are highly alert to strangers or suspicious information. Although these users have rich social behaviors and complex social circles, they have the ability to prevent attacks. (2) Some users may be potential victims who have not been attacked. Attackers have a high rate of success if they attack these users. Although these users are not in our positive samples dataset, we believe that these users need to improve their awareness of network security in order to prevent future attacks.

8.3. The Validity of Classifiers. In order to verify the effectiveness of our user analysis model, we plot the ROC curve for each classifier. As Figure 13 shows, the x-axis is the FPR of the classifier and y-axis is the TPR corresponding to the FPR. As

Table 4: Experimental results of using different classifiers.

MLAs	TPR	FPR	Precision	F_1Score
LR	86.85%	16.38%	84.25%	85.53%
RF	95.50%	17.32%	84.74%	89.80%
GBDT	98.65%	19.16%	83.85%	90.65%
NB	88.31%	17.74%	83.39%	85.78%
SVM	87.64%	16.78%	84.05%	85.80%

Table 5: The AUC and accuracy of different classifiers.

MLAs	AUC	Accuracy
LR	0.9312	0.8524
RF	0.9515	0.8910
GBDT	0.9607	0.8978
NB	0.9303	0.8529
SVM	0.9308	0.8544

Table 6: The rank score of features.

#	Feature name	Rank score
F1	Activity level	0.181
F2	Reverse follow probability	0.175
F3	Social circle complexity	0.390
F4	Interaction behavior frequency	0.154
F5	Obvious degree of preference	0.516

we can see, the ROC curves for these classifiers are all above the x = y, which prove that our classifiers are valid.

To compare the classification effect of each classifier, we calculated the area under each ROC curve (AUC) and the accuracy of each classifier and record the calculation results in Table 5. The larger the value of AUC is, the more effective the classifier is. From Table 5 we can see that the average AUC value of the five classifiers is 0.94086 and the average accuracy of the five classifiers is 0.8697. Among these classifiers, the GBDT classifier has the best classification effect compared to other classifiers.

After selecting the appropriate classifier, for each sample, we use the classifier to get the probability of a sample is positive or negative. We take the probability that the sample is a positive sample as the security score of the sample, which can provide guidance for preventing attackers from utilizing social networks to launch attacks.

8.4. Comparison. We compare the difference of detection accuracy between the model with one feature removed and the model with all features and plot Figure 14. In the figures, the x-axis is the machine learning algorithms of classifiers and y-axis is the accuracy corresponding to the classifier. As shown in Figure 14, each subfigure corresponds to a comparison model with one feature removed. Such as Figure 14(a), the comparison model trained without the feature activity level (F1), the blue bar is the model trained with F1, and the chocolate bar is the model trained without F1. We can see the detection accuracy of the model without F1 decrease obviously. By observing all subfigures of Figure 14, we can make a conclusion that the accuracy of the model with our newly identified features is higher than the models without our newly identified features.

Using all the features, our model can make a significant assessment for users' security risk in the social network. However, different feature plays different detection roles. To show the weight of each feature, we rank them using model based ranking. Through using the machine learning algorithm, we can build a prediction model for each individual feature and response variable directly. The rank results of the features are shown in Table 6. We can obverse that the features F3 and F5 are ranked higher than other features, which means that the weight of F3 and F5 is larger than other features' weight. In summary, our newly identified features are effective in assessing the risk of users' activities in social networks.

9. Conclusion

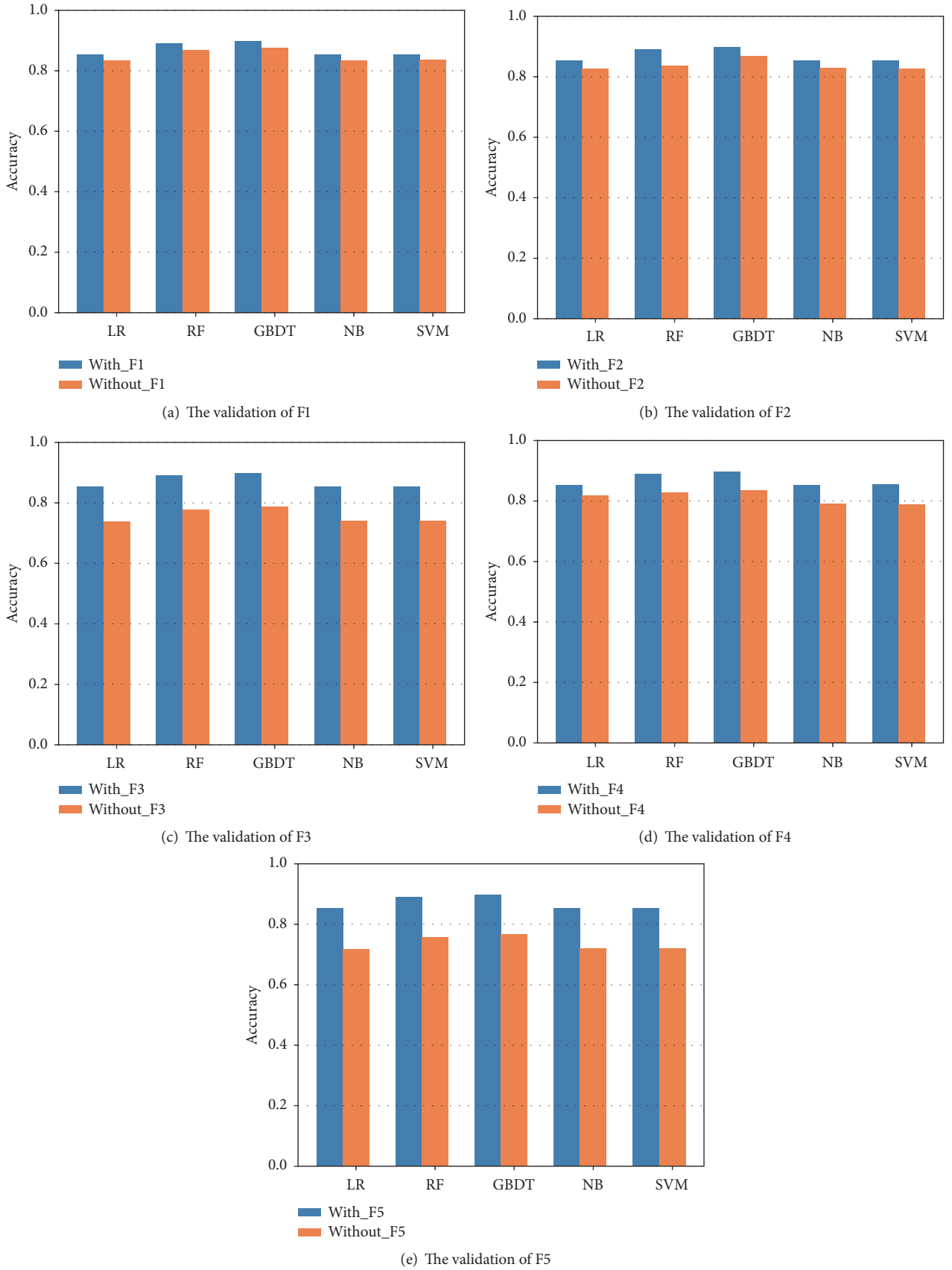
While enjoying the convenience of social networks, users also leave a large amount of personal information on social networks. The attacker can carry out more targeted cyberattacks by collecting personal information and social behaviors of the user in the social network, thereby greatly increasing the probability of success of the attack. We propose a novel user analysis model to identify potential victims by analyzing the users' personal information and social behaviors. We extract five features and use different machine learning algorithms to train our model. Finally, we chose the one that worked best. The security scores of users can guide company to prevent the cyberattack from social network, so as to reduce the harm caused by attacks. For future work, we will continually improve the performance of our model. Apart from this, we are sincere to communicate with other researchers working in this field.

Data Availability

The data used to support the findings of this study have not been made available because the data involves privacy protection. We cannot share the raw data outside our group.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.



(a) The validation of F1

(b) The validation of F2

(c) The validation of F3

(d) The validation of F4

(e) The validation of F5

Figure 14: The accuracy comparison with and without our newly identified features.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant No. 61772229 and No. 61472162; the Scientific and Technological Research Planning Projects in Colleges and Universities of Jilin Province under Grant No. JJKH20190168KJ.

References

[1] S. weibo data center, "Weibo user development report," 2017, http://data.weibo.com/report/reportDetail?id=404.

[2] J. Min, "North korean defectors and journalists targeted using social networks and kakaotalk," https://securingtomorrow.mcafee.com/mcafee-labs/north-koreandefectors-journalists-targetedusing-social-networks-kakaotalk/, 2018.

[3] K. Thakur, T. Hayajneh, and J. Tseng, "Cyber security in social media: challenges and the way forward," *IT Professional*, vol. 21, no. 2, pp. 41–49, 2019.

[4] C. Sky, "Charming kitten," 2017, http://www.clearskysec.com/charmingkitten/.

[5] M. A. H. Ghareeb Saad, "The desert falcons targeted attacks," https://securelist.com/thedesert-falcons-targeted-attacks/68817/, 2015.

[6] U. U. Khan, M. Ali, A. Abbas, S. Khan, and A. Zomaya, "Segregating spammers and unsolicited bloggers from genuine experts on Twitter," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 551–560, 2018.

[7] J. Cao, Q. Li, Y. Ji, Y. He, and D. Guo, "Detection of forwarding-based malicious URLs in online social networks," *International Journal of Parallel Programming*, vol. 44, no. 1, pp. 163–180, 2016.

[8] N. Griffin, "Monsoon - analysis of an apt campaign," 2016, https://www.forcepoint.com/zhhans/blog/security-labs/monsoonanalysis-apt-campaign.

[9] D. Regalado, N. Villeneuve, and J. S. Railton, "Behind the Syrian conflict's digital front lines," *FireEye*, 2015.

[10] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro, "Aiding the detection of fake accounts in large scale social online services," in *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, p. 15, USENIX Association, 2012.

[11] N. Z. Gong, M. Frank, and P. Mittal, "SybilBelief: a semi-supervised learning approach for structure-based sybil detection," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 6, pp. 976–987, 2014.

[12] D. Mulamba, I. Ray, and I. Ray, "Sybilradar: a graphstructure based framework for sybil detection in online social networks," in *Proceedings of the IFIP International Information Security and Privacy Conference*, pp. 179–193, Springer, 2016.

[13] S. Lee and J. Kim, "Detecting suspicious urls in twitter stream," in *Proceedings of the Network and Distributed System Security (NDSS)*, vol. 12, pp. 1–13, 2012.

[14] J. Cao, Q. Fu, Q. Li, and D. Guo, "Discovering hidden suspicious accounts in online social networks," *Information Sciences*, vol. 394-395, pp. 123–140, 2017.

[15] Q. Fu, B. Feng, D. Guo, and Q. Li, "Combating the evolving spammers in online social networks," *Computers & Security*, vol. 72, pp. 60–73, 2018.

[16] K. Krombholz, H. Hobel, M. Huber, and E. Weippl, "Advanced social engineering attacks," *Journal of Information Security and Applications*, vol. 22, pp. 113–122, 2015.

[17] ISACA, "Advanced persistent threat awareness study results," 2015, http://www.isaca.org/knowledge-center/research/researchdeliverables/pages/advanced-persistent-threatsawareness-study-results.aspx.

[18] ISACA, "TrendMicro. Advanced persistent threat awareness study results," 2013, http://www.trendmicro.it/media/report/aptawareness-isaca-survey-report-en.pdf.

[19] K. Coronges, R. Dodge, C. Mukina, Z. Radwick, J. Shevchik, and E. Rovira, "The influences of social networks on phishing vulnerability," in *Proceedings of the 2012 45th Hawaii International Conference on System Sciences, HICSS 2012*, pp. 2366–2373, IEEE, USA, January 2012.

[20] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna, "Compa: detecting compromised accounts on social networks," in *Proceedings of the Network and Distributed System Security (NDSS)*, 2013.

[21] B. Alghamdi, J. Watson, and Y. Xu, "Toward detecting malicious links in online social networks through user behavior," in *Proceedings of the 2016 IEEE/WIC/ACM International Conference on Web Intelligence Workshops, WIW 2016*, pp. 5–8, IEEE, USA, October 2016.

[22] X. Han, N. Kheir, and D. Balzarotti, "PhishEye: live monitoring of sandboxed phishing kits," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1402–1413, ACM, Austria, October 2016.

[23] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 577–590, 2018.

[24] A. Paradise, A. Shabtai, R. Puzis et al., "Creation and management of social network honeypots for detecting targeted cyber attacks," *IEEE Transactions on Computational Social Systems*, vol. 4, no. 3, pp. 1–15, 2017.

[25] J. Nelson, X. Lin, C. Chen, J. Iglesias, and J. J. Li, "Social engineering for security attacks," in *Proceedings of the Multidisciplinary International Social Networks Conference on Socialinformatics ACM*, New York, NY, USA, 2016.

[26] S. Weibo, "The classification of weibo," 2018, https://d.weibo.com/.