applied sciences

MDPI

Article

An Attention-Based Graph Neural Network for Spam Bot Detection in Social Networks

Chensu Zhao [1,2,3], Yang Xin [1,2,*], Xuefeng Li [1,2], Hongliang Zhu [1,2], 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; zhao-cs@bupt.edu.cn (C.Z.); lxf3710@bupt.edu.cn (X.L.); zhuhongliang@bupt.edu.cn (H.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, Shandong 270104, China
* Correspondence: yangxin@bupt.edu.cn

Received: 13 October 2020; Accepted: 16 November 2020; Published: 18 November 2020

Abstract: With the rapid development of social networks, spam bots and other anomaly accounts' malicious behavior has become a critical information security problem threatening the social network platform. In order to reduce this threat, the existing research mainly uses feature-based detection or propagation-based detection, and it applies machine learning or graph mining algorithms to identify anomaly accounts in social networks. However, with the development of technology, spam bots are becoming more advanced, and identifying bots is still an open challenge. This paper proposes a new semi-supervised graph embedding model based on a graph attention network for spam bot detection in social networks. This approach constructs a detection model by aggregating features and neighbor relationships, and learns a complex method to integrate the different neighborhood relationships between nodes to operate the directed social graph. The new model can identify spam bots by capturing user features and two different relationships among users in social networks. We compare our method with other methods on real-world social network datasets, and the experimental results show that our proposed model achieves a significant and consistent improvement.

Keywords: social networks; spam bot detection; graph embedding; attention mechanism

1. Introduction

In recent years, online social networks (OSNs) such as Facebook and Twitter have become ever more popular with users, and they have become convenient service platforms for people to share and communicate. While people are immersed in the convenience and freshness brought by social networks, numerous malicious behaviors generated by anomaly accounts, such as advertising, malicious links and fake news, are common on social platforms [1].

The definition of an anomaly account in this article is a spam bot. Unlike other legitimate social bots, the spam bot is a type of account that uses automated programs to spread malicious, phishing or unsolicited content in social networks. The growing number of users and the open nature of social networks make them ideal targets for automated programs (Bots) [2]. As a result, spam bots are widely used to spread spam or malicious content. Manually managing a large number of spam accounts will cause high costs, so many spammers use the Twitter API to create custom programs to automatically publish spam tweets [3]. These spam bots use automated programs to launch various attacks on social networks. For example, to spread advertisements, post tweets containing pornographic information [4],

or hijack trend topics [5]. A recent study estimated that 9% to 15% of active Twitter accounts are spam bots [6]. Chen et al. [7] found that 10% to 50% of the tweets generated by seven popular URL shortening services on Twitter are always bots account. The malicious interaction behaviors or spam messages being generated by such spam bots seriously affect the trust and security of social platforms. Therefore, the effective identification of these spam bots has important practical significance in the development of OSNs.

Previous research on anomaly account detection has mainly focused on the detection of anomaly accounts or messages using a single feature set or simple relation structure. However, spam bots have become more advanced and complex in evading existing detection methods. Research has shown that social bots can interact with other accounts, post tweets on different topics, and display human-like activities [8]. Therefore, spam bots can use advanced intelligent methods to evade the existing anomaly detection system. For example, some spam bots release normal information most of the time, and occasionally release promotional activities, advertisements, spam, etc. Methods based on a single feature set (such as user behavior or content) have struggled to accurately detect anomaly users or events, especially spam bots. At the same time, bots have gradually evolved into swarm intelligences. The behavioral features of a single user may seem legitimate, but when multiple such users appear as a group, they will show anomaly features [8]. For example, a group of users can collaborate to post some fake comments or threatening activities on Twitter [6]. In large organizations, malicious teams or internal groups usually work closely together to achieve this malicious purpose.

There are still some challenges to addressing these problems. First of all, when detecting anomaly accounts and their behaviors in social networks by simply using features such as behavior/content, it is necessary to extract these features that can effectively separate anomaly accounts and legitimate accounts. However, spam bots usually pretend to be a legitimate account to evade the detection system. They usually imitate the behavior of legitimate users, such as the number and frequency of tweets. In addition, spam bots also apply complex social engineering techniques to manipulate information; secondly, the design idea of the graph structure based on the detection system is to assume that anomaly accounts struggle to effectively connect with many legitimate accounts in social networks. However, in order to obtain higher network popularity, even legitimate users have begun to buy fake bot accounts and indirectly add these anomaly users to their relationships. Therefore, a detection system based on this hypothesis may eventually misclassify unknown accounts on the social graph.

Therefore, it is not enough to consider the behavior/content features or interactive relationship of users unilaterally. We propose a new semi-supervised graph embedding a model based on a graph attention network, which detects spam bots by fusing various user's features and relational structures. The algorithm consists of three steps. Firstly, the bipartite graph of the following relationship and the retweet relationship between users is extracted from the social network. Secondly, the fusion model of the two kinds of neighborhood relations is build. Finally, an attention mechanism-based graph convolution neural network for spam bot detection is proposed.

The rest of the paper is organized as follows. Related work is described in Section 2. The proposed approach for spam bot detection is detailed in Section 3. Experiments and evaluations are presented in Section 4. Section 5 draws conclusions.

2. Related Works

This section reviews the related works from three aspects: spam bot detection, graph convolutional networks and attention mechanism.

2.1. Spam Bot Detection

The existing spam bot detection models can be divided into two categories: feature-based method and propagation-based method.

In the first method, the feature types include the profile feature, the user behavior feature, the network structure feature, and the tweet content feature. It usually depends on applying well-known machine

learning algorithms to the accounts to be detected. Kudugunta and Ferrara [9] extracted context features from user metadata and designed a deep neural network method based on an LSTM architecture to detect spam bots. Yang et al. [10] used four feature sets based on account, text, graph, and automation (e.g., frequency of usage related to twitter API), and three machine learning classification algorithms to identify bot accounts achieved good results. In the study of Cresci et al. [11], inspired by DNA sequences, according to the type of tweets, each account is modeled as a series of behavior data, and the corresponding "digital DNA" signature is generated. Account similarity is measured by calculating the length of the LCS. Loyola et al. [12] introduced a new feature model, which includes the Twitter account usage features and features other than sentiment analysis of tweet content. They proposed a classifier based on a contrasting pattern to realize the Twitter bot detection. BotOrNot [13] extracted more than 1000 features from a Twitter bot account divided into six categories. Then, the random forest classifier algorithm was used to detect the robot. Recently, Li et al. [14] used node2vec and doc2vec embedding the as input features of the text view and the social graph view, and proposed a semi-supervised model based on an automatic encoder framework. However, the model does not explicitly capture the interaction between users in the social graph.

Although these feature-based methods have achieved good results, they ignore the relationship structure between users in social networks. They are easy to be imitated by bot accounts to evade detection. According to our observation, various associations between users also play an important role in spam bot detection.

The association relationship between accounts in social networks has the nature of a graph. The graph-based method uses legitimate accounts and anomaly accounts to devise different structural modes or connection methods in the formed graph to convert anomaly account detection problems into the node classification problem in the graph, and then use the algorithm of graph mining to distinguish legitimate accounts and anomaly accounts.

Jia et al. [15] use a set of labeled anomaly nodes and/or legitimate nodes in social networks and then use random walks to propagate the label information in the graph to predict the remaining nodes' labels. Wang et al. [16] proposed propagation-based methods, which assume some correlation between two users and use Markov Random Field (MRF) for modeling. However, these methods cannot combine the features of tweet text. El-Mawass et al. [17] linked similar accounts based on shared applications. They built an MRF model on the similarity graph, which used the similarity between users to spread information about their label belief. Mulamba et al. [18] proposed a method to classify users based on the inherent topology or structure of the underlying OSNs graph.

The graph-based method uses the network information in social networks and the connection between accounts to detect bot activities. However, such methods cannot be extended to large-scale online social networks and are vulnerable to time attacks.

2.2. Graph Convolutional Networks

Recently, graph-based deep learning algorithms have been increasingly studied. Much research is devoted to extending the traditional convolutional neural network (CNN) that works in the Euclidean domain to arbitrary graphs [19]. Kipf et al. [20] proposed a graph convolutional network (GCN), which constructs graph convolution networks by local first-order approximation of spectral convolution. The features of nodes are aggregated from local neighbors. Since then, a large number of researchers have set foot in this field. William et al. [21] proposed an inductive framework, GraphSAGE. As a node embedding algorithm, the framework effectively generates node embedding for invisible data using node sampling and feature aggregation technology, which solves the limitation that GCN needs to calculate the full graph Laplacian. The output embedding of each layer of nodes depends on all the neighbors of the previous layer' neighbors and shows how to extract nodes from the neighborhood of nodes aggregate information. Veličković et al. [22] proposed an attention-based architecture to classify graph structure data.

At present, a graph convolution neural network has been applied in many fields, such as in a recommendation system [23–26], in malicious account detection [27], etc. Ali et al. [28] proposed a spam bot detection model based on a graph convolutional neural network (GCNN) by using the features of nodes and aggregating the features of node neighborhoods. In addition to the feature set, they also considered the social graph, which can detect spam bots better.

2.3. Attention Mechanism

The attention mechanism was first proposed in the visual field, and the Bengio team proposed it in 2014, and it has been widely used in various fields of deep learning in recent years. Then, a new type of network structure consisting only of self-attention and feedforward neural network appeared. When the attention mechanism is used to calculate the representation of a single sequence, it is usually called self-attention or internal attention [29]. The Google machine translation team completely abandoned network structures such as RNN and CNN, and only used the attention mechanism for machine translation tasks, and achieved good results [29].

In essence, the attention mechanism can allocate more computing resources to the more valuable parts of the representation, that is, certain attributes or instances have higher weights. One of the benefits of attention mechanisms is that they allow the processing of variable-sized inputs, focusing on the most relevant part of the input to make decisions [22]. The attention mechanism has also become a recent research hotspot. Inspired by the attention mechanism, the graph attention network (GAT) [22] introduced the attention mechanism into GCN. By calculating the attention coefficient between nodes, the importance between nodes and their neighborhood is learned, and the neighborhood is fused to classify nodes. GAT allows each node to make decisions on its most relevant neighbors. In this paper, a new social spam bot detection model is proposed, and a novel GAT-based method is applied to the problem of spam bot detection for the first time.

3. The Proposed Approach

In this section, we first illustrate the problem of spam bot detection in social networks, then describe the preliminary contents of two subgraphs in social networks. Finally, we present the approach to modeling spam bot detection.

3.1. Formulation of The Problem

Spam bot detection is essentially a binary classification problem. Our goal is to build a classifier that can accurately assign labels to accounts in the test set based on the features of a group of trained users and/or social networks.

It can be described as follows. Given the social graph $G = (V, E)$, the purpose of the spam bot detection problem is to learn a classification function $f : N \to Y$. In the case of a given training set, the social network account nodes in set N are classified into the correct class with the credibility label Y by using the multielement information in account propagation. All kinds of information in account propagation, including both account information and network structure information, should be effectively integrated.

3.2. Subgraph Construction

In this subsection, we introduce the model of the user-follow graph and the user–retweet graph in social networks.

3.2.1. User-Follow Subgraph

We define the user-follow graph as a directed graph $G_f = (V, E_f)$, where each node $u_i \in V$ represents a user in social networks, and each edge $(u_i, u_j) \in E_f$ represents a follow relationship in the network. We use E_1 to represent a set of unidirectional edges, e.g., $E_1 = \left\{(u_i, u_j) \middle| (u_i, u_j) \in E_f and (u_j, u_i) \notin E_f\right\}$,

and E_2 is a bidirectional edge set, i.e., $E_2 = \left\{(u_i, u_j) \middle| (u_i, u_j) \in E_f and (u_j, u_i) \in E_f\right\}$. For each user $u_i \in V$, Figure 1 shows an example of a follow relationship in a simple social graph.

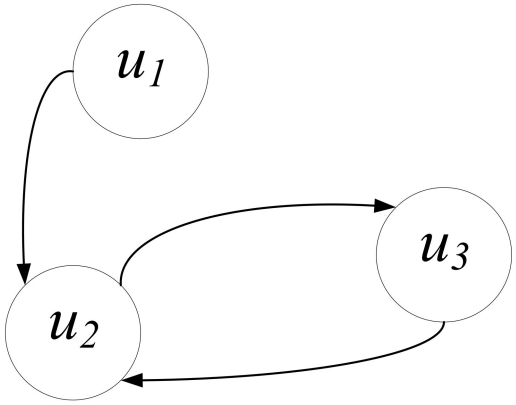

Figure 1. The follow relationship between users in social networks.

As shown in Figure 1, unidirectional edges (u_1, u_2) represent that user u_1 is following user u_2, and bidirectional edges (u_2, u_3) represent that user u_2 and user u_3 are following each other. Based on Twitter definitions, user u_1 is a follower of user u_2, user u_2 is a friend of user u_1, and user u_2 and user u_3 are mutual friends. There is no direct relationship between user u_1 and user u_3. In this paper, we denote the set of all neighbor relations of user u_i as $E_f = E_1 \cup E_2$.

There are several explicit natural patterns in user following relationships that we can use to further enhance our model [19]. Different types of neighbors have different effects on the label of an unknown user. In social networks, according to the homogeneity of the social network, assuming that account u_j and account u_i are bidirectional neighbors, u_i usually has the same label as u_j, because legitimate accounts are not linked to anomaly accounts with the bidirectional edge in most cases. In the unidirectional relationship, if the user u_i has many unidirectional incoming neighbors, the user u_i tends to be marked as a legitimate account because the anomaly accounts in OSNs are unlikely to attract a large number of other accounts, especially legitimate accounts. If the user u_i has many unidirectional outgoing neighbors, then u_i is likely to be an anomaly account because the anomaly accounts need to pay attention to a large number of other accounts to realize the dissemination of spam information, while the legitimate account is selectively following other account.

3.2.2. User–Retweet Subgraph

We define the user–retweet subgraph as a directed graph $G_r = (V, E_r)$, wherein each node $u_i \in V$ represents a user in social networks, and each edge $(u_i, u_j) \in E_r$ represents a retweet relationship in the network. That is, user u_i has retweeted user u_j. The user–retweet subgraph is widely used in social network analysis. Previous studies have shown that retweets can judge the influence of users better than followers [30]. Influential users have strong relevance via their retweet influence. The probability of anomaly users retweeting other anomaly users is 71 times higher than retweeting legitimate users. The relationship between anomaly users is very dense, and the anomaly users with medium influence have higher network centers [31]. This analysis is also applicable to the first-order neighborhood of anomaly users.

Figure 2 shows the retweet relationship between users. The unidirectional edge (u_2, u_1) represents that user u_2 retweets tweets from user u_1. Note that the direction of influence flow is opposite to that of retweets, so we are actually dealing with graphs with reverse edges [32]. As shown in Figure 2, if users u_2, u_4 and u_5 all retweet u_1, but user u_1 retweets nobody, u_1 may still be a central and influential node. If user u_1 is an anomaly account, users u_2, u_4 and u_5 also tend to be marked as anomaly accounts according to the influence of the retweet relationship.

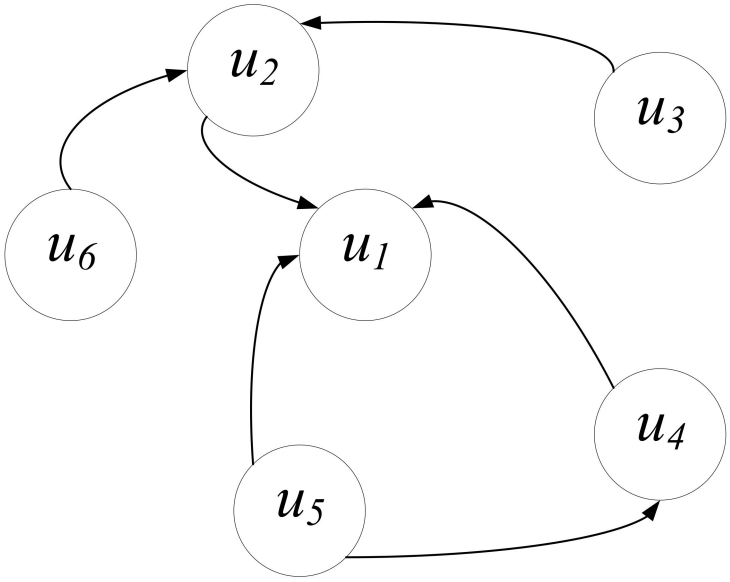

Figure 2. The retweet relationship between users in social networks.

3.3. GAT-Based Spam Bot Detection Model

In this subsection, we describe the proposed method for spam bot detection in social networks. First, since our goal is to identify spam bots in social networks, it can be formulated as a node classification problem on a directed bipartite graph with attribute nodes and edges. Therefore, we improve the GAT [22] model and propose a spam bot detection model that aggregates different neighborhood relations. The proposed method considers different types of neighborhood information. The framework of the whole model can be shown in Figure 3. By explicitly considering the neighbor relationship in the two relational subgraphs described in Section 3.2, the attention mechanism is used to aggregate neighbor nodes' features onto the central node, and the importance of different nodes is learned. Finally, the whole model is trained end-to-end.

Next, we introduce how to model and describe the algorithm of the proposed method in detail. To model the graph structure, we obtain the user embedding by collecting the embedding of neighbors. However, before aggregating each node's neighbor information, we should know that in the social network structure, any two user nodes can be connected through different relationships (such as followers). In a given relationship, each user has some neighbors based on this relationship. Since each neighbor has different degrees of influence on the user, the purpose of node-level attention is to learn the importance of neighbors to this node in the social graph, and assign different attention values to all neighbors. We utilize node-level attention to express the importance of neighbor nodes relative to a node in the social graph, and finally form node embedding. Specifically, we use self-attention [29] on the nodes to learn the weights of various nodes.

First, consider the user–follow subgraph. For a node $i \in V$ in the user–follow subgraph, the feature vector corresponding to the l level is $h_i^f = \left\{h_1^f, h_2^f, \ldots, h_N^f\right\}$, $h_i^f \in R^{d(l)}$, where N is the number of nodes and $d(l)$ is the number of features in each node. After using the attention mechanism to implement the aggregation operation, the new feature vector is represented as $h_i^{f'} = \left\{h_1^{f'}, h_2^{f'}, \ldots, h_N^{f'}\right\}$, $h_i^{f'} \in R^{d(l)}$.

In order to get a good enough representation ability, the input features need to be transformed into higher-level features. The input feature needs at least one linear transformation to get the output feature, so we need to train a weight matrix for all nodes: $W^f \in R^{d(l+1)*d(l)}$, which is the relationship between the $d(l)$ features of input and the $d(l+1)$ features of output.

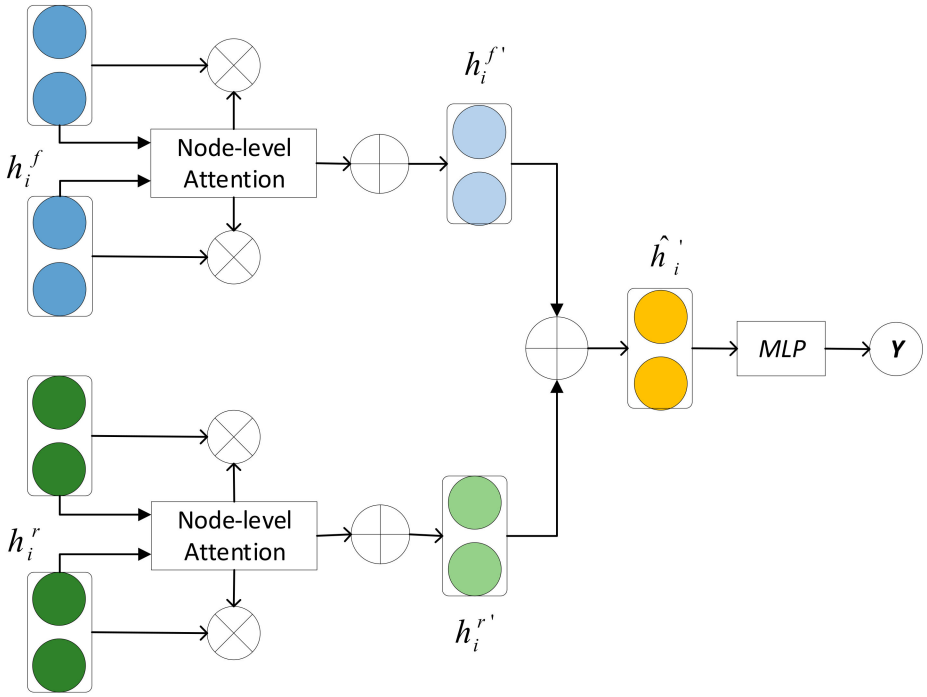

Figure 3. The framework of the proposed model.

Given a pair of nodes, $i \in V$ and $j \in V$, which are connected by a relationship (i, j), we can learn the weight coefficient of e_{ij}^{f} that indicates the importance of node j to node i. The importance of nodes to (i, j) can be formulated as:

$$e_{ij}^{f} = att(W^{f} h_{i}^{f}, W^{f} h_{j}^{f}) \tag{1}$$

where $W^{f} h_{i}^{f}$ means the line transformation for the feature h_{i}^{f} of node i in the user–follow subgraph, and att is a self-attention mechanism for each node, in which a single layer feedforward neural network is used. We use LeakyReLU as the activation function:

$$\text{LeakyReLU}(x) = \begin{cases} x, & if\ x \geq 0 \\ negative\_slope * x, & otherwise \end{cases} \tag{2}$$

where x is the input, negative slope is a negative gradient, and the default value is 1×10^{-2}. The nonlinear activation of LeakyReLU is added:

$$e_{ij}^{f} = \text{LeakyReLU}(a^{T}[W^{f} h_{i}^{f} \| W^{f} h_{j}^{f}]) \tag{3}$$

where $\|$ represents the splicing operation, and a^{T} is the weight vector after transposition, which is used to parameterize the attention mechanism att.

Equation (3) indicates the importance of node j to node i without considering the information of graph structure. Therefore, this attention mechanism is introduced into the graph structure through masked attention. Thus, we only need to calculate the e_{ij}^{f} of node $j \in N_{i}^{f}$, where N_{i}^{f} is the neighbor of node i (including itself) in the user–follow subgraph. In order to allocate the weight better, we need to

normalize the correlation between the current central node and all its neighbors, and get the weight coefficient α_{ij}^{f} by the SoftMax function:

$$\alpha_{ij}^{f} = \text{softmax}_j(e_{ij}^{f}) = \frac{\exp(e_{ij}^{f})}{\sum_{k \in N_i^f} \exp(e_{ik}^{f})} \tag{4}$$

By processing the above formula, the sum of the weight coefficients of all the neighbors of the current central node is 1.

After full expansion, the coefficients calculated by the attention mechanism can be expressed as follows:

$$\alpha_{ij}^{f} = \frac{\exp(\text{LeakyReLU}(a^T[W^f h_i^f \| W^f h_j^f]))}{\sum_{k \in N_i^f} \exp(\text{LeakyReLU}(a^T[W^f h_i^f \| W^f h_k^f]))} \tag{5}$$

It can be seen that the weight coefficients of (i, j) depend on their features. In addition, the weight coefficient calculated by Equation (5) is asymmetric. Therefore, their contributions are different from each other. Note also that the weight coefficients α_{ij}^{f} are asymmetric, which means that they contribute differently to each other. The normalized attention coefficient is used to calculate the linear combination of corresponding features as the final output feature of each node:

$$h_i^{f\prime} = \sigma(\sum_{j \in N_i^f} \alpha_{ij}^{f} W^f h_j^f) \tag{6}$$

where σ denotes ELU nonlinear activation.

The user–retweet subgraph is treated in the same way. However, when we deal with the connection between the user–follow and user–retweet subgraphs simultaneously, we usually do not know the importance of transformation information, which comes from different subgraphs. We cannot simply average the information together, but adaptively estimate the learning process of different subgraphs. Then, the embedding of the node i can be aggregated by the projection features of neighbors with corresponding coefficients, as follows:

$$h_i' = \sigma(\frac{1}{2}(\beta \sum_{j \in N_i^f} \alpha_{ij}^{f} W^f h_j + (1-\beta) \sum_{j \in N_i^r} \alpha_{ij}^{r} W^r h_j)) \tag{7}$$

where β is a free parameter to be estimated, the superscripts f and r represent the user–follow subgraph and user–retweet subgraphs, respectively.

In order to stabilize the learning process of self-attention, we use the multi-head attention mechanism which is used in the GAT [22] paper. We use K independent attention heads to perform the transformation of Equation (7), then use the average value and delay the application of the final nonlinearity to obtain the following output feature representation:

$$\hat{h}_i' = \sigma(\frac{1}{K}\sum_{k=1}^{K}(\frac{1}{2}(\beta \sum_{j \in N_i^f} \alpha_{ij}^{f} W^f h_j + (1-\beta) \sum_{j \in N_i^r} \alpha_{ij}^{r} W^r h_j))) \tag{8}$$

where the K is the total number of times the attention mechanism is executed.

In our experiment, to optimize the task of node classification, we integrate the representation of nodes into the full connection layer, including SoftMax for prediction, to infer the classification label. For the semi-supervised node classification task, we minimize the cross-entropy (CE) loss:

$$Loss = -\sum_{i \in V} y_i \cdot \log(\hat{y}'_i) + (1 - y_i) \cdot \log(1 - \hat{y}'_i) \tag{9}$$

where y_i and $\hat{y}'_i$ are the ground truth and the predicted class label for node i, respectively.

For the spam bot detection, we minimize the cross-entropy loss as the optimization objective function, which can be optimized by back-propagation with the help of a labeled node, and learn node embedding simultaneously. The whole algorithm is described in Algorithm 1.

Algorithm 1. The main process of our approach

Input: The graphs G_f and G_r; The feature vectors $\left\{h_i^f, h_i^r, \forall i \in V\right\}$; The number of attention head K
Output: prediction labels vector Y
1: for $k = 1 \ldots K$ do
2: for $i \in V$ do
3: for $j \in N_i$ do
4: Compute the weight coefficient ${\alpha^f}_{ij}$ in user–follow subgraph using Equation (5)
5: Compute the weight coefficient ${\alpha^r}_{ij}$ in retweet–follow subgraph using Equation (5)
6: end for
7: compute the node embedding h'_i using Equation (7) with coefficient β
8: Using Equation (8) to average the learned embeddings $\hat{h}'_i$ from all attention head
9: end for
10: compute cross-entropy loss using Equation (9) and do back-propagation
11: Update model parameters and prediction labels vector Y
12: return Y

4. Experiments

4.1. Dataset

The Twitter 1KS-10KN dataset [10] contains labeled spam bots and legitimate users collected on Twitter, as well as the followers and followings of each account. In addition, it also contains their corresponding tweets and social network information. The dataset contains 11,000 nodes and 2,342,816 edges. The authors have shared their datasets on the internet for scientific research, which we use in this article to evaluate our approach.

We use lightweight features that do not require complex calculations to form a feature set. The features are shown in Table 1:

Table 1. Feature set.

Feature	Description
account_age	The number of days an account was created on Twitter
no_followers	The number of followers for an account on Twitter
no_followings	The number of followings for an account on Twitter
no_userfavourites	The number of favorites received by an account on Twitter
no_statuses	The number of tweets posted by the account on Twitter, including retweets
no_tweets	The number of tweets posted by the account on Twitter
no_retweets	The number of tweets retweeted

4.2. Evaluation Metrics

It is very important to choose a meaningful evaluation criterion for a binary classification task. When measuring the performance of the algorithm, the most commonly used indicators are accuracy,

precision, recall and F1-score, but the accuracy rate cannot reflect the overall situation in the imbalanced dataset [33]. In order to more truly reflect the overall classification effect, we use the recall, precision and F1-score to measure the performance of the proposed spam bot detection method. The performance indicators are calculated as follows:

(1) Recall

The recall rate is the ratio of the correct classification of positive samples. It focuses on evaluating the quantity of all the positive data that are successfully predicted as positive.

(2) Precision

Precision indicates the ratio of positive samples correctly predicted in the total predicted positive samples. It focuses on assessing the quantity of all the data predicted as positive, which is the real positive data.

(3) F1-score

The F1-score is a harmonic average of recall and precision. It is used as an evaluation standard to measure the classifier's comprehensive performance, and this metric is used as an important evaluation metric to measure overall performance in our approach.

4.3. Compared Methods

(1) MLP: MLP is a feedforward artificial neural network model, which maps a set of input vectors to a set of output vectors, and trains based on the feature set defined in the feature part.
(2) BP [34]: The belief propagation (BP) algorithm is an approximate calculation based on MRF, in which information is transmitted iteratively between nodes in the graph, and the labels of nodes are inferred from the prior knowledge of nodes and other adjacent nodes.
(3) RF [35]: This is a random forest classifier with multiple decision trees, and the output class is determined by the mode of the categories output by individual trees.
(4) GCN [20]: This is a semi-supervised graph convolution network designed for graph structure. The edge information is used to aggregate the nodes to generate a new node representation.
(5) GraphSAGE [21]: GraphSAGE extends GCN into an inductive learning task by training the function (convolution layer) of neighbors of aggregation nodes, which play a generalization role for unknown nodes.
(6) GAT [22]: This is a semi-supervised neural network with a graph attention mechanism. The attention mechanism is used to aggregate the neighbor nodes to realize the adaptive allocation of different neighbor weights.

4.4. Parameter Settings

Our model uses two hidden layers with 16 hidden units, and the number of attention heads is eight. We optimize the parameters of the verification set and report the adjusted settings. We use the Adam optimizer to train all the models. The learning rate is 0.005, with a maximum of 200 epochs. The window of 10 epochs is used to stop ahead of time. We also use a dropout rate of 0.5.

For conventional methods, the MLP and BP methods are compared using the experimental results in the paper of Ali [28]. For the RF method, we use the default parameter settings from the original literature. For the method based on the graph neural network, on the basis of most of the parameter settings recommended in the original paper, we optimize some parameters to fit the dataset we use. We set two hidden layers for them. For GCN and GraphSAGE, we use 128 and 32 neurons, respectively. For GAT, the number of hidden neurons is set to 16, and the number of attention heads is eight.

4.5. Result Analysis

4.5.1. Comparison with Baselines

In this section, we first compare our algorithm with three classical machine learning algorithms. In order to ensure the objectivity of the results, all the conventional algorithms use their default parameters. The classification results are given in Table 2 in terms of recall, precision, and F1-score on the test data.

Table 2. Classification results of classical machine learning methods.

Method	Recall	Precision	F1-Score
RF	0.66	0.88	0.76
MLP	0.73	0.81	0.77
BP	0.54	0.56	0.55
Our Approach	0.88	0.93	0.91

Our method considers both feature set and graph structure. We can see that our method is superior to other algorithms by comparing with classical machine learning algorithms such as RF, MLP and BP, which only consider the feature set or graph structure. Especially for recall metric, our method gets a high score of 0.88, which is 15% higher than that of the MLP algorithm. The comprehensive evaluation metric of the F1-score represents the comprehensive performance of the classifier. The F1 score of our method is 15% higher than the second-ranked RF algorithm. Although the RF algorithm's precision reaches 0.88, which is close to our method, its recall value is only 0.66. This result reflects that the RF algorithm is affected by imbalanced data, and the classifier makes a judgment result that tends towards the majority class. Therefore, although the precision is high, about 34% of spam bots are classified as legitimate accounts, which will still cause unpredictable security implications for the social network, and cannot be applied to real spam bot detection. The BP algorithm classifies users from the graph structure. Due to an imbalanced dataset's influence, most users tend to be recognized as legitimate users, so the classification effect is poor.

Compared with conventional machine learning algorithms on the same dataset, the recall and F1 scores of our method are much higher than those of other methods. It can be seen that the classification effect of the classifier based on multi-feature representation is much better than that of the classifier based only on features in both single and comprehensive evaluation.

4.5.2. Comparison with State-of-the-Art Methods

The above baseline detection methods are all supervised models and can only use the social network's labeled part. Therefore, they have poor performance in the semi-supervised environment of the real world [19]. Accordingly, we choose GCN, GraphSAGE, and GAT, which belong to the graph neural network, to compare with our method.

Because the dataset used in this paper pays more attention to the following relationship, the retweet subgraph extracted according to Section 3.2 is sparse. The effect is worse when used alone for spam bot classification than for embedding with the following relationship. Therefore, the advanced graph neural network algorithm for experimental comparison only considers the following relationship. The classification results are given in Table 3 regarding recall, precision, and F1-score on the test data.

Table 3. Comparison of different algorithms on the dataset.

Method	Recall	Precision	F1-Score
GCN	0.76	0.87	0.81
GraphSAGE	0.80	0.88	0.84
GAT	0.83	0.87	0.85
Our Approach	0.88	0.93	0.91

In this part, due to the use of a more advanced graph neural network algorithm, the indicators have been significantly improved compared with the traditional machine learning algorithm, especially when the comprehensive evaluation metric of the F1 score, which is used to measure the algorithms' experimental performance, is relatively stable. Due to the introduction of GAT's self-attention mechanism, the correlation between node features is better integrated into the model. The node by node operation is free from the constraints of the Laplacian matrix. Therefore, the GAT algorithm shows a better performance in the experiment. Our method improves GAT and is superior to other algorithms in all metrics. The average increase of the three evaluation indicators is about 5%. In particular, the value of the F1 score is 6% higher than the second-ranked algorithm.

From the experimental results, the three graph-based neural network algorithms' classification performances are relatively close, especially in the precision metric. It can be seen that this kind of method is better than the traditional method in both accuracy and robustness. Because it considers more neighbor types, our method achieves better classification results.

Besides this, when evaluating the proposed GCNwithMRF method in the paper of Wu et al. [19], considering the class imbalance problem in the detection of spam bots in the social network, the author uses the Area Under the Precision-Recall Curve (PRAUC) calculated from the prediction score as the evaluation metric. This is because the accurate recall curve is better in evaluating class imbalanced data sets' performances, while the ROC curve is deceptive in this case [36]. Therefore, we use Scikit-learn [37] to calculate the PRAUC value. Our method's PRAUC value is 0.91, which is better than that of the GCNwithMRF method (0.87).

Through comparison, we can see that even in the imbalanced data, our method still maintains a relatively stable overall performance, and shows better robustness.

5. Conclusions

This paper studies the problem of spam bot detection and proposes a novel attention-based semi-supervised graph embedding model to solve it. The approach can identify spam bots by aggregating user features and neighborhood relationships between users. Firstly, based on the directed social graph's operation, the model integrates a feature-based method and a propagation-based method, modeling spam bot detection by aggregating features and neighbor relationship. Secondly, we also analyze and fuse the following relationship and the retweet relationship between users, which further improves the results. The fusion model identifies spam bots by capturing users' features and different neighborhood relationships in social networks. The improved graph attention network considers the features of users and considers the different interactions between users. Experiments on real Twitter data show that the proposed method is efficient and robust for spam bot detection.

In the future, we plan to mine more abundant hidden feature representations, and construct detection models based on the various structural data and rich semantic data from social networks, to further improve spam bot detection performance in social networks.

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

Funding: This research was funded by "National Key R&D Program of China under Grant 2017YFB0802300", "Major Scientific and Technological Special Project of Guizhou Province (20183001)", and "Foundation of Guizhou Provincial Key Laboratory of Public Big Data (2018BDKFJJ008, 2018BDKFJJ020, 2018BDKFJJ021)".

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

References

1. Adewole, K.S.; Anuar, N.B.; Kamsin, A.; Varathan, K.D.; Razak, S.A. Malicious accounts: Dark of the social networks. *J. Netw. Comput. Appl.* **2017**, *79*, 41–67. [CrossRef]
2. Wei, F.; Nguyen, U.T. Twitter Bot Detection Using Bidirectional Long Short-Term Memory Neural Networks and Word Embeddings. In Proceedings of the 2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), Los Angeles, CA, USA, 12–14 December 2019; pp. 101–109.
3. Yang, C.; Harkreader, R.; Gu, G. Empirical Evaluation and New Design for Fighting Evolving Twitter Spammers. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 1280–1293. [CrossRef]
4. Singh, M.; Bansal, D.; Sofat, S. Detecting Malicious Users in Twitter using Classifiers. In Proceedings of the 7th International Conference on Security of Information and Networks-SIN '14, Glasgow, UK, 9–11 September 2014; ACM Press: Glasgow, UK, 2014; pp. 247–253.
5. VanDam, C.; Tan, P.-N. Detecting hashtag hijacking from Twitter. In Proceedings of the 8th ACM Conference on Web Science, Hannover, Germany, 22–25 May 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 370–371.
6. Varol, O.; Ferrara, E.; Davis, C.A.; Menczer, F.; Flammini, A. Online Human-Bot Interactions: Detection, Estimation, and Characterization. *arXiv* **2017**, arXiv:170303107.
7. Chen, Z.; Subramanian, D. An Unsupervised Approach to Detect Spam Campaigns that Use Botnets on Twitter. *arXiv* **2018**, arXiv:180405232.
8. Cresci, S.; Di Pietro, R.; Petrocchi, M.; Spognardi, A.; Tesconi, M. The Paradigm-Shift of Social Spambots: Evidence, Theories, and Tools for the Arms Race. In Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, 3–7 April 2017; International World Wide Web Conferences Steering Committee: CHE, Geneva, 2017; pp. 963–972.
9. Kudugunta, S.; Ferrara, E. Deep neural networks for bot detection. *Inf. Sci.* **2018**, *467*, 312–322. [CrossRef]
10. Yang, C.; Harkreader, R.; Zhang, J.; Shin, S.; Gu, G. Analyzing spammers' social networks for fun and profit: A case study of cyber criminal ecosystem on twitter. In Proceedings of the 21st International Conference on World Wide Web, Lyon, France, 16–20 April 2012; Association for Computing Machinery: New York, NY, USA, 2012; pp. 71–80.
11. Cresci, S.; Di Pietro, R.; Petrocchi, M.; Spognardi, A.; Tesconi, M. DNA-Inspired Online Behavioral Modeling and Its Application to Spambot Detection. *IEEE Intell. Syst.* **2016**, *31*, 58–64. [CrossRef]
12. Loyola-González, O.; Monroy, R.; Rodríguez, J.; López-Cuevas, A.; Mata-Sánchez, J.I. Contrast Pattern-Based Classification for Bot Detection on Twitter. *IEEE Access* **2019**, *7*, 45800–45817. [CrossRef]
13. Davis, C.A.; Varol, O.; Ferrara, E.; Flammini, A.; Menczer, F. BotOrNot: A System to Evaluate Social Bots. In Proceedings of the 25th International Conference Companion on World Wide Web, Montreal, QC, Canada, 11–15 April 2016; International World Wide Web Conferences Steering Committee: Geneva, Switzerland, 2016; pp. 273–274.
14. Li, C.; Wang, S.; He, L.; Yu, P.S.; Liang, Y.; Li, Z. SSDMV: Semi-Supervised Deep Social Spammer Detection by Multi-view Data Fusion. In Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM), Singapore, 17–20 November 2018; pp. 247–256.
15. Jia, J.; Wang, B.; Gong, N.Z. *Random Walk Based Fake Account Detection in Online Social Networks*; IEEE: Denver, CO, USA, 2017; pp. 273–284.
16. Wang, B.; Zhang, L.; Gong, N.Z. SybilSCAR: Sybil detection in online social networks via local rule based propagation. In Proceedings of the IEEE INFOCOM 2017-IEEE Conference on Computer Communications, Atlanta, GA, USA, 1–4 May 2017; pp. 1–9.
17. El-Mawass, N.; Honeine, P.; Vercouter, L. Supervised Classification of Social Spammers using a Similarity-based Markov Random Field Approach. In Proceedings of the 5th Multidisciplinary International Social Networks Conference, Saint-Etienne, France, 16–18 July 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 1–8.
18. Mulamba, D.; Ray, I.; Ray, I. On Sybil Classification in Online Social Networks Using Only Structural Features. In Proceedings of the 2018 16th Annual Conference on Privacy, Security and Trust (PST), Belfast, UK, 28–30 August 2018; pp. 1–10.

19. Wu, Y.; Lian, D.; Xu, Y.; Wu, L.; Chen, E. Graph Convolutional Networks with Markov Random Field Reasoning for Social Spammer Detection. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 1054–1061. [CrossRef]
20. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2017**, arXiv:160902907.
21. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems 30*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2017; pp. 1024–1034.
22. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. *arXiv* **2018**, arXiv:171010903.
23. Zhao, H.; Yao, Q.; Li, J.; Song, Y.; Lee, D.L. Meta-Graph Based Recommendation Fusion over Heterogeneous Information Networks. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; Association for Computing Machinery: New York, NY, USA, 2017; pp. 635–644.
24. Wang, J.; Huang, P.; Zhao, H.; Zhang, Z.; Zhao, B.; Lee, D.L. Billion-scale Commodity Embedding for E-commerce Recommendation in Alibaba. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 839–848.
25. Ying, R.; He, R.; Chen, K.; Eksombatchai, P.; Hamilton, W.L.; Leskovec, J. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 974–983.
26. Grbovic, M.; Cheng, H. Real-time Personalization using Embeddings for Search Ranking at Airbnb. In Proceedings of the Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 311–320.
27. Liu, Z.; Chen, C.; Yang, X.; Zhou, J.; Li, X.; Song, L. Heterogeneous Graph Neural Networks for Malicious Account Detection. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Turin, Italy, 22–26 October 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 2077–2085.
28. Ali Alhosseini, S.; Bin Tareaf, R.; Najafi, P.; Meinel, C. Detect Me If You Can: Spam Bot Detection Using Inductive Representation Learning. In Proceedings of the 2019 World Wide Web Conference on-WWW '19, San Francisco, CA, USA, 19–21 May 2019; ACM Press: San Francisco, CA, USA, 2019; pp. 148–153.
29. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2017; pp. 5998–6008.
30. Cha, M.; Haddadi, H.; Benevenuto, F.; Gummadi, K.P. Measuring User Influence in Twitter: The Million Follower Fallacy. *Icwsm* **2010**, *10*, 30.
31. Ribeiro, M.H.; Calais, P.H.; Santos, Y.A.; Almeida, V.A.F.; Meira, W., Jr. Characterizing and Detecting Hateful Users on Twitter. *arXiv* **2018**, arXiv:180308977.
32. Ribeiro, M.H.; Calais, P.H.; Santos, Y.A.; Almeida, V.A.F.; Meira, W., Jr. "Like Sheep among Wolves": Characterizing Hateful Users on Twitter. *arXiv* **2018**, arXiv:180100317.
33. Zhao, C.; Xin, Y.; Li, X.; Yang, Y.; Chen, Y. A Heterogeneous Ensemble Learning Framework for Spam Detection in Social Networks with Imbalanced Data. *Appl. Sci.* **2020**, *10*, 936. [CrossRef]
34. Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*; Elsevier: Amsterdam, The Netherlands, 2014; ISBN 978-0-08-051489-5.
35. Fu, H.; Xie, X.; Rui, Y.; Gong, N.Z.; Sun, G.; Chen, E. Robust Spammer Detection in Microblogs: Leveraging User Carefulness. *ACM Trans. Intell. Syst. Technol.* **2017**, *8*, 83:1–83:31. [CrossRef]

36. Davis, J.; Goadrich, M. The relationship between Precision-Recall and ROC curves. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; Association for Computing Machinery: New York, NY, USA, 2006; pp. 233–240.
37. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.