Hindawi
Security and Communication Networks
Volume 2022, Article ID 9119388, 14 pages
https://doi.org/10.1155/2022/9119388

Research Article

A New Joint Approach with Temporal and Profile Information for Social Bot Detection

Zhou Yang,[1] **Xingshu Chen**,[1,2] **Haizhou Wang**,[1] **Wenxian Wang**,[2] **Zhenxiong Miao**,[1] **and Tao Jiang**[3]

[1]*School of Cyber Science and Engineering, Sichuan University, Chengdu 610207, China*
[2]*Cyber Science Research Institute, Sichuan University, Chengdu 610207, China*
[3]*China Electronics Technology Cyber Security Co.,Ltd, Chengdu 610041, China*

Correspondence should be addressed to Haizhou Wang; whzh.nc@scu.edu.cn

Received 24 January 2022; Accepted 6 April 2022; Published 7 May 2022

Academic Editor: Zhe-Li Liu

With the increasing popularity of online social networks (OSNs), a huge number of social bots have emerged. Social bots are involved in various cybercrimes like cyberbullying and rumor dissemination, which have seriously affected the normal order of OSNs. Nowadays, existing studies in this field almost focus on English OSNs like Twitter and Facebook. However, it is difficult to directly apply these detection technologies to Sina Weibo, which is one of the largest Chinese microblogging services in the world. In addition, social bots are evolving rapidly and time-consuming feature engineering may not perform well in detecting newly emerging social bots. In this paper, we propose a new joint approach with Temporal and Profile information for social bot detection (TPBot). The approach includes data collection module, feature extraction module, and detection module. To begin with, data collection module uses a web crawler to obtain user data from Sina Weibo. Next, the feature extraction module regards the user posts as temporal data to extract temporal-semantic and temporal-metadata features. Furthermore, this module extracts features based on users' profile. Finally, a detection model based on BiGRU and attention mechanism is designed in the detection module. The results show that TPBot performs better than baselines with the F1-score of 0.9837 on the Sina Weibo dataset. Moreover, we have also conducted an experiment on the two datasets collected from Twitter to evaluate the generalization ability of TPBot. It is found that TPBot outperforms baselines on the new datasets and has good generalization ability.

1. Introduction

In recent years, the rapid development of Internet technology makes it more convenient for people to share information on online social networks (OSNs), which have gradually become an important part of public life [1, 2]. At the same time, with the increasing number of active users in OSNs, there are many OSN accounts that are abused by certain individuals or organizations. These accounts are automatically created and controlled by programs, namely, social bots [3]. Social bots can imitate the behaviors of human accounts and be active in OSNs for a long time [4]. At first, social bots were used to serve users and their behaviors included chatting with users [5], automatically posting news (https://bbcnewslabs.co.uk/projects/bots/), and so on. However, there are a growing number of malicious social bots that attempt to control public opinion and even distort reality [6–9]. For instance, some candidates used social bots to publicize their policy and interfere with public opinion during political elections [6, 7]. During the Covid-19 pandemic, malicious social bots spread various tendentious speeches to mislead public opinion [8, 9]. Nowadays, Sina Weibo has become one of the most popular Chinese OSNs in the world [10]. A large number of malicious social bots, which spread malwares, spam, and harmful links [11, 12] in Sina Weibo, have caused harm to the normal order of the platform. Therefore, it is of great importance to detect social bots in Sina Weibo.

Most of the existing work [13–19] is undertaken in English OSNs, and only a few studies [20–23] focus on Chinese OSNs, such as Sina Weibo. Considering the differences in main languages, interaction with users, information sharing, and features of social bots in different OSNs, it is difficult to directly apply the existing detection technologies based on other OSNs to Sina Weibo. Moreover, social bots are constantly evolving and developing [24], while features extracted through time-consuming feature engineering may be effective in detecting only a specific category of social bots [25]. Hence, it is difficult to perform quite well in social bot detection only by feature engineering. In summary, there is still a lot of work to do to improve the performance of the approaches applied to social bot detection in Sina Weibo.

1.1. Challenges. Early research on social bot detection mainly adopted graph-based approaches [26–31], which analyze the variability between the social graph formed by bots and normal users. Graph-based approaches are useful to identify social bots and measure the influence of users. Nevertheless, such approaches have high execution time cost and are good for relatively small datasets [4].

With the wide use of machine learning algorithms, machine learning–based approaches [15, 16, 19, 24, 32–36] have become the most popular approach in the field of social bot detection. The machine learning approaches extract a set of effective features from user data such as profiles, posts, and relationships. Compared with graph-based approaches, machine learning–based approaches are easier to implement and use. However, machine learning–based approaches also suffer from time-consuming feature engineering and need to be undertaken more in-depth [25]. In summary, research on social bot detection mainly faces the following three challenges:

(i) It is a time-consuming and laborious task to construct detection features manually. In order to more accurately detect social bots, many existing studies [37, 38] have constructed a large number of features from multiple aspects, which makes feature engineering time-consuming and laborious.

(ii) Existing studies ignored the latent relationship between posts. Users in OSNs publish different posts at different times, so there are latent temporal patterns between posts published by the same user. However, some studies [23, 38] are limited to computing the average and variance of indicators (the number of URLs, the number of hashtags, the number of mentions, etc.) as the similarity of posts, which cannot fully reflect the latent relationship between posts.

(iii) The generalization ability of detection approaches still needs to be improved. Due to the differences between several OSNs, most of the current detection approaches are only applied to detect social bots in a single OSN, which means it is difficult for them to achieve a good performance in other OSNs. Although a few studies [27] have proposed some approaches that can detect social bots across OSNs, the performance of these approaches is not good enough. Therefore, a social bot detection approach with good generalization ability needs further development.

1.2. Contribution. To address the above limitations, this paper proposes TPBot, a deep learning–based approach, which uses Temporal and Profile information for social bot detection. TPBot consists of three modules: data collection module, feature extraction module, and detection module. Firstly, the data collection module is responsible for using a web crawler to collect user data, including user profile and posts. Secondly, three categories of features are extracted in the feature extraction module based on user behavior, content, and profile dimensions. Finally, the detection module designs a classifier model based on BiGRU and attention mechanism, which aims to capture the latent factors in the features from the upstream module and obtain the classification label (social bots or normal users). The main contributions of our work are summarized as follows:

(i) The temporal patterns of user postings are extracted automatically by BiGRU, which captures the latent features of social behavior and content information. Different from existing studies that regarded posts of users as plain text, this paper concentrates on the order of posts and treats them as a time series. We establish BiGRU to automatically extract the latent temporal patterns between user posts. It can not only eliminate the difference between various similarity algorithms but also simplify the process of feature extraction and realize the depth mining of hidden features.

(ii) A new joint approach for social bot detection is proposed, which makes use of temporal and profile information. The approach can generate user representations to identify social bots by fusing temporal-semantic information, temporal-metadata information, and profile information in Sina Weibo. The experimental results show that TPBot has the best performance compared to the state-of-the-art baselines.

(iii) A deep learning–based model with good generalization ability is evaluated on different real-world datasets. At present, most of the current detection approaches are only adaptable for a single OSN, and it is difficult to directly apply them to other OSNs. We test the detection model on the datasets from Sina Weibo and Twitter. As a result, the model outperforms several baseline approaches, proving its good generalization ability.

2. Related Work

The studies related to social bots detection have gradually emerged in the last decade, and can be divided into three categories: graph-based approaches [26–31], machine learning–based approaches [15, 16, 19, 24, 32–36], and other approaches [8, 39–42].

2.1. Graph-Based Approaches. Since graph structures are often used to represent social network structures, graph-based approaches have also been applied to the social bot detection. Based on the assumption that normal users do not actively follow social bots, Feng et al. [28] first built an undirected graph using the bidirectional following relationship of the target user. Then, they constructed matrixes of the target user and its associated users by calculating Jaccard coefficients. Finally, they compute the similarity of these matrixes and took it as the probability that the user is a social bot. The algorithm can accomplish a high detection rate of 86.27% at a low false positive rate of 8.54%. Similarly, Boshmaf et al. [29] designed a system called Íntegro to detect social bots, which assigns weights to the adjacent edges of victims users (i.e., users befriend fake accounts) in the social graph. This system employs an improved random walk that starts from a real user to rank users. Íntegro ultimately makes the ranks of most real users higher than fake accounts, thus achieving the goal of effective classification. The experimental result showed that the quality of user ranking of this system was significantly better than that of other systems.

Graph-based approaches can also be combined with machine learning algorithms. The typical approach is to model the user information based on the graph approaches, and then use the machine learning algorithms to detect social bots according to the modeled information. For example, in Ref. [30], Ahmad and Abulaish first extracted a set of features and modeled the social network with weighted graphs. They leveraged unsupervised machine learning approaches to cluster these weighted graphs and achieved a good performance on three different datasets. To improve the performance of the detection approaches, Guo et al. [27] expected to deeply mine unknown linkages from the view of heterogeneous information networks. They first inferred representative vectors of occasional relations. Then, a graph neural network framework was developed for spammer detection. The results showed that their approach performed about 5 %–10 % better than baselines. In Ref. [31], a system called BotCamp was proposed, which created graph structures based on the social behavior of users. BotCamp clustered the collected bots based on these graphs and other collected information to detect social bot clusters.

Graph-based approaches consider that social bots have significantly fewer links to other users than normal users, so the social graph properties of users can be used for detection. However, social bots can evade detection by constructing enough links with each other. On the other hand, it is difficult to obtain all the relationships of users due to the OSN restrictions [43], which have been a major factor limiting the further development of the approaches.

2.2. Machine Learning Approaches. Machine learning–based approaches are the most popular approaches in the field of social bot detection. Machine learning–based approaches regard the problem as a binary classification and use machine learning classifiers to identify social bots. Machine learning–based approaches can be divided into classical machine learning approaches [15, 16, 24, 32, 33] and deep learning approaches [19, 34–36], which are described below.

2.2.1. Classical Machine Learning Approaches. Classical machine learning approaches construct a set of features to model user information and classify users using classical machine learning algorithms. Botometer [15] is an off-the-shelf system, which is the first publicly available interface for Twitter bots detection. Botometer first divides user features into several categories, including user profile-based features, network-based features, and tweet-based features. Then, a total of more than 1,000 features are constructed to measure the "botness" of a Twitter account. Although this system extracts a wide range of features, each feature is simple and needs to be undertaken more in-depth. Similarly, an integrated social media content analysis platform was proposed by Al-Qurishi et al. in Ref. [16], which leveraged three levels of features, i.e., user-generated content, social graph connections, and user profile activities to detect social bots. The authors also proposed a novel approach regarding the process of data extraction and classification to contextualize large-scale networks. Finally, classical machine learning classifiers such as support vector machine (SVM), random forest (RF), etc., were applied and RF reached the highest accuracy with 0.9607.

In addition, some studies have introduced other new features for social bot detection. Ji et al. [24] comprehensively analyzed the evasion mechanisms used by existing social bots and validated those mechanisms by applying three state-of-the-art detection approaches to their collected traces. Then, based on the insights gained, they proposed a new detection approach including nine newly identified features and two new correlation mechanisms. The experimental results indicated that their approach performed well under various classifiers and RF achieved the best performance with the F1-score reaching 0.9630. Dickerson et al. [32] considered that human and nonhuman users are different in tweet sentiment and analyzed the novel semantic features. In their work, a sentiment-aware architecture called SentiBot was proposed for identifying social bots in Twitter. The experimental results showed that a number of sentiment-related factors were key to the identification of social bots, which significantly increased the area under the ROC curve.

Wang et al. [33] found that most social bots have similarities in their tweets because of creators' purposes and current technology limitations. Based on these assumptions, they constructed semantic similarity features using several computing methods of content similarity. Some machine learning classifiers such as RF, decision tree (DT), etc., were applied to detect social bots. Finally, the results showed that the Latent Semantic Analysis (LSA) model made the classifier perform the best.

Classical machine learning approaches need to analyze and extract features of social bots from multiple aspects for more accurate detection. However, the features of social bots are complex and it is difficult to extract features completely by manual feature extraction. Therefore, deep learning approaches start to be more wildly used in social bot detection for mining more hidden features.

2.2.2. Deep Learning Approaches. With the development of deep learning techniques, more and more researchers have applied them in the field of social bots detection. Compared with classical machine learning approaches, deep learning approaches can replace manual feature extraction and mine more hidden features.

Kudugunta and Ferrara et al. [34] proposed a deep neural network based on contextual long short-term memory (LSTM) architecture that exploits both content and metadata to detect social bots at the tweet level. They also utilized a technique based on synthetic minority oversampling to enhance the dataset. The results demonstrated that their architecture can achieve a high classification accuracy with 0.9633 from just one single tweet. In a recent study [19], a deep learning model based on BiLSTM and attention mechanism was proposed, which further improved the accuracy of the tweet-level detection model with an average accuracy of 0.9975.

For account-level classification, Cai et al. [35] designed a CNN-LSTM network aiming to learn an effective representation of social user and then detect social bots by jointly modeling social behavior and content information. In Ref. [36], Ping and Qin et al. proposed a deep learning model, namely, DeBD. DeBD extracted the joint features and the temporal features and got an average F1-score of 0.9970. In our previous study [23], a framework based on deep neural networks and active learning called DABot was proposed to detect social bots in Sina Weibo. DABot first extracted 30 features from four categories, nine of which were completely new features. Then, a new deep neural network model called RGA was built to implement the detection of social bots. Moreover, DABot employed active learning to efficiently expand the labeled data. Finally, the results showed that DABot reached an accuracy with 0.9887 after expanding the dataset.

Deep learning approaches use deep neural networks to learn OSN user representation automatically. Nevertheless, the existing deep learning approaches do not employ all the aspects of information to generate user representations. For example, the authors only used content information in Refs. [35, 36], ignoring profile and behavior information. On the other hand, only tweets were used to detect social bots in Refs. [19, 34], which makes it easier for social bots to evade detection by manual creation of tweets. Moreover, our previous study [23] has the limitation of manual feature extraction, although it used four categories of features. As can be seen, deep learning approaches need further development.

2.3. Other Approaches. In addition to graph-based and machine learning approaches, there are also anomaly based approaches [39, 40], crowdsourcing-based approaches [8, 41], and proactive approaches [42]. Since these approaches are not common, they are uniformly categorized as other approaches in this paper.

Anomaly based approaches consider that legitimate OSN users would have no motivation to behave in an odd way. Therefore, users with unusual behaviors are likely to be malicious social bots. By randomly selecting six million English Twitter users, Echeveria and Zhou et al. [39] discovered a group of users with an odd trend in the tweet locations. Furthermore, the group showed some unusual properties such as randomly quoting the same novel. These odd behaviors convinced the authors to classify them as social bots. Chavoshi et al. [40] hypothesized that humans would not have many specific relevant activities over a long period of time. They first designed the DeBot model to track the relevant activities of the target users. After continuous attention over time, the authors inferred that users that posted at least 40 tweets in an hour were social bots. In addition, they labeled a group of users with such activity or highly similar activity as social bots.

The crowdsourcing-based approaches require many people to manually label and classify social bots. In Ref. [41], Alarifi et al. selected and trained 10 volunteers as crowdsourced workers to manually label Twitter accounts. Cresci et al. [8] conducted a test to measure the accuracy of the crowdsourcing in detecting social bots. They hired crowdsourced workers from a crowdsourcing website and asked them to classify pre-labeled datasets. The test results showed that the crowdsourcing-based approaches successfully detected traditional spam bots and genuine accounts but cannot accurately detect malicious spam bots.

Among these studies, the anomaly based approaches need to continuously collect a large number of user data for analysis, which is more effective when most of the social bots behave oddly. Conversely, the anomaly based approaches will fail when there are only a few social bots or the social bots have no odd behaviors. The crowdsourcing-based approaches involve high cost and time, so this technique is often used to collect labeled datasets.

3. Methodology

In this section, we describe in detail the proposed TPBot approach for social bot detection. TPBot is mainly composed of three modules: data collection module, feature extraction module, and detection module. The overall framework of the approach is shown in Figure 1.

Firstly, the data collection module uses a multi-threaded web crawler to collect data from Sina Weibo, including profiles and posts of users. Subsequently, the three submodules included in the feature extraction module are responsible for modeling content of posts, metadata of posts, and profile information, respectively. The semantic feature sequence, the metadata feature sequence, and the profile feature are the output vectors of this module. Finally, the detection module implements a detection model based on BiGRU and attention mechanism. It inputs the three different types of features extracted from the upstream module into the model for training or prediction to obtain the final classification results.

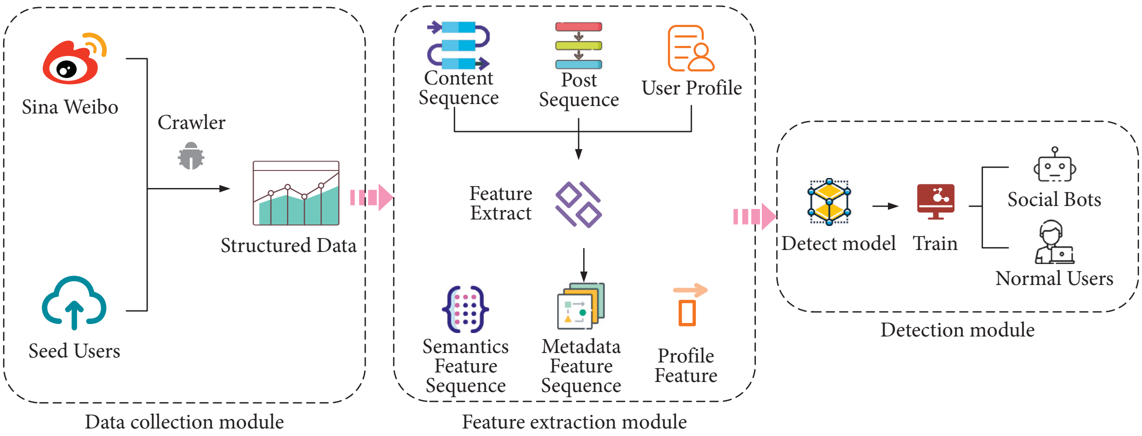


Figure 1: The architecture of the proposed social bot detection approach, TPBot.

3.1. Data Collection Module. Due to Sina Weibo adjusting its credit rules in December 2020, this module extends the original dataset SWLD-20K collected in Ref. [23], which was collected from October 2019 to January 2020. In order to collect more comprehensive data, after screening the users who are still alive in the original dataset, we develop a high-performance web crawler to crawl data of these users to construct the dataset SWLD-20K*. The SWLD-20K* provides a larger number of posts and more comprehensive attribute values. Table 1 shows the changes in the dataset size before and after, and the new dataset was collected from April 2021 to June 2021.

3.2. Feature Extraction Module. In order to effectively distinguish social bots from normal users, this module constructs the semantic feature sequence, the metadata feature sequence, and the profile feature as features of the user based on posts and profile information. The extracted features will be input into the detection model for further feature extraction and fusion to identify social bots.

3.2.1. Semantic Feature Sequence. For purposes such as product and political promotion, social bots often post a large amount of similar or even same textual content [38]. On the other hand, in order to ensure the spamming rate, there is usually a gap between the quality of content generated by social bots compared with normal users [33]. Consequently, both semantic features of posts and the relationship between them are important to distinguish social bots. Before inputting the content of posts into the detection model, they need to be sorted according to the publish time. Then, they will transform into the semantic features via the BERT pre-training model [44]. Given an OSN user u, we regard the timeline of posts published by u as a sequence of $T_u = [t_1, t_2, \ldots, t_d]$. Only one post can be published by the same user at a certain moment, thus the content sequence of u can be described as

$$C_u = \left[c_{t_1}, c_{t_2}, \ldots, c_{t_d}\right], \tag{1}$$

Table 1: Overview of the Weibo datasets.

Category	User number	Post number (SWLD-20K)	Post number (SWLD-20K*)
Normal users	10,000	118,199	1,779,569
Social bots	10,000	96,307	760,487
Total	20,000	214,506	2,540,056

where d denotes the maximum number of posts that is input into the model, c_{t_i} is the content of a post at t_i. After experiments, it is found that a better detection effect can be achieved when d is set to 100. If the number of posts of the user is less than d, the length of the content sequence is made d by filling in empty characters. If the number of posts of the user is greater than d, only the first posts of the user are processed. Then, the content sequence C_u is input into the BERT pre-training model. Each content in C_u is mapped to a vector $s_{t_i} \in \mathfrak{R}^{\omega \times \nu}$, where ω is the length of the content and is the dimension of BERT pretraining model output vector. Finally, the BERT model outputs a list of vectors, i.e., the user semantic feature sequence is

$$S_u = \left[s_{t_1}, s_{t_2}, \ldots, s_{t_d}\right]. \tag{2}$$

3.2.2. Metadata Feature Sequence. In addition to textual content, posts of OSN users also contain rich nontextual information, such as the number of likes, the number of comments, the publish time of posts, etc., which are collectively referred to as the metadata of posts in this paper. Compared with normal users, the metadata of posts of social bots has obvious differences. In our work, eight metadata-based features are constructed after referring to existing studies, as shown in Table 2.

(1) The number of likes, comments, and reposts: In Ref. [45], the authors examined the number of likes, comments, and reposts of posts as features for social bot detection. Likes, comments, and reposts represent the popularity of a user in OSN. In general, posts of social bots have few likes, comments, and reposts. Therefore, in this paper, the number of likes,

Table 2: Overview of the metadata-based features.

Symbol	Feature name	Source
λ_{nl}	The number of likes	[45]
λ_{nc}	The number of comments	[45]
λ_{nr}	The number of reposts	[45]
λ_{or}	The originality of post	[45]
λ_{pt}	The publish time of post	New
λ_{ne}	The number of emojis in posts	New
λ_{te}	The number type of emojis in posts	New
λ_{np}	The number of pictures in posts	[23]

comments, and reposts of a post is defined as λ_{nl}, λ_{nc}, and λ_{nr}.

(2) The originality of post: Many social bots generate their own posts by reposting or copying other users' posts, and thus the frequency of reposted posts of social bots is higher than that of normal users [46]. In this paper, we use λ_{or} to define the originality of user post, and take 1 when the post is reposted from other users, otherwise, take 0.

(3) The publish time of post: Because social bots achieve automatic posting with the help of computer programs, they can be active in OSN for a long time. In Ref. [47], the authors found that social bots posts are far more than the normal users between 2:00 AM and 8:00 AM in Sina Weibo. In our work, a post published late at night is denoted as λ_{pt} and is used as one of the features, which is calculated by

$$\lambda_{pt} = \begin{cases} 1, & \text{if } 2 \leq T \leq 8, \\ 0, & \text{else,} \end{cases} \tag{3}$$

where T is the number of hours of the publish time.

(4) The number and type of emojis in posts: While normal users often choose to insert emojis into the content to express specific emotions when posting in OSN, social bots rarely use emojis, so the emojis number λ_{ne} and emojis type λ_{te} in posts can also be used to identify social bots.

(5) The number of pictures in posts: Users often enrich their posts by posting pictures. In Ref. [23], the number of pictures in posts is one of the important features to detect social bots, which is defined as λ_{np}.

Like C_u, the post sequence is obtained by taking out posts of u according to the publish time, which can be denoted as

$$\mathrm{Post}_u = \left[\mathrm{post}_{t_1}, \mathrm{post}_{t_2}, \ldots, \mathrm{post}_{t_d}\right], \tag{4}$$

where d denotes the maximum number of posts that is input into the model, and post_{t_i} denotes the post at t_i. Next, the feature extraction is performed on the sequence of posts Post_u, and the metadata feature sequence is described as

$$M_u = \left[m_{t_1}, m_{t_2}, \ldots, m_{t_d}\right], \tag{5}$$

where m_{t_i} is the metadata feature vector corresponding to the post post_{t_i}. If the number of posts of the user is less than d, it is necessary to fill in zero vectors of the same dimension to make the length of the metadata feature sequence be d. If the number of posts of the user is greater than d, only the features of the user's first d posts are extracted as the metadata feature sequence.

3.2.3. Profile Feature. Profile feature P_u is a feature vector extracted based on user profile information. Social bots and normal users show obvious differences in terms of profile completeness and personalization. Combined with the characteristics of OSNs, this paper extracts four features as the user profile feature vector P_u, as shown in Table 3.

(1) Length of nickname: In Refs. [23, 48], the length of nickname is used to detect social bots, so the length of username is also adopted as a feature in this paper. We denote it as δ_{ln}, and the value range of δ_{ln} is $\{\delta_{\mathrm{ln}} \mid 2 \leq \delta_{\mathrm{ln}} \leq 30\}$.

(2) Length of introduction: Users set different introductions to display personalized content and attract the attention of other users. Since many social bots are unable to set personalized introductions on a large scale, the introductions of social bots are usually simple or even missing. Thus, we define the length of introduction as δ_{li}.

(3) Ratio of followers to following: The number of following and followers can reflect the social relationship of a user. Social bots are often used to act as followers of others, and they follow a large number of users but are rarely followed by other users. The ratio of followers to following is considered in Refs. [20, 38], which is denoted in this paper as δ_{rff}, and it is given by

$$\delta_{rff} = \frac{\varphi}{\varphi + \theta}, \tag{6}$$

where φ represents the number of followers and θ denotes the number of following.

(4) Comprehensive Level of user: Sina Weibo assigns a comprehensive level to each user based on content contribution, identity characteristics, credit history, social relationships, and consumption preferences. It can be seen that the comprehensive level of users is closely related to their behaviors in OSNs, and social bots have a lower level compared to normal users. Therefore, we quantify this credit rating and denote it as δ_{cl}. The value range of δ_{cl} is $\{\delta_{cl} \mid 1 \leq \delta_{cl} \leq 5\}$.

3.3. Detection Module. This module constructs a social bots detection model based on BiGRU and attention mechanism, which consists of input layer, feature extraction layer, fusion layer, attention layer, and inference layer. The model architecture is shown in Figure 2.

The input layer receives three different feature vectors, i.e., the semantic feature sequence S_u, the metadata feature sequence M_u, and the profile feature P_u. Then, the two components included in the feature extraction layer are, respectively, responsible for further extracting features from

Table 3: Overview of the profile-based features.

Symbol	Feature name	Source
$\delta_{\ln}$	Length of nickname	[23, 48]
δ_{li}	Length of introduction	New
δ_{rff}	Ratio of followers to following	[20, 38]
δ_{cl}	Comprehensive level of user	New

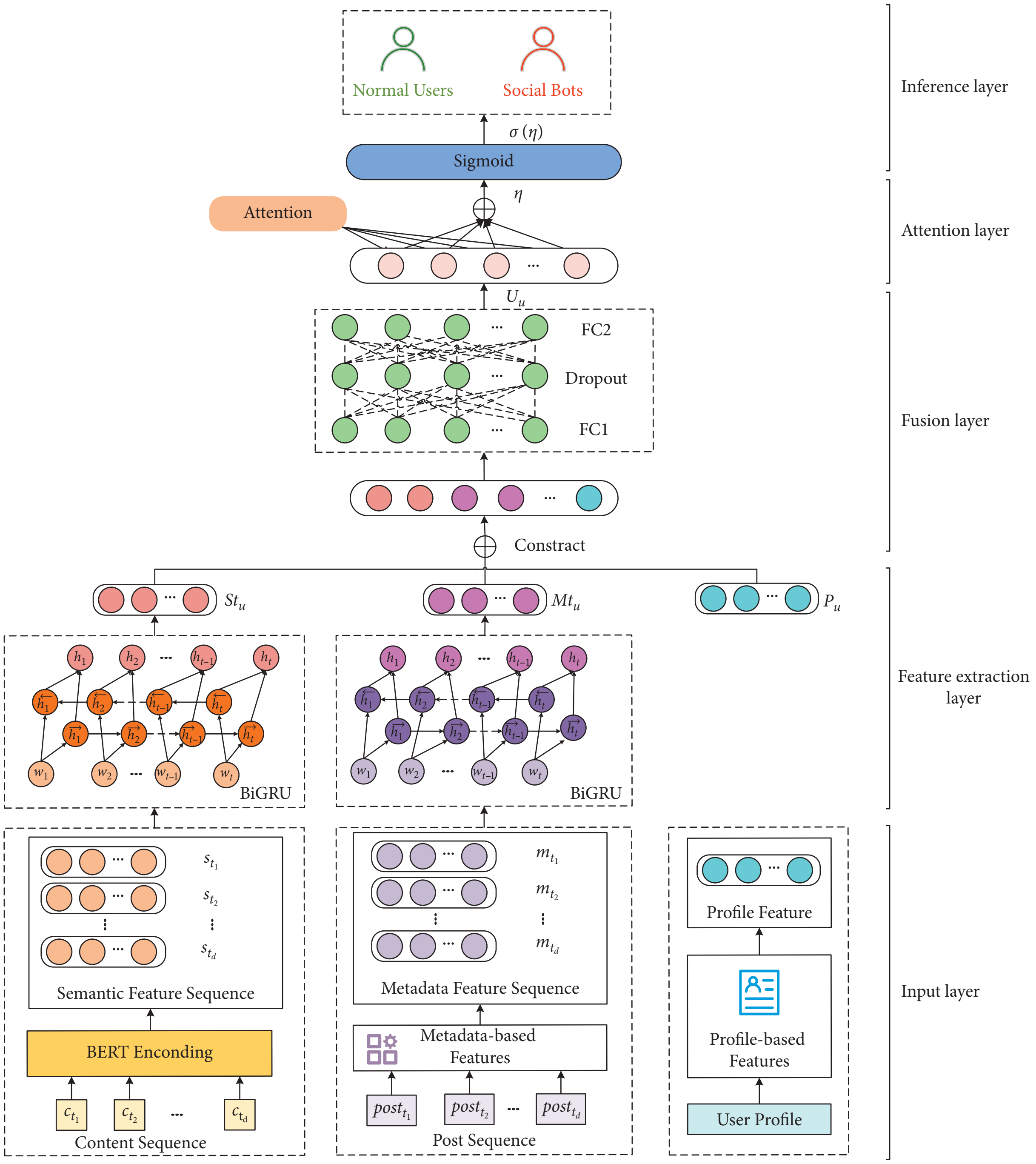


Figure 2: The architecture of the detection model based on deep learning.

the input semantic feature sequence S_u and the metadata feature sequence M_u. The details of these two components will be further described in the following sections. The fusion layer then generates the user's representation by combining the temporal-metadata feature, the temporal-semantic feature, and the profile feature together. After the fusion layer, an attention layer and an inference layer are added to obtain the classification label (social bot or normal user).

3.3.1. Feature Extraction Layer. The feature extraction layer consists of two components, namely, the temporal-metadata feature extraction component and the temporal-semantic feature extraction component. Since the posts of OSN users are time-ordered, both components use BiGRU to extract the temporal patterns of semantic feature sequence and metadata feature sequence. The features extracted in the two components are defined as St_u and Mt_u, respectively, which represent the temporal-semantic feature and the temporal-metadata feature. BiGRU consists of forward GRU and backward GRU, which can extract features from the input sequence $w_u = \{w_1, w_2, \ldots, w_d\}$ [49]. After the sequence is input to the BiGRU model, the forward hidden state output $\overrightarrow{h}_t$ and the backward hidden state output $\overleftarrow{h}_t$ at time t are denoted as

$$\begin{aligned}\overrightarrow{h}_t &= \overrightarrow{G}_{RU}\left(w_t, \overrightarrow{h}_{t-1}\right),\\ \overleftarrow{h}_t &= \overleftarrow{G}_{RU}\left(w_t, \overleftarrow{h}_{t+1}\right).\end{aligned} \tag{7}$$

The hidden state output of BiGRU at moment t consists of $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ spliced together, i.e., $h_t = [\overrightarrow{h}_t, \overleftarrow{h}_t]$. The set of hidden states $H = \{h_1, h_2, \ldots, h_d\}$ is output after the sequence training is completed.

3.3.2. Fusion Layer. In the fusion layer, the features extracted from the semantic feature sequence, the metadata feature sequence, and the profile feature are fused and a neural network consisting of two fully connected layers and a drop layer are used to generate a more accurate representation vector of the user. The representation vector of the user can be calculated by

$$U_u = \alpha_1 \cdot P_u + \alpha_2 \cdot St_u + \alpha_3 \cdot Mt_u + \beta, \tag{8}$$

where "+" indicates that the elements are summed. The weight matrices of the profile feature P_u, the temporal-semantic feature St_u, and the temporal-metadata feature Mt_u are defined as α_1, α_2, α_3, respectively, which need to be learned by the neural network. The bias vector is defined as β.

3.3.3. Attention Layer. The attention mechanism filters and focuses a small number of important factors from a large amount of information [50]. We use the attention mechanism to extract the information that is important to the classification result and achieve the purpose of improving the classification effect. Given a user representation vector $U_u = (u_1, u_2, \ldots, u_d)$, the importance weight of u_i is defined as ξ_i, and the output of the attention layer can be expressed as

$$\eta = \sum_{i=0}^{k} \xi_i u_i. \tag{9}$$

3.3.4. Inference Layer. In the inference layer, a fully connected layer with a sigmoid activation function is employed to realize binary classification. The vector η is the output of the attention layer and thus the probability that the user is a social bot can be calculated by

$$\hat{y} = \frac{1}{1 + e^{-\eta}}. \tag{10}$$

The value range of $\hat{y}$ is limited to (0, 1), so that binary classification can be realized. Significantly, a higher value of $\hat{y}$ shows that the user is more likely to be a social bot.

4. Experimental Evaluation

4.1. Environmental Setup. In our work, Tensorflow (https://www.tensorflow.org/) and Scikit-learn (https://scikit-learn.org/) were used to construct all the detection models in the experiments. All experiments were undertaken on a workstation with 128G RAM, Intel Xeon Gold 6130 2.10 GHz, and NVIDIA Tesla V100. Each experiment is repeated 10 times and the average value was calculated as the final experimental results for presentation. For the experimental dataset, the percentage of training data was set to 70% as default, and it is tuned according to real experimental situations.

4.2. Evaluation Metrics. Accuracy, Precision, Recall, and F1-score are used to evaluate the performance of the detection approaches in the experiments. The confusion matrix is shown in Table 4. The true positive (TP) is the number of samples where both predicted and actual results are social bots. The true negative (TN) is the number of samples where the predicted result is a social bot but is actually a normal user. The false negative (FN) is the number of samples where the predicted result is a normal user but is actually a social bot. The false positive (FP) is the number of samples where the predicted result is a social bot but is actually a normal user. The computational expressions of the four metrics is as follows:

$$\begin{aligned}\text{Accuracy} &= \frac{|\text{TP} + \text{TN}|}{|\text{TP} + \text{TN} + \text{FP} + \text{FN}|}\\ \text{Recall} &= \frac{|\text{TP}|}{|\text{TP} + \text{FN}|}\\ \text{Precision} &= \frac{|\text{TP}|}{|\text{TP} + \text{FP}|}\\ F1 - \text{score} &= \frac{2 \cdot \text{Precison} \cdot \text{Recall}}{\text{Precison} + \text{Recall}}.\end{aligned} \tag{11}$$

4.3. Experiments and Analysis

4.3.1. Feature Ablation Tests. In this paper, the contribution of different categories of features to the model performance will be evaluated in feature ablation tests. To verify the effectiveness of a particular category of features, a subset of the feature set will be obtained by removing the features of that

Table 4: Illustration of confusion matrix.

Actual values	Predicted values	
	Positive samples	Negative samples
Positive samples	TP	FN
Negative samples	FP	TN

category from the full feature set. The subset of features is represented by the set-difference function given as

$$F\backslash F_1 = \{x \mid x \in F \wedge x \notin F_1\}, \quad (12)$$

where F is the set of all features, F_1 is the subset of F with a particular category of features, and x is all user data of a feature. We perform feature ablation tests based on the full feature set and three subsets of the feature set. Table 5 shows the details of the feature sets we used in the feature ablation test.

The results are shown in Figure 3, and we find that the model performs best on the feature set F. This suggests that each category of features extracted in this paper is able to effectively distinguish social bots from normal users. In addition, TPBot performs worst when using the feature set of $F\backslash$Profile, which indicates that the profile feature set is the most distinguishable. This phenomenon may be caused by the feature of comprehensive level, which is scored by Sina Weibo based on different aspects. The performance of the model with the feature set of $F\backslash$Metadata is closest to that with F, which proves that the feature set based on metadata contributes the least to the model performance compared to other categories of features. We can also find that the detection model has the lowest recall when using the feature set of $F\backslash$Semantic, that is, the semantic feature set can significantly improve the recall of the model. This proves that the feature set of $F\backslash$Semantic can effectively improve the ability of the model to correctly identify social bots.

4.3.2. *Performance Comparison with the Baseline Approaches.*

In order to evaluate the performance of TPBot proposed in this paper, a performance comparison with the baseline detection approaches is carried out on the SWLD-20K* dataset. All of the baseline studies are briefly described as follows:

(i) LR: The logistic regression model is a classifier focusing on binary classification and thus can be used to construct a social bot classifier [51].

(ii) SVM: A set of important features based on user behavior and post content was proposed in Ref. [52], and a bot detection algorithm based on SVM was applied to detect social bots in Sina Weibo.

(iii) RF: Random forest algorithm is widely used in the field of social bot detection and has achieved good detection results [14].

(iv) GBDT: GBDT is an iterative decision tree algorithm with strong generalization capability. This algorithm is used in the field of social bot detection to discover multiple distinguishing features and feature combinations [33].

(v) AdaBoost: AdaBoost is employed in social bot detection and performs well in Ref. [34], which uses a 10-dimensional user vector based on user profile.

(vi) RGA: RGA [23] is a deep learning-based bot detection model, which constructs a 30-dimensional user vector in Sina Weibo.

Table 5: The description of feature sets.

Feature set	Categories of features included
F	Profile, semantic, metadata
$F\backslash$Profile	Semantic, metadata
$F\backslash$Semantics	Profile, metadata
$F\backslash$Metadata	Profile, semantic

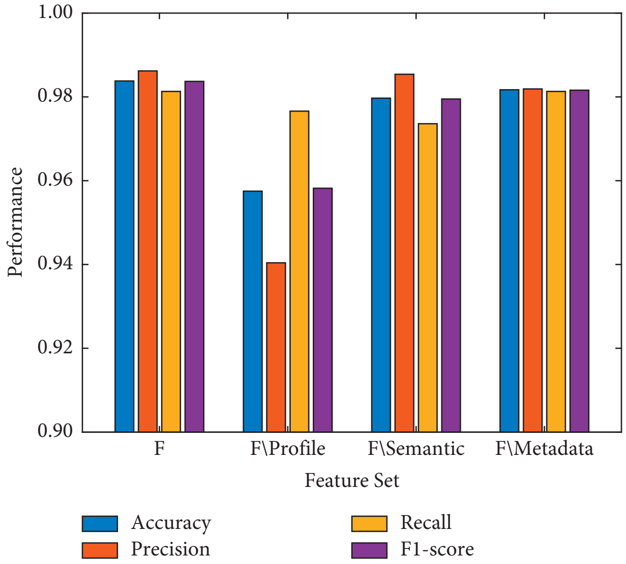


Figure 3: The performance of the proposed TPBot in the feature ablation tests.

The experimental results are shown in Table 6, and Figures 4 and 5. As shown in Figure 4, the social bot detection approach proposed in this paper outperforms the other six bot detection approaches. On the whole, the performance of deep learning approaches is better than that of classical machine learning approaches. This is because the deep learning approaches can deal with more complex data and mine hidden features. Among classical machine learning approaches, SVM performs poorly and RF performs the best, which may be because RF used the most comprehensive features to identify social bots. At the same time, RGA has a worse performance in deep learning approaches, which may be because RGA does not fully mine the latent relationship between posts. Importantly, TPBot has the best performance among these approaches on almost all metrics, with accuracy, precision, recall, and F1-score of 0.9838, 0.9862, 0.9813, and 0.9837, respectively. It proves that TPBot has great advantages over the state-of-the-art approaches in detecting social bots.

Table 6: Numerical results of the baseline approaches and proposed TPBot in detecting social bots.

Method	Accuracy	AUC	Precision	Recall	F1-score
LR [51]	0.9405	0.9405	0.9526	0.9268	0.9395
SVM [52]	0.9367	0.9366	0.9638	0.9070	0.9345
RF [14]	0.9745	0.9745	0.9819	0.9665	0.9742
GBDT [33]	0.9700	0.9699	0.9734	0.9662	0.9697
AdaBoost [34]	0.9616	0.9616	0.9615	0.9615	0.9615
RGA [23]	0.9814	0.9811	0.9859	0.9767	0.9813
TPBot	0.9838	0.9971	0.9862	0.9813	0.9837

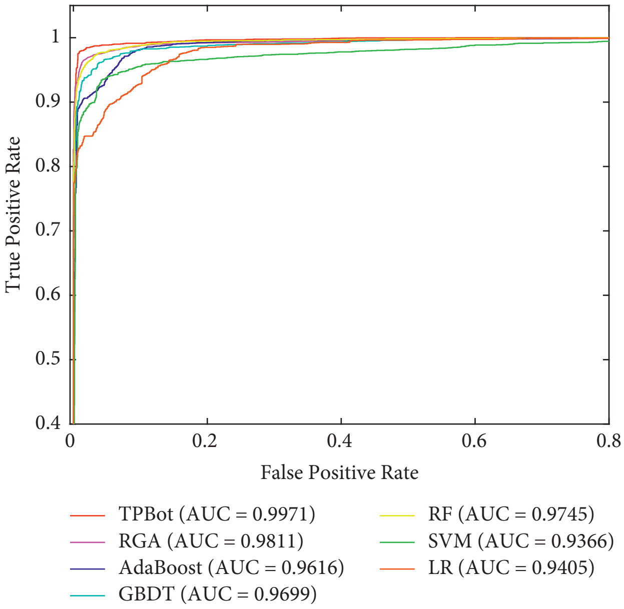


Figure 4: ROC comparison.

The ROC curve is shown in Figure 4. The area under the curve is the AUC score, which can be explained as the probability that a randomly chosen positive example (social bot) is ranked higher than a randomly chosen negative sample (normal user). Specifically, the higher the AUC score, the larger the area under the ROC curve. As indicated in the figure, TPBot has the largest AUC value of 0.9971, which proves that it has the best performance in different thresholds. The trade-off between the model accuracy and recall can be observed using the PR curve in Figure 5. The area under the PR curve is the average accuracy (AP), which describes the accuracy and recall pairs in different situations [53]. We find that TPBot has the largest AP value. Therefore, TPBot outperforms the other detection approaches.

4.3.3. Performance with Different Ratios of Positive and Negative Samples. The ratio of positive and negative samples in the training set may affect the performance of the detection model. In order to observe the performance of the model with different ratios of positive and negative samples, we construct the training dataset according to the ratio of positive samples to negative samples of 30%, 40%, 50%, 60%, 70%, and 80%, and compare the performance of TPBot and the baseline detection approaches on these datasets. The dataset with a positive sample to negative sample ratio of 30% has 3,000 social bots and 10,000 normal users, the dataset with a ratio of 40 % has 4,000 social bots and 10,000 normal users, and so on. The experimental results are shown in Figures 6(a)–6(d).

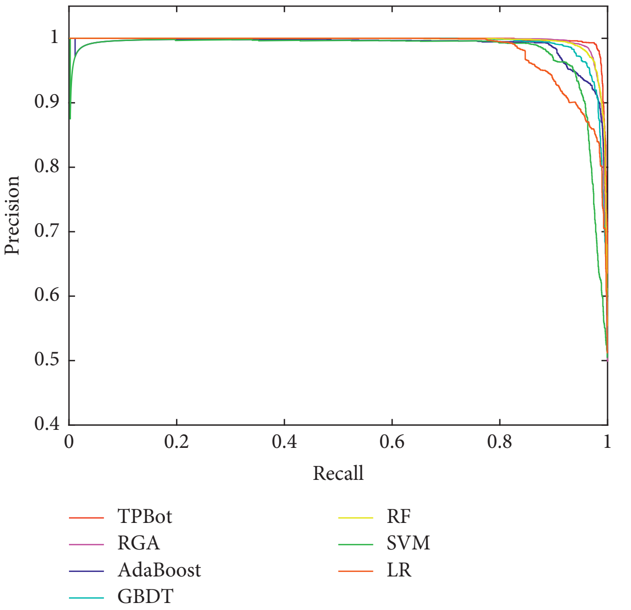


Figure 5: PR comparison.

As shown in Figures 6(a)–6(d), the proposed TPBot has the best performance on almost all metrics, at different positive and negative sample ratios. According to the results in Figures 6(b)–6(c), the precision of LR, SVM, and RF are all higher than TPBot when the positive and negative sample ratios are 50%, 60%, 70%, but the recall of all these approaches is obviously lower than TPBot at the same positive and negative sample ratios. Based on the above, it can be seen that the F1-score can more accurately evaluate the model compared with other evaluation metrics. As shown in Figure 6(d), it can be easily observed that with the increase of the proportion of social bots, the F1-scores of almost all approaches will increase. It is worth noting that, compared with other detection approaches, although the RGA performs worst when the positive and negative sample ratio is 30%, the F1-score of the RGA has a greater increase with the increase of the proportion of social bots. This proves that the proportion of positive samples and negative samples has a great impact on the performance of RGA, which may be due to the class imbalance problem. Importantly, the proposed TPBot achieved stable and outstanding performance at different sample ratios.

4.3.4. Performance in Other OSNs. In order to evaluate the performance of the proposed approach in other OSNs, we test TPBot and the baseline approaches on the two datasets collected from Twitter. The performance of these approaches is shown in Figure 7 and Table 7. The two datasets are named cresci-2017 dataset [8] and cresci-2015

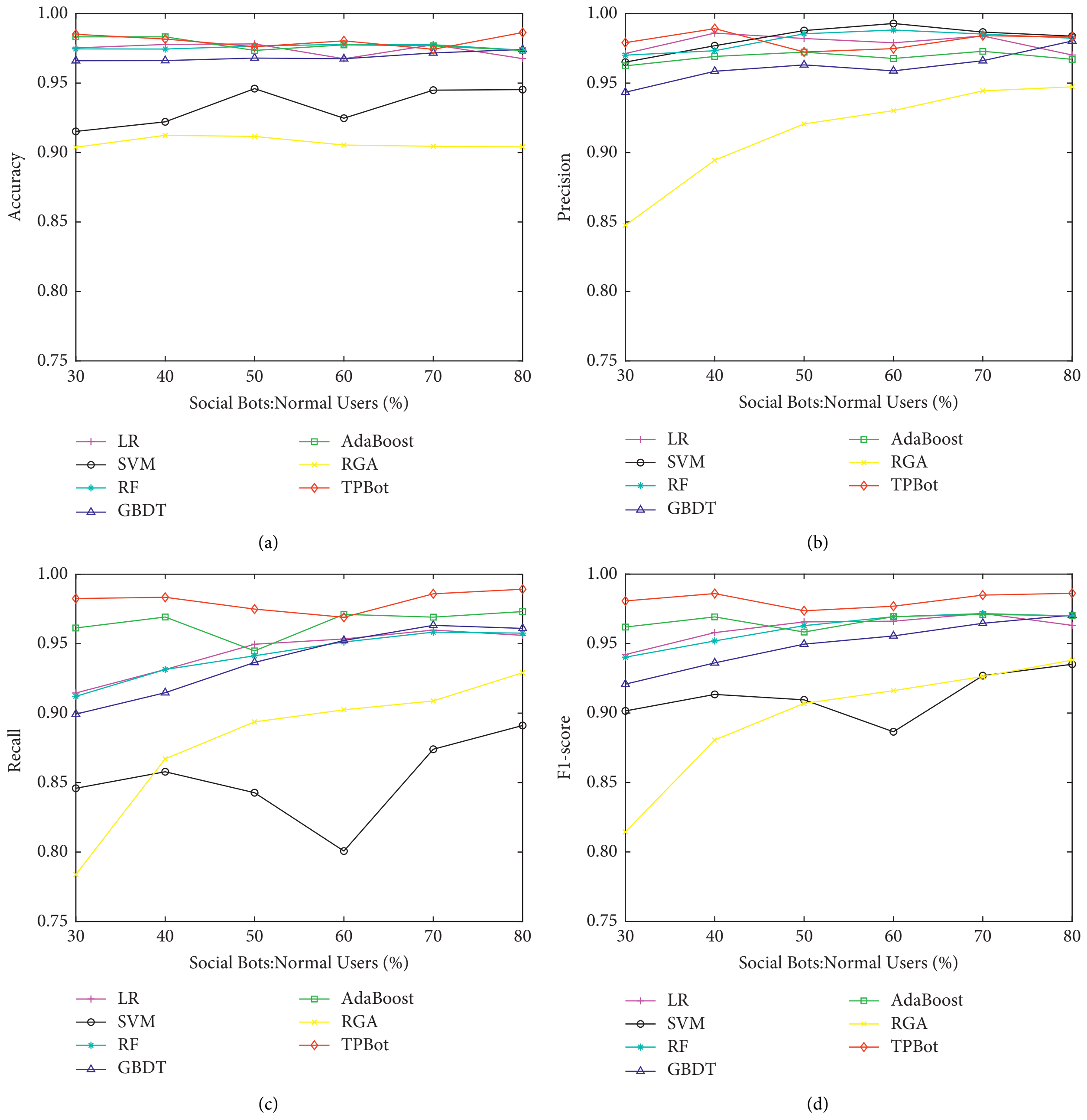
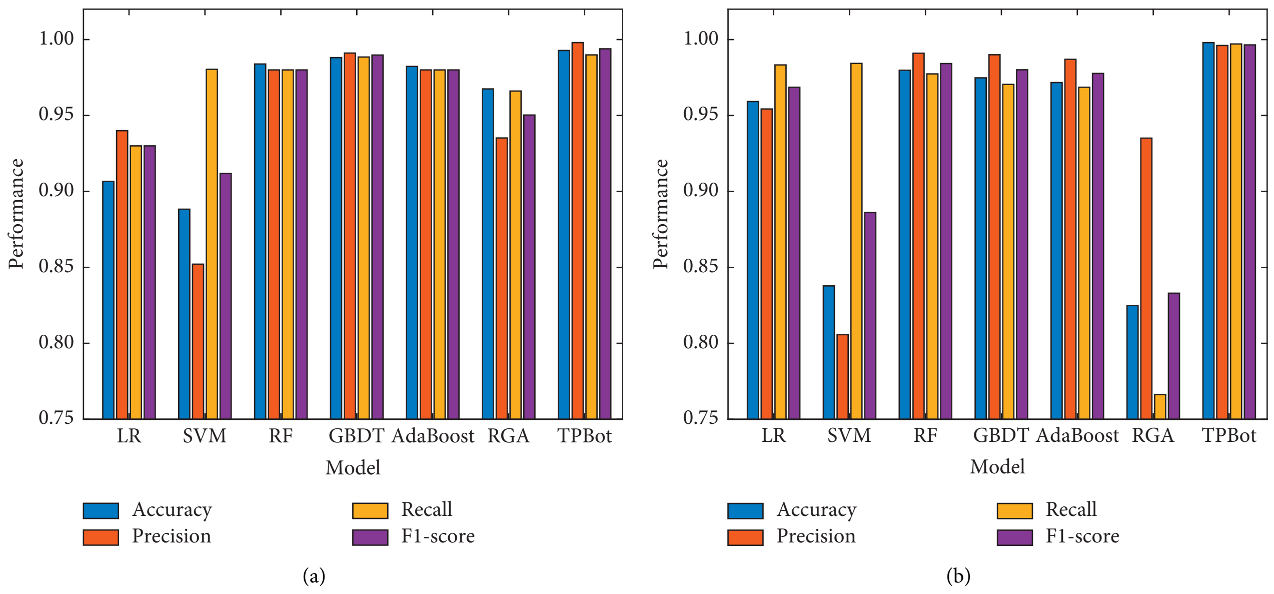


Figure 6: Results with different ratios of positive samples to negative samples. (a) Accuracy. (b) Precision. (c) Recall. (d) F1-score.

dataset [14], and their information is briefly shown in Table 8. The cresci-2017 dataset and the cresci-2015 dataset have been widely used in social bot detection in recent years. Some researchers have used the datasets (completely or partially) to evaluate their approaches for social bot detection and have achieved good detection results [25, 34, 36, 37, 54–56].

As shown in Figures 7(a)–7(b), RGA and SVM, which are built for the Sina Weibo, significantly perform worse than other detection approaches on the two Twitter datasets. This indicates the poor generalization ability of the above two models. The above result can be attributed to two aspects. Firstly, the gap between Chinese texts and English texts may bring about some errors when extracting content-based features, which are very important for the above two models. Secondly, the difference between OSNs leads to the loss of some attributes, which reduces the performance of the models. Meanwhile, compared with other detection approaches based on Twitter, the proposed TPBot has a better performance, proving its good generalization ability. In fact, the good performance of the proposed TPBot may be attributed to the use of the BERT pre-training model, which eliminates the gap between Chinese texts and English texts.



(a)

(b)

Figure 7: Performance comparison of the baseline approaches and proposed TPBot. (a) Performance comparison on the cresci-2017 dataset. (b) Performance comparison on the cresci-2015 dataset.

Table 7: Numerical results of the baseline approaches and proposed TPBot in different Twitter datasets.

Method	Cresci-2017 dataset				Cresci-2015 dataset			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
LR [51]	0.9066	0.9400	0.9300	0.9300	0.9592	0.9543	0.9833	0.9686
SVM [52]	0.8883	0.8521	0.9804	0.9118	0.8378	0.8057	0.9843	0.8661
RF [14]	0.9839	0.9800	0.9800	0.9800	0.9798	0.9910	0.9774	0.9842
GBDT [33]	0.9881	0.9911	0.9885	0.9898	0.9748	0.9900	0.9705	0.9801
AdaBoost [34]	0.9823	0.9800	0.9800	0.9800	0.9717	0.9870	0.9686	0.9777
RGA [23]	0.9675	0.9352	0.9661	0.9503	0.8249	0.9351	0.7651	0.8330
TPBot	0.9928	0.9980	0.9899	0.9939	0.9980	0.9961	0.9971	0.9965

Table 8: Overview of Twitter datasets.

Dataset	Group name	User number	Post number
Cresci-2017 [8]	Genuine accounts	3,474	8,377,522
	Social spambots	4,912	3,457,344
Cresci-2015 [14]	Humans	1,950	2,631,730
	Fake followers	3,351	196,027

5. Conclusion and Future Works

In this paper, we propose a new joint approach with temporal and profile information for social bot detection, which avoids laborious feature engineering and can construct features automatically to cope with the rapid evolution of social bots. Specifically, unlike existing studies that regard users' posts as plain text, the proposed approach treats the content of posts as temporal data. We extract the semantic information of content using the BERT language model and the latent temporal patterns between posts using BiGRU. Similarly, this paper also considers the metadata of posts as temporal data, and extracts the latent temporal patterns among the metadata of posts using a similar method. In addition, the profile feature of the user is extracted in this paper and fused with the above two features as the representation of the social bot in OSNs. The experimental results show that the approach is efficient. The proposed TPBot can effectively detect social bots, and it has the best performance compared with other commonly used detection approaches.

There are a large number of different types of social bots in OSNs, so it is one of our future research directions to analyze the behavioral intent of social bots and detect malicious social bots. In addition, social bots controlled by the same organization or individual often follow each other and mimic human accounts to form communities to avoid detection, so detecting social bot clusters is also a problem we can explore in the future.

Data Availability

The data supporting this paper are from previously reported studies and datasets, which have been cited. The processed data are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under grant nos. 61802270, 61802271, 81602935, and 81773548. In addition, this work was also partially supported by Joint Research Fund of China Ministry of Education and China Mobile Company under grant no. CM20200409, and Science and Engineering Connotation Development Project of Sichuan University under grant no. 2020SCUNG129.

References

[1] R. Tang, S. Jiang, X. Chen, H. Wang, W. Wang, and W. Wang, "Interlayer link prediction in multiplex social networks: an iterative degree penalty algorithm," *Knowledge-Based Systems*, vol. 194, Article ID 105598, 2020.

[2] W. Wang, Y. Qiu, S. Xuan, and W. Yang, "Early rumor detection based on deep recurrent q-learning," *Security and Communication Networks*, vol. 2021, Article ID 5569064, 2021.

[3] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots," *Communications of the ACM*, vol. 59, no. 7, pp. 96–104, 2016.

[4] M. Orabi, D. Mouheb, Z. Al Aghbari, and I. Kamel, "Detection of bots in social media: a systematic review," *Information Processing & Management*, vol. 57, no. 4, Article ID 102250, 2020.

[5] A. Xu, Z. Liu, Y. Guo, V. Sinha, and R. Akkiraju, "A new chatbot for customer service on social media," in *Proceedings of the 35th CHI Conference on Human Factors in Computing Systems*, pp. 3506–3510, New York; NY, USA, May, 2017.

[6] J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, A. Flammini, and F. Menczer, "Detecting and tracking political abuse in social media," *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media*, vol. 5, pp. 297–304, 2011.

[7] J. Pastor-Galindo, M. Zago, P. Nespoli et al., "Spotting political social bots in twitter: a use case of the 2019 Spanish general election," *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2156–2170, 2020.

[8] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "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*, pp. 963–972, New York; NY, USA, April, 2017.

[9] W. Shi, D. Liu, J. Yang, J. Zhang, S. Wen, and J. Su, "Social bots' sentiment engagement in health emergencies: a topic-based analysis of the COVID-19 pandemic discussions on twitter," *International Journal of Environmental Research and Public Health*, vol. 17, no. 22, p. 8701, 2020.

[10] F. Yang, Y. Liu, X. Yu, and M. Yang, "Automatic detection of rumor on sina weibo," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1–7, Beijing China, August, 2012.

[11] J. Yuan, Y. Liu, and L. Yu, "A novel approach for malicious url detection based on the joint model," *Security and Communication Networks*, vol. 2021, Article ID 4917016, 2021.

[12] A. M. Al-Zoubi, H. Faris, J. f. Alqatawna, and M. A. Hassonah, "Evolving support vector machines using whale optimization algorithm for spam profiles detection on online social networks in different lingual contexts," *Knowledge-Based Systems*, vol. 153, pp. 91–104, 2018.

[13] Q. Cao, X. Yang, J. Yu, and C. Palow, "Uncovering large groups of active malicious accounts in online social networks," in *Proceedings of the 21st ACM SIGSAC Conference on Computer and Communications Security*, pp. 477–488, Scottsdale Arizona USA, November, 2014.

[14] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "Fame for sale: efficient detection of fake twitter followers," *Decision Support Systems*, vol. 80, pp. 56–71, 2015.

[15] C. A. Davis, O. Varol, E. Ferrara, A. Flammini, and F. Menczer, "Botornot: a system to evaluate social bots," in *Proceedings of the 25th International Conference Companion on World Wide Web*, pp. 273-274, Montréal Québec Canada, April, 2016.

[16] M. Al-Qurishi, M. S. Hossain, M. Alrubaian, S. M. M. Rahman, and A. Alamri, "Leveraging analysis of user behavior to identify malicious activities in large-scale social networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 799–813, 2018.

[17] G. C. Santia, M. I. Mujib, and J. R. Williams, "Detecting social bots on facebook in an information veracity context," *Proceedings of the 13th International AAAI Conference on Web and Social Media*, vol. 13, pp. 463–472, 2019.

[18] N. El-Mawass, P. Honeine, and L. Vercouter, "Similcatch: enhanced social spammers detection on twitter using Markov random fields," *Information Processing & Management*, vol. 57, no. 6, Article ID 102317, 2020.

[19] L. Ilias and I. Roussaki, "Detecting malicious activity in twitter using deep learning techniques," *Applied Soft Computing*, vol. 107, Article ID 107360, 2021.

[20] H. Chen, J. Liu, Y. Lv, M. H. Li, M. Liu, and Q. Zheng, "Semi-supervised clue fusion for spammer detection in sina weibo," *Information Fusion*, vol. 44, pp. 22–32, 2018.

[21] 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.

[22] Y. Wu, Y. Fang, S. Shang, L. Wei, J. Jin, and H. Wang, "Detecting social spammers in sina weibo using extreme deep factorization machine," in *Proceedings of the 21st International Conference on Web Information Systems Engineering (WISE 2020)*, pp. 170–182, Amsterdam and Leiden, Netherlands, October, 2020.

[23] Y. Wu, Y. Fang, S. Shang, J. Jin, L. Wei, and H. Wang, "A novel framework for detecting social bots with deep neural networks and active learning," *Knowledge-Based Systems*, vol. 211, Article ID 106525, 2021.

[24] Y. Ji, Y. He, X. Jiang, J. Cao, and Q. Li, "Combating the evasion mechanisms of social bots," *Computers & Security*, vol. 58, pp. 230–249, 2016.

[25] M. Fazil, A. K. Sah, and M. Abulaish, "Deepsbd: a deep neural network model with attention mechanism for socialbot detection," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4211–4223, 2021.

[26] M. Kolomeets, A. Chechulin, and I. Kotenko, "Bot detection by friends graph in social networks," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, vol. 12, no. 2, pp. 141–159, 2021.

[27] Z. Guo, L. Tang, T. Guo, K. Yu, M. Alazab, and A. Shalaginov, "Deep graph neural network-based spammer detection under the perspective of heterogeneous cyberspace," *Future Generation Computer Systems*, vol. 117, pp. 205–218, 2021.

[28] B. Feng, Q. Li, X. Pan, J. Zhang, D. Guo, and Groupfound, "An effective approach to detect suspicious accounts in online social networks," *International Journal of Distributed Sensor Networks*, vol. 13, no. 7, Article ID 1550147717722499, 2017.

[29] Y. Boshmaf, D. Logothetis, G. Siganos et al., “Íntegro: leveraging victim prediction for robust fake account detection in large scale OSNs,” *Computers & Security*, vol. 61, pp. 142–168, 2016.

[30] F. Ahmed and M. Abulaish, “A generic statistical approach for spam detection in online social networks,” *Computer Communications*, vol. 36, no. 10-11, pp. 1120–1129, 2013.

[31] N. Abu-El-Rub and A. Mueen, “Botcamp: bot-driven interactions in social campaigns,” in *Proceedings of the 28th International Conference on World Wide Web*, pp. 2529–2535, Perth Australia, April, 2019.

[32] J. P. Dickerson, V. Kagan, and V. Subrahmanian, “Using sentiment to detect bots on twitter: are humans more opinionated than bots?” in *Proceedings of the 6th IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 620–627, Netherlands, November, 2014.

[33] Y. Wang, C. Wu, K. Zheng, and X. Wang, “Social bot detection using tweets similarity,” *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, in *Proceedings of the 14th International Conference on Security and Privacy in Communication Systems*, pp. 63–78, Singapore, August, 2018.

[34] S. Kudugunta and E. Ferrara, “Deep neural networks for bot detection,” *Information Sciences*, vol. 467, pp. 312–322, 2018.

[35] C. Cai, L. Li, and D. Zeng, “Detecting social bots by jointly modeling deep behavior and content information,” in *Proceedings of the 26th ACM Conference on Information and Knowledge Management*, Ireland, October, 2017.

[36] H. Ping and S. Qin, “A social bots detection model based on deep learning algorithm,” in *Proceedings of the 18th IEEE International Conference on Communication Technology*, pp. 1435–1439, Chongqing, China, October, 2018.

[37] K.-C. Yang, O. Varol, P.-M. Hui, and F. Menczer, “Scalable and generalizable social bot detection through data selection,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, pp. 1096–1103, 2020.

[38] M. Fazil and M. Abulaish, “A hybrid approach for detecting automated spammers in twitter,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2707–2719, 2018.

[39] J. Echeverria and S. Zhou, “Discovery, retrieval, and analysis of the’star wars’ botnet in twitter,” in *Proceedings of the 9th IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 1–8, The Hague, Netherlands, December, 2017.

[40] N. Chavoshi, H. Hamooni, and A. Mueen, “Debot: twitter bot detection via warped correlation,” in *Proceedings of the 2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 817–822, Barcelona, Spain, December, 2016.

[41] A. Alarifi, M. Alsaleh, and A. Al-Salman, “Twitter turing test: identifying social machines,” *Information Sciences*, vol. 372, pp. 332–346, 2016.

[42] S. Cresci, M. Petrocchi, A. Spognardi, and S. Tognazzi, “Better safe than sorry: an adversarial approach to improve social bot detection,” in *Proceedings of the 10th ACM Conference on Web Science*, pp. 47–56, Amsterdam Netherlands, May, 2019.

[43] W. Zeng, R. Tang, H. Wang, X. Chen, and W. Wang, “User identification based on integrating multiple user information across online social networks,” *Security and Communication Networks*, vol. 2021, Article ID 5533417, 2021.

[44] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: pretraining of deep bidirectional transformers for language understanding,” *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 4171–4186, 2018.

[45] S. Mohammad, M. U. Khan, M. Ali, L. Liu, M. Shardlow, and R. Nawaz, “Bot detection using a single post on social media,” in *Proceedings of the 3rd World Conference on Smart Trends in Systems*, pp. 215–220, London, UK, July, 2019.

[46] Z. Chen, R. S. Tanash, R. Stoll, and D. Subramanian, “Hunting malicious bots on twitter: an unsupervised approach,” in *Proceedings of the 9th International Conference on Social Informatics*, pp. 501–510, Oxford, UK, September, 2017.

[47] J. Pan, Y. Liu, X. Liu, and H. Hu, “Discriminating bot accounts based solely on temporal features of microblog behavior,” *Physica A: Statistical Mechanics and Its Applications*, vol. 450, pp. 193–204, 2016.

[48] O. Varol, E. Ferrara, C. Davis, F. Menczer, and A. Flammini, “Online human-bot interactions: detection, estimation, and characterization,” *Proceedings of the 11th International AAAI Conference on Web and Social Media*, vol. 11, pp. 280–289, 2017.

[49] D. Tang, B. Qin, and T. Liu, “Document modeling with gated recurrent neural network for sentiment classification,” in *Proceedings of the 12th Conference on Empirical Methods in Natural Language Processing*, pp. 1422–1432, Sapporo, Japan, July, 2015.

[50] M.-T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation,” in *Proceedings of the 12th Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, Honolulu Hawaii, October, 2015.

[51] X. Zhu, Y. Nie, S. Jin, A. Li, and Y. Jia, “Spammer detection on online social networks based on logistic regression,” *Web-Age Information Management*, in *Proceedings of the 16th International Conference on Web-Age Information Management*, pp. 29–40, Qingdao, China, June, 2015.

[52] X. Zheng, Z. Zeng, Z. Chen, Y. Yu, and C. Rong, “Detecting spammers on social networks,” *Neurocomputing*, vol. 159, pp. 27–34, 2015.

[53] T. Qiu, X. Liu, X. Zhou, W. Qu, Z. Ning, and C. P. Chen, “An adaptive social spammer detection model with semi-supervised broad learning,” *IEEE Transactions on Knowledge and Data Engineering*, no. 1–1, 2020.

[54] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, “Emergent properties, models, and laws of behavioral similarities within groups of twitter users,” *Computer Communications*, vol. 150, pp. 47–61, 2020.

[55] R. De Nicola, M. Petrocchi, and M. Pratelli, “On the efficacy of old features for the detection of new bots,” *Information Processing & Management*, vol. 58, no. 6, Article ID 102685, 2021.

[56] P. Pham, L. T. T. Nguyen, B. Vo, and U. Yun, “Bot2vec: a general approach of intra-community oriented representation learning for bot detection in different types of social networks,” *Information Systems*, vol. 103, Article ID 101771, 2022.