Knowledge-Based Systems 258 (2022) 110038

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

SybilFlyover: Heterogeneous graph-based fake account detection model on social networks

Siyu Li [a], Jin Yang [a,*], Gang Liang [a], Tianrui Li [b], Kui Zhao [a]

[a] School of Cyber Science and Engineering, Sichuan University, Chengdu, 610207, Sichuan, China
[b] School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, 611756, Sichuan, China

ARTICLE INFO

Article history:
Received 10 June 2022
Received in revised form 13 October 2022
Accepted 14 October 2022
Available online 21 October 2022

Dataset link: http://mib.projects.iit.cnr.it/dataset.html

Keywords:
Sybil detection
Fake user
Heterogeneous graph

ABSTRACT

Organized social robot accounts can launch Sybil attacks on online social networks (OSNs) for various malicious purposes, thus significantly affecting the user experience of online communities and damaging the reputation of OSNs. Therefore, detecting Sybil accounts in OSNs is crucial in cyberspace governance. Among the existing Sybil account detection methods, structure-based methods appear to be the most promising because of their ability to model the states and behaviors of social network accounts effectively. However, most of these structure-based methods only model the social accounts in OSNs while ignoring the content generated in OSNs, such as tweets of Twitter accounts. To address this deficiency, this study proposes an efficient model called SybilFlyover, which uses a heterogeneous graph-based method to represent all types of entities existing in an OSN uniformly, as well as complex relationships between the entities in a directed heterogeneous social network graph. During the modeling process, content-based social network information is injected into the model using a method based on prompt learning to achieve more accurate modeling of the real state of an OSN. Finally, the social network graph was processed using a transformer-based method to identify nodes representing Sybil accounts. The results of an experiment on a real public dataset demonstrate that the proposed SybilFlyover model outperforms existing state-of-the-art baseline models in Sybil account detection.

1. Introduction

The recent advent of online social networks (OSN) has reshaped communication between individuals and society. Activities such as making friends, exchanging information, and even releasing news have been increasingly conducted with the help of OSNs. The OSN platforms provide an increasing number of roles, and mainstream OSN platforms even act as a de-facto information infrastructure. However, OSNs represented by Twitter are highly vulnerable to Sybil attacks. A large number of organized social bots are known as Sybil accounts when the accounts are fake and not stolen from legitimate users [1,2]. On the Internet, these accounts can cause various problems that severely affect the normal use of OSNs. For instance, study [3] showed that 71% of retweeting users were classified as bots, of which 37% were suspended by Twitter. It has been estimated that approximately 15% of Twitter accounts are Sybil accounts [4]. A group of organized Sybil accounts can be manipulated to publish advertisements on OSNs, perform organized online fraud activities, collect information without authorization, and even interfere with elections using misleading [5]. These activities will not only have a negative impact on the order of the online community in general but can also affect user experience, resulting in a negative impact on OSN platforms [6]. Therefore, the development of effective methods for detecting Sybil accounts in OSNs is crucial and urgently needs to be addressed.

Many scholars have conducted research on this topic and have proposed various Sybil account detection methods. Currently, the vast majority of Sybil account detection methods can be categorized into two classes: feature-based methods and structure-based methods. Feature-based methods mainly detect abnormal profile features of Sybil accounts to differentiate them from normal nodes in an OSN. In contrast, structure-based methods make judgments based on the abnormal behaviors of the accounts in an OSN. In addition, structure-based methods use graph-structured data to model the social network structure and behavior of social network accounts. However, they used different techniques to process social network representation graphs to identify nodes that represent Sybil accounts. Structure-based methods can be further divided into random-walk-based methods [7–13], label-propagation-based methods [14–17], and graph neural network (GNN)-based methods [18–20], according to the graph processing technique used.

However, most structure-based methods focus on social accounts and their behaviors when modeling an OSN while ignoring

* Corresponding author.
E-mail address: yangjin66@scu.edu.cn (J. Yang).

https://doi.org/10.1016/j.knosys.2022.110038
0950-7051/

the content generated by the OSN. These methods [7–12] generate a homogeneous social network graph based on the information about the accounts in an OSN and the relationships between the accounts (e.g., follow an account), and then perform various optimization operations by updating the weights of the edges. The methods proposed in [14–20] introduce assumptions from various perspectives, define some labels or scores for account nodes, and set the corresponding propagation rules before processing the homogeneous social network graph and identifying Sybil nodes in the graph. The method presented in [13] uses a heterogeneous graph to represent the friendly relationships between the accounts in an OSN uniformly and the interactions between them, but ignores the content entities in the OSN. In addition, it does not introduce content-related information into the model to enhance its ability to grasp the real state of an OSN.

We believe that the introduction of content-based information as a supplementary measure to the modeling of OSN accounts and their relationships (i.e., the texts generated by social network accounts during various activities and the interactive relationships, such as create, like, and retweet, between the text content of different accounts) can help the Sybil detection model represent OSN more accurately, thus creating a condition for more accurate detection of the differences between Sybil accounts and ordinary accounts. In the SybilFlyover model proposed in this study, the important entities in an OSN (such as Twitter) are divided into two categories: users and posts. All sorts of entities and events in OSN are uniformly represented in a heterogeneous social network graph $\langle users, tweets, relations\rangle$ so as to provide richer information and a more reliable foundation for the model to identify Sybil nodes.

The main contributions of this study can be summarized as follows:

- A method based on a heterogeneous graph was used to accurately model complex relationships in the Twitter social network. The proposed SybilFlyover model overcomes a major limitation of traditional structure-based Sybil detection models that can model OSNs only from the perspective of user accounts, thus allowing uniform modeling of various accounts' behaviors in an OSN using a heterogeneous graph.
- A content-based text enhancement representation method was used to introduce more information into the model from the perspective of tweet content, making it easier for the model to detect fake social accounts. The proposed SybilFlyover model can extract feature information and model OSNs more precisely by using a method based on prompt learning to obtain the feature representations of account and tweet text, and then assign the representations to the nodes and edges in the heterogeneous graph.
- A heterogeneous graph neural network based on the attention mechanism is used to process the heterogeneous OSN representation graph. Graph feature abundance was incorporated into the model to achieve a good balance between the rich information of the social network graph and the massive computing workload involved in processing the features in a heterogeneous graph. The proposed SybilFlyover model can achieve an accuracy of 99.81% in Sybil account detection tasks in an experiment on a real public dataset, demonstrating that SybilFlyover is superior to the existing methods of [9,12,13,16,18–21].

2. Related works

2.1. Feature-based methods

Feature-based Sybil account detection methods, also known as content-based methods, make judgments primarily based on malicious user-level behaviors, focusing on the feature differences between normal accounts and Sybil accounts (such as nickname style and self-description in the user profiles of social network accounts, the records of abnormal activities of users, etc.). Feature-based methods can be further classified into traditional machine learning-based and deep learning-based methods according to the method employed.

Methods based on traditional machine learning: This type of method was used in early studies in the detection field. These methods use the prediction results of traditional machine learning classifiers, such as SVM, as a basis for detecting Sybil accounts. The need for Sybil account detection has arisen with the broad use of social networks. Chu et al. [22] devised and studied a feature-based Sybil detection method for Twitter. They used a decision tree to distinguish normal accounts from robotic accounts from the perspective of entropy. Yang et al. [23] trained the SVM classifier to identify Sybil accounts using their activity characteristics, such as the frequency of friend requests and probability of request acceptance. Miller et al. [24] used stream clustering algorithms StreamKM++ and DenStream to identify spam senders in social networks from the perspective of anomaly detection, rather than classification. Xiao et al. [25] proposed a cluster-based approach for finding groups of fake accounts registered by the same attacker from statistical patterns. Varol et al. [4] proposed a social robot detection model containing more than 1000 features based on the results of extensive feature engineering on a large number of accounts and their behaviors, indicating that approximately 15% of active Twitter accounts were Sybil accounts. Sahoo et al. [21] applied random forest and bagging-based methods to detect fake and troll profiles and integrated this detection model into a Chrome extended application, achieving good results. Walt et al. [2] explored the possibility of applying Sybil account detection methods to fake human accounts, also known as troll accounts, and obtained an F1-score of 49.75%, indicating that the processed features that had been previously used to detect fake accounts generated by bots were not similarly successful in the detection of fake accounts generated by humans.

Methods based on deep learning: Deep learning models represented by CNN have achieved success in many fields, including computer vision. Unlike conventional machine learning methods, deep learning models do not require cumbersome feature engineering and exhibit better generalization performance. Therefore, they have been widely used for Sybil account detection. Kudugunta et al. [26] inputted metadata in Twitter account profiles and tweets posted by accounts into an LSTM-based model to detect robot accounts. The authors of another study [27] proposed a ResNet-based social robot detection model with an attention mechanism, and validated the effectiveness of the method on real data from Sina Weibo. Feng et al. [28] proposed the BotFlowMon model, which focuses on the network traffic generated by OSN accounts during their activity. This model can identify Sybil accounts according to network traffic characteristics without the need to analyze users or the content of the tweets. Wanda et al. [29] proposed DeepProfile, which uses a method based on a Dynamic-CNN to identify fake profiles in OSNs. They further proposed a Sybil detection model, DeepFriend [30], which combines DeepProfile with a social graph and uses an ordinary CNN, rather than GNNs, to process social graphs, where the input is a feature gathering the neighborhood link information, not the whole graph.

In summary, feature-based detection methods can effectively identify Sybil accounts in an OSN based on the behavioral characteristics of each account, and they have a relatively high detection efficiency. However, several published studies have shown that some Sybil accounts can acquire the ability to mimic the behaviors of normal accounts as social robots constantly evolve, particularly the target accounts of their malicious behaviors. This implies that the latest generation of social robots can evade the detection performance of feature-based detection methods by avoiding the generation of malicious features.

2.2. Structure-based methods

Structure-based detection methods typically model user accounts and their interactive behaviors in a social network and determine whether the OSN account being probed is a Sybil account according to the malicious behaviors perpetrated by the account against other accounts (e.g., abnormally following a large number of accounts) or the abnormal behaviors of this account (e.g., a newly registered account posting a large number of tweets in a short period), which represents the applications of graph-based anomaly detection (GBAD) [31]. Owing to their inherent characteristics, graph structure data are ideally suited for modeling structural relationships in social networks, therefore, these data have often been used in structure-based detection methods [3]. In real-world applications of detecting Sybil accounts in OSNs, a social network relationship graph has usually been used. In a social network relationship graph, nodes represent users, including normal users and Sybil users, and edges illustrate the behaviors of users and their relationships (e.g., friend relation and follow action). Based on the modeling and analysis of the social network graph, these models classify the nodes as normal users or Sybil users.

The method based on the random walk: Yu et al. [7] first proposed the SybilGuard, a graph-based social network robot detection method. This method employs a random walk algorithm to analyze the social network graph and identify abnormal nodes among the marked negative sample nodes (i.e., non-Sybil normal users). Later, they proposed an improved model called SybilLimit [8], that adopted the hypothesis expressing a short random walk starting from a Sybil node could easily reach other Sybil nodes, while the hypothesis expressing a short random walk starting from a marked normal user node would be difficult to reach other normal user nodes. The effectiveness of SybilLimit was experimentally confirmed. SybilRank [9] is a random walk-based method in which the concept of benign scores is introduced into the social network graph to quantify the difference between the user node being probed and the known normal user nodes. The opposite hypothesis, expressing that badness scores should be used to evaluate the difference between the user node being probed and Sybil accounts to identify Sybil accounts, was adopted in the CIA method proposed by Yang et al. [10]. In another study [11], the Integro model focuses on the victims of Sybil accounts (i.e., the real users interacting with the robot accounts) in the process of fake account detection. This method can achieve accurate identification results, and is applicable to large-scale OSNs. The SmartWalk model proposed by Liu et al. [12] uses an improved random walk algorithm applied to a social network graph, allowing an adaptive step-length adjustment according to the initial node and path. The SybilSAN model proposed by Zhang et al. [13] employs a two-layer hypergraph to describe users and their behaviors in a social network, and combines three random walk-based algorithms to identify Sybil accounts.

Method based on label propagation: The SybilBelief model proposed by Gong et al. [14] processes social network graphs using a method based on loop belief propagation. This method adopts the concept of label propagation and the Markov random field (pMRF) model. When normal and Sybil users are known, the model first assigns a priori probability to each user node in the social network graph as its label and then, for each user node, calculates the posterior probability of the node being a Sybil node using the LBP iteratively. GANG [15] used directed graphs to model social networks and successfully addressed several deficiencies of SybilBelief. The SybilSCAR [16,17] model proposed by Wang et al. uses the LBP algorithm to obtain a priori knowledge from both known normal users and Sybil users, and can ensure scalability when processing social network graphs.

GNN-based method: Considering the spectral domain, Furutani et al. [18] used a GNN model based on low-pass filtering to process social network graphs. Sun et al. [19] introduced GCN into social network graph processing and developed the TrustGCN method. The proposed method exhibits relatively high performance and is robust against data noise when detecting Sybil accounts. BalaAnand et al. [20] proposed EGSLA using a graph-based semi-supervised learning algorithm and adopting reasonable assumptions to spot fake users on Twitter.

2.3. Other methods

In addition to the above-mentioned methods, there are other methods [32–34] that use relevant information provided during the account registration process to judge whether the account being probed represents a Sybil account. These methods have proven to be effective on some OSN platforms, such as WeChat and Facebook. In addition, these methods can identify Sybil accounts at the early stage of each account's life cycle, but it is challenging for them to meet high requirements, such as acquiring the IP address and device identification code of a device when sending a registration request.

3. Preliminaries

In this section, the theoretical basis of the heterogeneous graph and prompt learning is presented.

3.1. Heterogeneous graph

In recent years, graph structure data has attracted considerable attention because this format of data can represent knowledge and information about a non-Euclidean space in the form of (*entity*, *relationship*) [35], making it possible to model real-world problems using a graph-based system [36]. In addition, this type of data has a more extensive information representation ability than data in traditional European space (e.g., pictures, text, and other information with fixed dimensions and structure) and natural advantages in applications where irregular structure information needs to be expressed (e.g., social network relationships, chemical molecular structure, knowledge map). When a method based on graph structure data is combined with a graph calculation model (e.g., GCN [37], GraphSAGE [38]), which can overcome the anisotropy of data and process undirected graphs, or a graph calculation model (e.g., GAT [39]), which employs a self-attention mechanism and processes directed graphs, good results can be obtained in tasks of node classification, edge prediction, and graph embedding.

However, graph-structured data parts are usually homogeneous graphs that contain only one type of node and one type of edge in a graph, which restricts the ability of the graph to model data, because real-world objects are complex and diverse, and their differences cannot be reflected well in homogeneous graphs [40]. As shown in Fig. 1, in an OSN such as Twitter, there are at least two types of entities: users and tweets, and the relationships include following an account, commenting on a tweet, and so on. A heterogeneous graph overcomes the limitations of limited representation ability. A heterogeneous graph structure data piece $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is composed of two sets: a node set $\mathcal{V}$ and an edge set $\mathcal{E}$, where the edges connect pairs of nodes. Unlike homogeneous graphs, heterogeneous graphs also have a node-type mapping function $\phi : \mathcal{V} \rightarrow \mathcal{A}$ and an edge-type mapping function $\psi : \mathcal{E} \rightarrow \mathcal{R}$, where $\mathcal{A}$ and $\mathcal{R}$ denote the predefined node and edge types respectively, and it holds that $|\mathcal{A}|+|\mathcal{R}| > 2$. However, in homogeneous graphs, this relationship

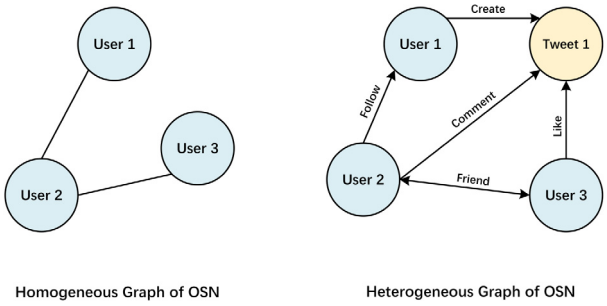

Fig. 1. Comparison of Graphs.

is $|\mathcal{A}| + |\mathcal{R}| = 2$. With the emergence of models with heterogeneous graph processing abilities, such as HAN [41], heterogeneous graph-based models have been used to model complex relationships and solve practical problems [42,43]. It is also worth noting that Yin et al. [44] proposed a method to incorporate multimodal data into the homogeneous graph for retweet time prediction. The method of using the co-embedding strategy, inputs the graph embedding and the representation of multimodal data into the hidden layer in parallel (while in a heterogeneous graph-based approach, additional information is given to different types of nodes before obtaining the representation of the graph), which is another way to introduce more information into the graph structure data.

3.2. Prompt learning

Text semantic representation is a basic NLP (Natural Language Processing) task. A piece of text must be transformed into the form of a feature vector, and the semantics contained in the text must be extracted into a dense vector during this process, such that the feature vector can be understood and processed by relevant models in subsequent tasks such as text classification and causal inference. Traditional semantic representation methods include the Word2Vec [45] model based on COBW and the BERT [46] model based on transformer [47] architecture. However, other studies [48] demonstrate that traditional text representation models represented by BERT have a significant deficiency in text embedding: they cannot fully represent semantics.

Using the BERT model as an example, its transformer [47] architecture employs masked language modeling (MLM) to extract the semantic representation capability from the training corpus. The MLM uses a semi-supervised method to hide some texts in the training materials using a [mask] token and then creates a model to predict the text to achieve the purpose of learning semantic information. However, BERT acquires anisotropy as a result of this method [48]. In addition, embedding offset and other issues also affect the text-representation ability of the model.

The prompt learning [49] method uses a more intuitive and explanatory approach to enable the model to learn text semantics. Prompt learning constructs the prompt template "[x] means [mask]" based on the BERT model to prompt the model to output a text representation vector. [x] is the input text placeholder and [mask] is the mask placed to expect the text representation obtained by the model. In real-world applications [50], we select the $top-k$ tokens in the input text according to the vector $h_{[MASK]}$ hidden by the mask and the MLM classification header of the BERT model. The respective weights of the k tokens are obtained according to the probability distribution, thus, the representation vector h of the text is obtained as follows:

$$h = \frac{\sum_{v \in v_{top-k}} W_v P\left([MASK] = v \mid h_{[MASK]}\right)}{\sum_{v \in v_{top-k}} P\left([MASK] = v \mid h_{[MASK]}\right)} \quad (1)$$

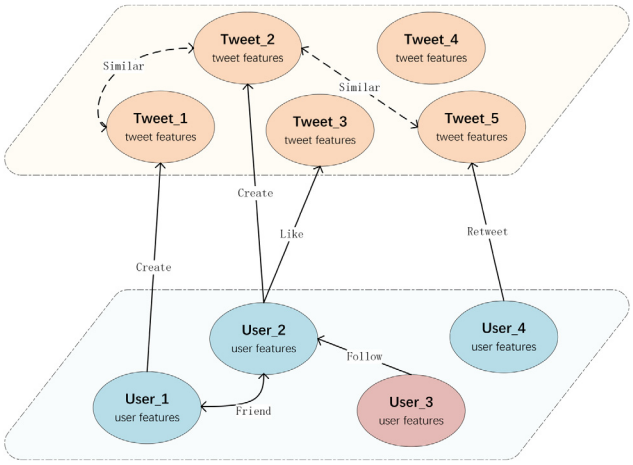

Fig. 2. Flyover-like Architecture.

where v represents the BERT tokens in the $top-k$ token set v_{top-k}, and W_v is the representation vector of the original text about v generated by BERT. Studies on text representation models using prompt learning-based methods, such as PromptBERT [50], demonstrate that the text representation ability of the model after undergoing prompt learning is significantly better than the original model using the MLM method.

4. Problem definition

4.1. Heterogeneous graph-based OSN modeling

Because an OSN usually contains a large amount of data of various types and complex associations exist between data pieces, traditional Sybil detection models based on homogeneous graphs can model only the users and their relations in an OSN but cannot fully represent the information in the OSN, limiting their performance in detecting Sybil accounts. To overcome this limitation, this study proposes the SybilFlyover model, which uses a directed heterogeneous graph to model a variety of entities and relationships in OSNs.

For a directed heterogeneous social network graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $v \in \mathcal{V}$ denotes nodes in the graph, obtained by mapping the entities in a real social network in a graph; $\delta = (v_1, v_2) \in \mathcal{E}$ is an edge from node v_1 to node v_2 in the graph, obtained by mapping the relationship between entities in the real social network into the graph; $|\mathcal{V}|$ and $|\mathcal{E}|$ are the numbers of nodes and edges in the social network graph $\mathcal{G}$, respectively. In this study, two types of entities are defined in the Twitter social network: users and tweets, as well as the interaction between the two types of entities. Accordingly, two types of nodes are defined in a social network graph $\mathcal{G}$, user nodes and tweet nodes, as well as the heterogeneous edges connecting different types of nodes and their corresponding attributes.

To obtain a clearer representation, the heterogeneous graph was mapped to three-dimensional space, as shown in Fig. 2. In the figure, the relationships between the user subgraph and users are projected onto the lower layers, which represents the modeling method used by the traditional Sybil detection models based on a homogeneous graph, whereas the tweet subgraphs are projected onto the upper layers. The tweet subgraphs are associated with the user subgraphs as a result of the user's behavior in tweets (e.g., tweeting, forwarding, and likes), thus conferring more information to the model. After such mapping, the resulting social network graph $\mathcal{G}$ exhibits a structure similar to the urban

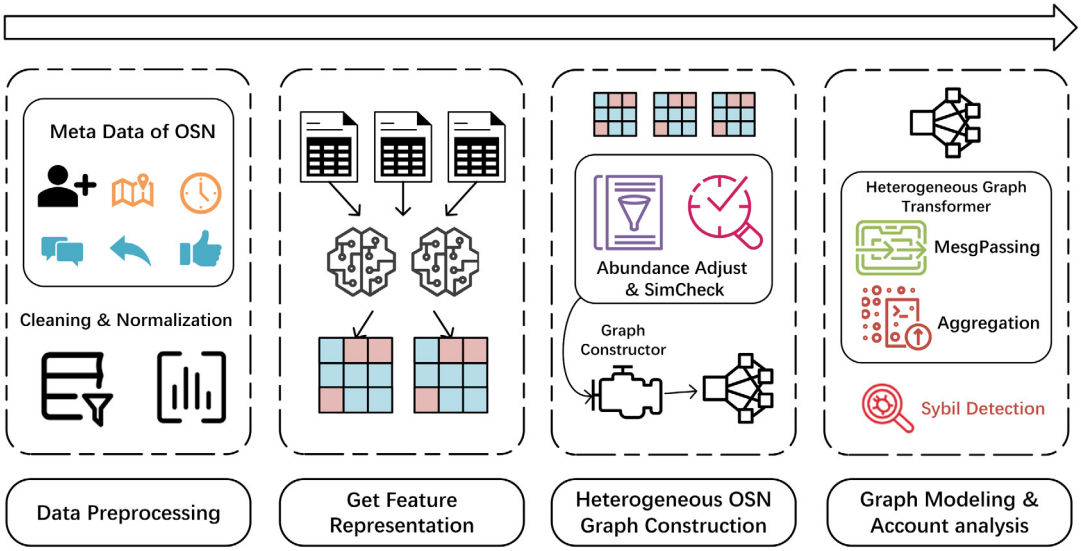

Fig. 3. Framework of SybilFlyver.

three-dimensional transportation network, which has been the main inspiration for the naming of our work. We believe that by adding the data of Twitter entities and related behaviors, we can inject more real information into the social network graph $\mathcal{G}$ on the dimensions of content and structure, which can be used to improve the model's ability to model social networks, thus improving the model performance in detecting Sybil accounts.

4.2. Design goals

The following design goals were set for SybilFlyover:

1 **Full use of information from both users and tweets to model social networks.** When a heterogeneous graph is used to model social networks, the aim is to prove that the user-tweet joint modeling method based on a heterogeneous graph used by the SybilFlyover model can reflect real relationships in social networks better than traditional user-only modeling methods based on a homogeneous graph or modeling methods adopted by other SOTA detection models. In other words, SybilFlyover can achieve better performance in detecting Sybil accounts than the other models.

2 **Accurate text semantic extraction.** The main idea of this study is that the introduction of Twitter nodes and their relationships with users can enhance the performance of Sybil account detection. However, to validate this idea, it is necessary to appropriately extract relevant information from tweets and proceed to the model. A prompt learning method-based model was used in this study to extract semantic information from the tweets. This model should have a performance level equal to or exceeding that of the traditional Transformer-based model.

3 **Calculation feasibility and expansibility.** After tweet-related information is added to the model to enable it to reflect the real situation in a social network more accurately, the data scale of the model also increases. For instance, the dataset used in this study contained 3900 pieces of user information and 2.75 million pieces of tweet information. Therefore, the feasibility of this method must be analyzed. In addition, because an OSN generates data in large quantities every hour and changes dynamically, the proposed model should be scalable; namely, its performance fluctuation should be within an acceptable range, even when the amount of data to be processed in a task changes significantly.

5. Proposed framework

5.1. Overview of SybilFlyover

The SybilFlyover model proposed in this study consists of four modules, as shown in Fig. 3. First, the data from the MIB dataset were cleaned and preprocessed using a data preprocessing module. Subsequently, the text information in the data (e.g., nickname, self-description of each Twitter user, and tweet text) is represented by the feature representation module in the form of feature vectors. The graph construction module then creates a heterogeneous OSN representation graph using the feature vectors and other data. Finally, the graph analysis module uses a method based on an attention mechanism to model the feature graph and determine whether the probed users are Sybil accounts. In the following section, each module in SybilFlyover is described in detail.

5.2. Data preprocessing

In OSNs represented by Twitter, the dimensions of the user-generated data can vary significantly, which is also reflected in the public dataset used in this study. In addition, noise-containing data samples with different weights may cause oscillations in the convergence and affect the generalization performance of the model. Therefore, it is necessary to clean the original data and use an appropriate feature scaling method to perform global scaling on the features contained in the dataset to avoid the aforementioned issue.

The association relationships in the dataset were cleaned to remove invalid relations between OSN entities. For a series of relations $\delta_i = (v_{i1}, v_{i2})_{i=1,2,\ldots,n}$, their validity is determined by assigning them to a valid mask, which is defined as follows:

$$\mathcal{E} = \delta_i \bigoplus Avail\_mask_i \tag{2}$$

where

$$Avail\_mask_i = \begin{cases} 1, & (v_{i1} \in \mathcal{V}) \cap (v_{i2} \in \mathcal{V}) \\ 0, & (v_{i1} \notin \mathcal{V}) \cup (v_{i2} \notin \mathcal{V}) \end{cases} \tag{3}$$

For feature scaling, Z-score normalization [51] was used to scale the cleaned data to ensure that the standard deviation of the feature distribution was one, and the mean value was zero. The corresponding operation can be expressed as follows:

$$\bar{x}_{(m,n)} = \frac{x_{(m,n)} - \mu_n}{\sigma_n} \tag{4}$$

where (m, n) is the nth feature of the mth entity; $x_{(m,n)}$ represents the original data; $\bar{x}_{(m,n)}$ denotes the normalized data; and μ_n and σ_n represent the mean and standard deviation of the nth feature, respectively.

5.3. Get feature representation

In the public dataset used in this study, various data samples described Twitter users, tweets, and the relationships between them from different perspectives. For instance, digital data can describe the number of tweets forwarded, thus reflecting the activity level of a particular tweet. Category data can be used to determine the time zone of a particular Twitter user, thus revealing hidden user information. Namely, an account that often posts tweets at midnight in the local time zone is more likely to be deemed as a Sybil account. This type of information can be encoded relatively easily and linked to corresponding users or tweets as feature information. For the number-type feature of value n or features of category type $\{X_i\}_{i=1,\ 2,\ \ldots,n}$, and the corresponding features Y and Z, the feature presentation can be obtained as follows:

$$Num\_feature = \bigcup_{i=1}^{n} (Loc(X_i) \cap Loc(Y_i) \cap Loc(Z_i)) \tag{5}$$

where $Loc\ (x)$ is the location function used to obtain the correspondence relationship between features and nodes.

However, the dataset used in this study contains a significant amount of textual information, such as the content information of tweets and the usernames of Twitter accounts. The semantic information contained in these texts must be extracted by means of encoding and put into semantic tensors so that it can be correctly understood and processed by the model in the subsequent modules. This process is also called a semantic representation. Traditional semantic representation models, such as BERT, can accomplish the task of extracting semantic information from texts and putting it into dense tensors. However, their semantic representation capabilities require further improvement. In SybilFlyover, we introduced the latest prompt learning method into this subtask and used promptBERT [50] to obtain a better representation of text information. For text features X and Y with the number of n respectively, the feature representation is obtained as follows:

$$Text\_feature = \bigcup_{i=1}^{n} (PromptBERT(X_i) \cap PromptBERT(Y_i)) \tag{6}$$

where, $PromptBERT\ (x)$ is the output representation tensor of the PromptBERT model for text x.

5.4. Construction of heterogeneous OSN graph

In this module, the feature tensors of the nodes and edges are combined to form a graph, yielding a heterogeneous representation of the target OSN. However, the dataset used in this study contains 34 features related to Twitter accounts and tweets. If all features are included in the OSN representation graph, the model will be extremely large, and the running speed of SybilFlyover will be reduced. Therefore, we propose to introduce the graph feature abundance λ into this module. The purpose of this measure is to introduce features of an appropriate scale into the model to improve the Sybil account detection performance of the model on the premise of ensuring calculation feasibility.

First, a series of necessary features, such as the nickname, the number of followers of a particular user account, and content of tweets, are defined. These features are the main indicators that can be used to describe the entities in OSN, and constitute the main part of the feature map. Second, a series of gain features, such as the region and time zone in which a particular user account is located, are defined. These features are supplementary descriptions of the entities in an OSN. The incorporation of these features facilitates accurate modeling of the OSN. The graph feature abundance $\lambda \in [0, 1]$ was used to control the number of gain features ultimately used to construct a heterogeneous social network graph, as shown in Algorithm 1. SybilFlyover can dynamically control the number of features embedded in the social network graph by adjusting the value of the graph feature abundance and can use as many features as possible to accurately describe the OSN from a different perspective while ensuring the calculation feasibility of the constructed graph. In this study, we set $\lambda = 0.86$.

Algorithm 1 Feature selection with Graph feature abundance

Input: Set of all feature embedding matrix: *feature_accessed*
Graph feature abundance λ
Output: Set of feature embedding matrix used: *feature_used*

for $feature_i$ **in** *feature_accessed* **do**:
  $feature\_score = SHAP(feature_i) = \{\xi_1, \xi_2, \ldots, \xi_n\}$
end for
// Get importance scores of features using SHAP

feature_used $\Longleftarrow$ *neces_features* **in** *feature_accessed*
for *gain_features* **in** *feature_accessed* **do**:
sort(*necessary_features*) **by** $\xi_i \bigotimes size(gain\_feature_i)$
feature_used $\Longleftarrow$ *gain_features* **under** *Capacity* $* \lambda$
end for
// Select gain feature based on feature score and feature size

return *feature_used*

After selecting the features to be used through screening, we can model the entities and relationships in the Twitter social network in the form of a heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. There are a number of heterogeneous mapping relationships involved, which can be expressed as follows:

$$\phi : \mathcal{V} \rightarrow \mathcal{A} = \{User | Tweet\} \tag{7}$$

$$\psi : \mathcal{E} \rightarrow \mathcal{R} = \{Follow | Friend | Create | Retweet | Like | Comment\} \tag{8}$$

Thus, a directed heterogeneous modeling graph $\mathcal{G}$ of the Twitter social network is obtained, where heterogeneous nodes representing different entities in a social network serve as the backbone, and the heterogeneous edges representing different connection relationships in the social network serve as links. To ensure high-quality modeling, more accurate representations were assigned to the features. Details of the heterogeneous nodes and edges are listed in Table 1.

Finally, to obtain better feature information based on tweet content, SybilFlyover also compares the similarity of the tweet text (for the sake of computing cost, we only do this for tweets that have interactive behavior with the user node). If the number and weight of the keywords overlap to a certain extent, we believe that the two tweets are similar, and a small weight edge is added between the corresponding tweet nodes in the generated social network graph as a supplement.

In this way, the discrete data pieces are sorted in this module according to their real logical relationships in the social network.

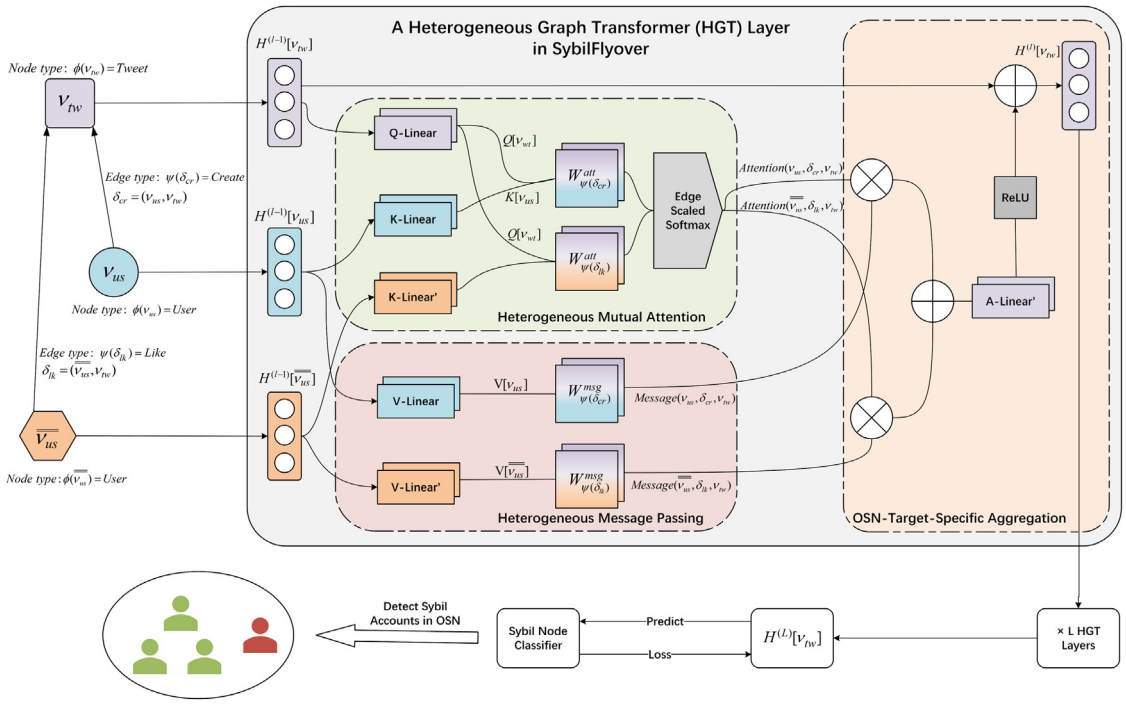

Fig. 4. Pipeline of Heterogeneous Graph Transformer in SybilFlyover.

Table 1
Detailed information about heterogeneous nodes and links.

Type	Element	Symbol	Main feature
Node	User	v_{us}	nick_name: text description: text follower_num: num
Node	Tweet	v_{tw}	content_text: text liked_num: num retweeted_num: num
Link	Follow	δ_{fl}	user_to_user: genres
Link	Friend	δ_{fd}	user_to_user: genres
Link	Friend	δ_{cr}	user_to_tweet: genres
Link	Retweet	δ_{rt}	user_to_tweet: genres
Link	Like	δ_{lk}	user_to_tweet: genres
Link	Comment	δ_{cm}	user_to_tweet: genres
Link	Similar	δ_{sm}	tweet_to_tweet: genres

A good balance between the data scale and modeling fineness level is achieved by introducing graph feature abundance λ, and the data pieces of all dimensions in the social network are organized into a directed heterogeneous modeling graph $\mathcal{G}$, thereby achieving the accurate modeling of complex relationships in the Twitter social network from the perspectives of user and content.

5.5. Graph modeling and account analysis

A heterogeneous graph enables us to accurately model the Twitter social network in a more fine-grained manner by determining a meaningful vector representation for the entity represented by each node. However, this property of a heterogeneous graph presents challenges for the processing of graph structure data. The model requires a more powerful data processing capability (which requires an extra design) to process the information of heterogeneous structures composed of various types of nodes and edges and the corresponding heterogeneous content information associated with each node [52] demonstrates that although heterogeneous graphs pose an additional challenge to graph processing models, the method of processing heterogeneous graphs is similar to that used to process homogeneous graphs [38]. The procedure for processing a heterogeneous graph can be divided into two stages: message passing and aggregation. Therefore, we employ the HGT framework [52] to process the heterogeneous OSN modeling graphs (as shown in Fig. 4) and add a heterogeneous mutual attention stage to introduce a self-attention mechanism [47] to the model, so as to obtain a better representation of heterogeneous graph data. For the lth layer in the HGT model, its operation on the representation graph $\mathcal{G}$ of a heterogeneous social network can be expressed as:

$$H^{l}[t] \leftarrow Aggregate_{\forall v_1 \in N(v_2), \forall \delta \in E(v_1,v_2)}\left(Att\left(v_1, v_2\right) \bullet Msg\left(v_1\right)\right) \quad (9)$$

To describe in detail how SybilFlyover processes a heterogeneous social network representation graph $\mathcal{G}$, we use the triple $\langle \phi\left(v_1\right), \psi\left(\delta\right), \phi\left(v_2\right)\rangle$ to define the meta relation of social network representation graph $\mathcal{G}$, i.e., the connection $\delta = (v_1, v_2)$ between two entity nodes v_1, v_2. To obtain the attention weight, we first use the zero vector to initialize the node embedding $\phi\left(v\right)^{emb}$ and the edge embedding $\psi\left(\delta\right)^{emb}$. Then, we use the node characteristics to initialize the attention matrices $W^{att}_{\phi(v_1)}$ and $W^{att}_{\phi(v_2)}$ of the original nodes and the target nodes, respectively, and finally use random values to initialize the matrices Key, Query, and Value (defined in the Transformer [47]) involved in the attention calculation, which are denoted as W^{K}, W^{Q} and W^{V}. In the first-layer HGT, the model is initialized in this way to obtain the ith K vector and Q vector of the target node v_2:

$$K^{i}\left(v_1\right) = K - Linear^{i}_{\phi(v_1)}\left(X^{v_2}_{\phi(v_2)} + \phi\left(v_2\right)^{emb} X^{v_2}_{\phi(v_2)} + \psi\left(\delta\right)^{emb}\right) \quad (10)$$

$$Q^{i}\left(v_2\right) = Q - Linear^{i}_{\phi(v_2)}\left(X^{v_2}_{\phi(v_2)} + \phi\left(v_2\right)^{emb}\right) \quad (11)$$

After the initialization is completed, the next step is to calculate the representation of the target node v_2. Similar to the way a transformer processes a string of words in a text, a set of projected attention weights of $h - head$ are calculated for each meta-relation of the hit node v_2 in the social network

Table 2
Statistical information of MIB dataset.

Dataset	Entity		Relationship		
	Account	Tweet	Follower	Friend	Total
TFP(@TheFakeProject)	469	563 693	258 494	24 110	500 204
E13(#elezioni2013)	1481	2 068 037	1 526 941	667 225	2 194 169
FSF(fastfollower)	1169	22 910	11 893	253 026	264 919
INT(intertwitter)	1337	58 925	23 173	517 485	540 658
TWT(twittertechnology)	845	114 192	28 588	729 839	758 427
HUM (human dataset)	1950	2 631 730	1 785 438	908 935	2 694 273
FAK (fake dataset)	3351	118 327	34 553	879 580	914 133
BAS (baseline dataset: $HUM \cup FAK$)	5301	2 750 057	1 819 991	17 885 151	3 608 506

representation graph $\mathcal{G}$.

$$Attention\left(v_1, \delta, v_2\right) = Softmax_{\forall v_1 \in N(v_2)}\left(\bigcup_{i=1}^{h} ATT\_head^i\left(v_1, \delta, v_2\right)\right) \tag{12}$$

With

$$ATT\_head^i\left(v_1, \delta, v_2\right) = \left(K^i\left(v_1\right) W_{\psi(\delta)}^{att} Q^i\left(v_2\right)^T\right) \bullet \frac{\left\langle \phi\left(v_1\right), \psi\left(\delta\right), \phi\left(v_2\right)\right\rangle}{\sqrt{d}} \tag{13}$$

$$K^i\left(v_1\right) = K - Linear_{\phi(v_1)}^i\left(H^{(l-1)}\left[v_1\right]\right) \tag{14}$$

$$Q^i\left(v_2\right) = Q - Linear_{\phi(v_2)}^i\left(H^{(l-1)}\left[v_2\right]\right) \tag{15}$$

where $N\left(v_2\right)$ is the function for obtaining all neighboring nodes of v_2; $\mu \in \mathbb{R}^{|\mathcal{A}|\times|\mathcal{R}|\times|\mathcal{A}|}$ is the tensor describing the meta-relation $\left\langle \phi\left(v_1\right), \psi\left(\delta\right), \phi\left(v_2\right)\right\rangle$ characteristics of the social network representation graph, which scales the attention weight adaptively.

In addition to obtaining the attention weights of meta-relations, it is necessary to complete the information transmission from the source nodes to the target nodes and add the meta relations of edges in this process to reduce the distribution difference between the heterogeneous nodes and edges. To accomplish this goal, we calculate the message transmission of the multi-head as follows [52]:

$$Message\left(v_1, \delta, v_2\right) = \bigcup_{i=1}^{h} MSG\_head^i\left(v_1, \delta, v_2\right) \tag{16}$$

With

$$MSG\_head^i\left(v_1, \delta, v_2\right) = M - Linear_{\phi(v_1)}^i\left(H^{(l-1)}\left[v_1\right]\right) W_{\psi(\delta)}^{msg} \tag{17}$$

where $M - Linear_{\phi(v_1)}^i : \mathbb{R}^d \rightarrow \mathbb{R}^{\frac{d}{h}}$ is the linear mapping function responsible for mapping the $\phi\left(v_1\right)$ type node v_1 to the ith message vector, and $W_{\psi(\delta)}^{msg} \in \mathbb{R}^{\frac{d^2}{h^2}}$ is a matrix constructed to introduce the heterogeneous edge differences.

After the attention weight and message transmission result of the meta-relations are obtained, the heterogeneous attention and messages are aggregated from the neighboring nodes, including the source node v_1 to the target node v_2:

$$\widetilde{H}^{(l)}\left[v_2\right] = \oplus_{\forall v_1 \in N(v_2)}\left(Attention\left(v_1, \delta, v_2\right), Message\left(v_1, \delta, v_2\right)\right) \tag{18}$$

where $\widetilde{H}^{(l)}\left[v_2\right]$ is the update vector obtained by the HGT layer that represents the meta-relation.

Finally, we applied a linear projection $A - Linear_{\phi(v_1)}$ to the updated vector $\widetilde{H}^{(l)}\left[v_1\right]$ followed by ReLU activation and residual connection. The vector of the target node v_2 is mapped back to its specified distribution to obtain the final output $H^{(l)}\left[v_2\right]$ of the lth layer of the HGT.

$$H^{(l)}\left[v_1\right] = A - Linear_{\phi(v_1)}\left(\sigma\left(\widetilde{H}^{(l)}\left[v_1\right]\right)\right) + H^{(l-1)}\left[v_2\right] \tag{19}$$

In this way, we use a combination of several layers of HGT to accomplish the convolution operation on the heterogeneous graph and obtain the representation of the social network graph $\mathcal{G}$. On this basis, the identification of Sybil nodes in graph $\mathcal{G}$ can be performed.

6. Experiments and evaluation

6.1. Dataset

We utilized the public dataset[1] from My Information Bubble (MIB) [53] to evaluate the performance of SybilFlyover in detecting Sybil accounts and to validate the effectiveness of the way the model is used. The relevant statistical figures for the MIB dataset are listed in Table 2.

The MIB dataset comprises five subdatasets: TFP, E13, FSF, INT, and TWT. The first two contain human accounts, whereas the remainder is composed of Sybil accounts. All accounts are real accounts in the Twitter social network obtained by researchers using webpage crawling. The MIB dataset contains rich features and good data balance. It is used as a test benchmark in many studies and is, therefore, highly representative. The MIB dataset was selected to ensure that the experimental results were useful for future research.

6.2. Baseline models

Eight representative models were selected as baseline models. These baseline models adopted different technical routes, and each could achieve state-of-the-art performance according to the technical route adopted.

The baseline models were as follows:

- Feature-based [21]: A feature-based fake account detection model is applied to a traditional machine learning algorithm that has already been integrated into the Chrome extension.
- SybilRank [9]: A state-of-the-art Random-Walk-based model.
- SmartWalk [12]: This model also adopts the random-walk-based method. Unlike SybilRank, which uses a fixed step length, SmartWalk can adaptively choose the step length of a random walk according to the initial node and path.
- SybilSAN [13]: A random-walk-based model using a two-layer hypergraph to model OSNs.
- SybilSCAR [16]: A state-of-the-art model based on loop-belief-propagation.
- GNN-based method [18]: A GNN-based model that uses low-pass filtering to process social network graphs and identify Sybil accounts.

[1] The dataset can be accessed at http://mib.projects.iit.cnr.it/dataset.html.

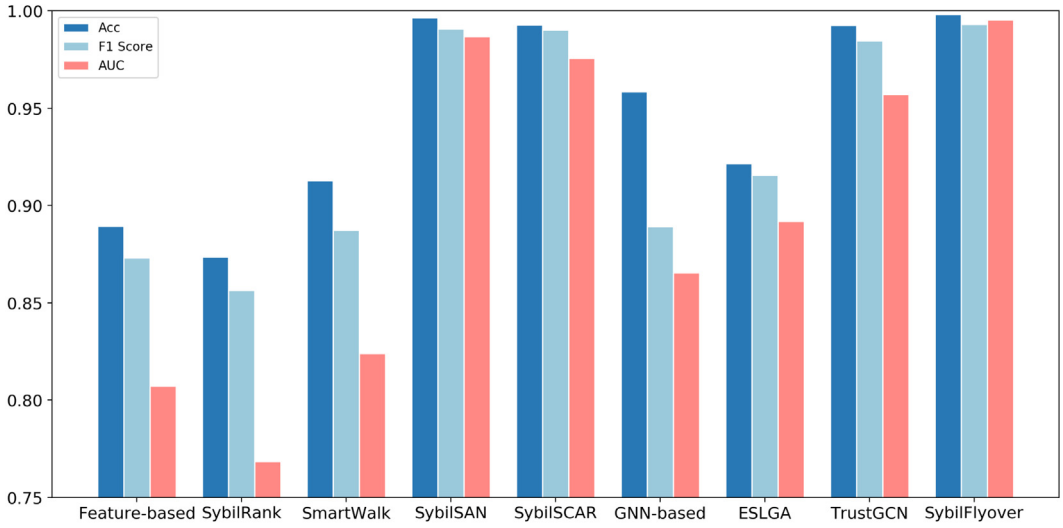

Fig. 5. Comparison with Baseline Models.

Table 3
Configuration of hyperparameters.

Hyperparameter	Value
HGT-layer num	2
Hidden-channel num	64
Out-channel num	6
Head num	2
Epoch	600
Batch size	32
Optimizer	Adam
Learning rate	0.002
Weight decay	0.001

- ESLGA [20]: A graph-based semi-supervised learning algorithm to detect fake users.
- TrustGCN [19]: A state-of-the-art GCN-based method.

We compared the performance levels of SybilFlyover and the five baseline models in detecting Sybil accounts.

6.3. Experimental results

We evaluated our SybilFlyover [54] and the eight selected baseline models on the MIB dataset [53] for authentic Twitter accounts. We evaluated the performance of the models in identifying Sybil accounts using the metrics of accuracy, F1-Score, and area under the receiver operating characteristic curve (AUC). Viswanath et al. [55] proved that the effectiveness of Sybil account detection could be evaluated using the AUC value. A high AUC value indicates that when a positive sample (i.e., Sybil account) and a negative sample (i.e., normal account) are randomly selected, the confidence probability of the positive sample being a Sybil account given by the model classifier is higher than that given to the negative sample. For each baseline model, the experiment used the parameter settings specified in the corresponding research paper. The experiments were performed on a workstation with an Intel Core i5-10600KF, 32 GB RAM, and dual NVIDIA GeForce 1080Ti with 11 GB graphic memory. The configuration of the hyperparameters used when training SybilFlyover is presented in Table 3. In the experiment, 80% of the data was specified for training, 10% for verification, and 10% for testing. To reduce the influence of randomness on the experimental results, we conducted five rounds of testing and took the average value of each evaluation metric as the final result.

Fig. 5 compares the performance levels of SybilFlyover and baseline models. SybilSAN has the best performance among traditional structure-based models. Among the GNN-based methods, the GCN-based method is superior to the naive-GNN-based method and the ESLGA. SybilFlyover outperforms feature-based models, traditional structure-based models, and GNN-based models in terms of Sybil detection, with an accuracy of 0.9981, F1-Score of 0.9930, and AUC value of 0.9954. In particular, the AUC value of SybilFlyover was larger than those of the other methods, indicating that SybilFlyover could achieve better performance in Sybil classification tasks than the other methods.

6.4. Ablation experiment

We conducted an ablation experiment to validate the efficacy of the methods used in SybilFlyover. As shown in Table 4, we designed the following tests: to validate the effectiveness of the tweet content enhancement method based on the heterogeneous graph, validate the effectiveness of the graph processing method based on self-attention, verify the effectiveness of the text representation method based on prompt learning, and verify the effectiveness of the heterogeneous graph representation learning model we applied.

In the ablation test to determine the effectiveness of the Twitter content enhancement method based on a heterogeneous graph, the comparison model SybilFlyover-Homo-Att processed a homogeneous social network graph containing only user information using a method based on self-attention. SybilFlyover also uses a self-attention-based method to process heterogeneous social network graphs containing user and tweet content information. The experimental results show that the ability of the model to identify Sybil accounts can be effectively enhanced by introducing tweet-text information into the model using a method based on a heterogeneous graph.

In an ablation test to determine the effectiveness of the graph processing method based on self-attention, the comparative model SybilFlyover-Homo processed social network graphs using a GCN without an attention mechanism. In comparison, the SybilFlyover-Homo-Att model uses a method based on the self-attention mechanism to calculate the social network graph, similar to the proposed SybilFlyover model. The experimental results indicate that the detection performance of the model can be effectively improved by introducing a self-attention mechanism.

Table 4
Design of ablation tests.

Model	Node type	Graph modeling approach	Text represent approach
SybilFlyover-Homo	User	GCN	N/A
SybilFlyover-Homo-Att	User	GAT	N/A
SybilFlyover-MLM	User, Tweet	HGT	BERT
SybilFlyover-HAN	User, Tweet	HAN	PromptBERT
SybilFlyover	User, Tweet	HGT	PromptBERT

Table 5
Result of ablation experiments.

Model	Accuracy	F1-Score	AUC
SybilFlyover-Homo	0.9536	0.9511	0.9329
SybilFlyover-Homo-Att	0.9676	0.9693	0.9547
SybilFlyover-MLM	0.9908	0.9879	0.9881
SybilFlyover-HAN	0.9926	0.9863	0.9885
SybilFlyover	**0.9981**	**0.9930**	**0.9954**

In the ablation test to verify the effectiveness of the text representation method based on prompt learning, the SybilFlyover-MLM comparison model obtained the text representation of tweets using the traditional BERT model. In comparison, SybilFlyover used a method based on prompt learning; that is, PromptBERT was used to obtain the text representation of tweets. The experimental results show that using a method based on prompt learning to obtain the text representation of tweets can introduce content-related information into the model more effectively and provide a better basis for the model to identify Sybil accounts.

In an ablation test to determine the effectiveness of the heterogeneous graph representation learning method used in SybilFlyover (i.e., HGT [52]), the comparative model SybilFlyover-HAN applied HAN [56] as graph representation learning model, which is also a heterogeneous graph representation learning model with an attention mechanism. Compared with HGT, HAN does not adopt the Transformer structure when processing graph embedding but adopts the traditional attention method. The HGT improves the message passing and aggregation algorithm applied in the model [57], which can also be reflected in the results of the ablation experiments (see Table 5). From the results of the above four ablation tests, it can be concluded that the main methods used in SybilFlyover are effective.

6.5. Microbenchmarking

In the design of SybilFlyover, we introduced graph feature abundance λ to control the problem of excessive graph features caused by modeling an OSN using a heterogeneous graph, resulting in the rapid growth of computing. Therefore, in this section, we describe experiments to discuss the choice of λ. In the experiment, we selected different λ values, injected different scales of *gain_features* into the model, and observed the detection performance of the model for Sybil accounts and changes in the corresponding training time. As shown in Fig. 6, we found that with an increase in λ, the Sybil account recognition ability of the model improved owing to more abundant features, and the marginal benefit decreased, followed by a slowdown in the improvement of Sybil account recognition ability and a large increase in model training time. Therefore, in this study, we set λ to 0.86.

6.6. Scalable experiment

We have introduced a large amount of information into SybilFlyover through a heterogeneous graph, working to better model

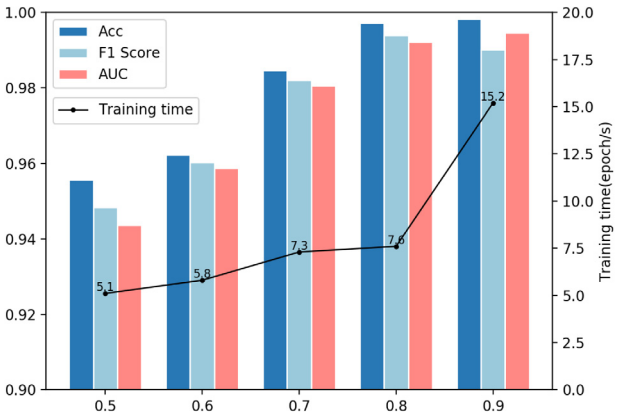

Fig. 6. Impact of Graph feature abundance λ.

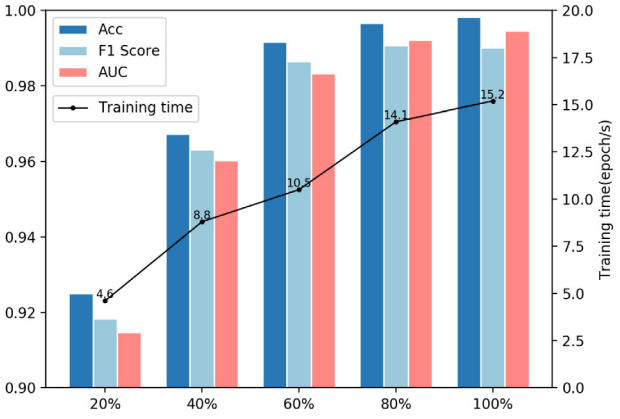

Fig. 7. Scalable experiment.

OSNs and hidden Sybil accounts. Although we balanced the increase in computational effort by introducing the graph feature abundance λ, we still need experiments to verify the scalability of SybilFlyover to ensure the applicability of the model in the face of larger data. For scalable experimentation, we randomly sampled 20%, 40%, 60%, 80%, and 100% of the dataset and kept the value of λ at 0.86, to observe the changes in the model's ability to detect Sybil accounts and, more importantly, the increase in the training time of the model. As shown in Fig. 7, we observed that the training time of the model increased linearly with the size of the data, and that the model performed poorly in identifying Sybil accounts when the amount of training data was small, but it could recover quickly with an increase in the amount of training data.

7. Conclusion

This study proposes a practical Sybil detection model called SybilFlyover. The proposed model can accurately model the entities and complex relationships in an OSN using a heterogeneous graph, obtain the features of Twitter text information using a method based on prompt learning, and introduce additional modeling information besides the traditional structural information into the social network graph to provide a more reliable basis for the model to identify Sybil accounts. By adopting the HGT framework, graph feature abundance was introduced into the model to attain a good balance between the rich information of the social network graph and the massive computing workload in processing the heterogeneous graph, which can accurately identify Sybil accounts from the heterogeneous social network

graph. In an experiment on a publicly available real dataset, SybilFlyover outperformed the most advanced Sybil detection models currently available for Sybil detection tasks, thus proving its effectiveness in real-world applications.

In the future, the application of successful experiences in other tasks to SybilFlyover could be analyzed. For instance, a multi-layer comparative learning scheme can be used to further improve the representation learning effect of a social network graph [58]. In addition, how to exploit mutual affinities between nodes and attributes better when realizing graph embedding could be explored [59]. The spatial-social index [60] could be introduced to the proposed model to optimize SybilFlyover so that it can handle web-scale data. Investigating the possibility of migrating the proposed model to scenarios with more fine-grained tasks, such as detecting fake profiles, fake identities, or troll accounts, could be an important follow-up study.

CRediT authorship contribution statement

Siyu Li: Conceptualization, Methodology, Software, Validation, Writing – original draft. **Jin Yang:** Supervision, Funding acquisition. **Gang Liang:** Formal analysis. **Tianrui Li:** Writing – review & editing. **Kui Zhao:** Resources, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The code is available in https://doi.org/10.24433/CO.9860846.v1 The public datasets we used could be accessed in http://mib.projects.iit.cnr.it/dataset.html.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61872254, No. 62162057), the Key Lab of Information Network Security of Ministry of Public Security, China (C20606), and the Sichuan Science and Technology Program, China (2021JDRC0004). And we want to convey our grateful appreciation to the corresponding author of this paper, Jin Yang. He has offered advice with huge values in all stages when writing this paper to us. We also would like to thank My Information Bubble (MIB) for sharing their datasets for evaluation. And we would like to thank CodeOcean for providing a reproducibility check of this work.

References

[1] J.N. Matthews, B.R. Voter, B. Hudson, J.S. White, S. Gurajala, Profile characteristics of fake twitter accounts, Big Data Soc. 3 (2) (2016).
[2] E. Van Der Walt, J. Eloff, Using machine learning to detect fake identities: bots vs humans, IEEE Access 6 (2018) 6540–6549.
[3] M. Latah, Detection of malicious social bots: A survey and a refined taxonomy, Expert Syst. Appl. 151 (2020) 113383.
[4] O. Varol, E. Ferrara, C. Davis, F. Menczer, A. Flammini, Online human-bot interactions: Detection, estimation, and characterization, in: Proceedings of the International AAAI Conference on Web and Social Media, Vol. 11, 2017, pp. 280–289.
[5] S. Rao, A.K. Verma, T. Bhatia, A review on social spam detection: Challenges, open issues, and future directions, Expert Syst. Appl. 186 (2021) 115742.
[6] Z. Kleinman, Elon musk puts Twitter deal on hold over fake account details, 2022, online, https://www.bbc.com/news/business-61433724.
[7] H. Yu, M. Kaminsky, P.B. Gibbons, A. Flaxman, Sybilguard: defending against sybil attacks via social networks, in: Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM, 2006, pp. 267–278.
[8] H. Yu, P.B. Gibbons, M. Kaminsky, F. Xiao, Sybillimit: A near-optimal social network defense against sybil attacks, in: 2008 IEEE Symposium on Security and Privacy, S&P, IEEE, 2008, pp. 3–17.
[9] Q. Cao, M. Sirivianos, X. Yang, T. Pregueiro, Aiding the detection of fake accounts in large scale social online services, in: 9th USENIX Symposium on Networked Systems Design and Implementation, NSDI 12, 2012, pp. 197–210.
[10] C. Yang, R. Harkreader, J. Zhang, S. Shin, G. Gu, 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, 2012, pp. 71–80.
[11] Y. Boshmaf, D. Logothetis, G. Siganos, J. Lería, J. Lorenzo, M. Ripeanu, K. Beznosov, H. Halawa, Íntegro: Leveraging victim prediction for robust fake account detection in large scale OSNs, Comput. Secur. 61 (2016) 142–168.
[12] Y. Liu, S. Ji, P. Mittal, Smartwalk: Enhancing social network security via adaptive random walks, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 492–503.
[13] X. Zhang, H. Xie, P. Yi, J.C. Lui, Enhancing Sybil detection via social-activity networks: A random walk approach, IEEE Trans. Dependable Secure Comput. (2022).
[14] N.Z. Gong, M. Frank, P. Mittal, Sybilbelief: A semi-supervised learning approach for structure-based sybil detection, IEEE Trans. Inf. Forensics Secur. 9 (6) (2014) 976–987.
[15] B. Wang, N.Z. Gong, H. Fu, GANG: Detecting fraudulent users in online social networks via guilt-by-association on directed graphs, in: 2017 IEEE International Conference on Data Mining, ICDM, IEEE, 2017, pp. 465–474.
[16] B. Wang, L. Zhang, N.Z. Gong, SybilSCAR: Sybil detection in online social networks via local rule based propagation, in: IEEE INFOCOM 2017-IEEE Conference on Computer Communications, IEEE, 2017, pp. 1–9.
[17] B. Wang, J. Jia, L. Zhang, N.Z. Gong, Structure-based sybil detection in social networks via local rule-based propagation, IEEE Trans. Netw. Sci. Eng. 6 (3) (2018) 523–537.
[18] S. Furutani, T. Shibahara, K. Hato, M. Akiyama, M. Aida, Sybil detection as graph filtering, in: GLOBECOM 2020-2020 IEEE Global Communications Conference, IEEE, 2020, pp. 1–6.
[19] Y. Sun, Z. Yang, Y. Dai, TrustGCN: enabling graph convolutional network for robust sybil detection in OSNs, in: 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM, IEEE, 2020, pp. 1–7.
[20] M. Balaanand, N. Karthikeyan, S. Karthik, R. Varatharajan, G. Manogaran, C. Sivaparthipan, An enhanced graph-based semi-supervised learning algorithm to detect fake users on Twitter, J. Supercomput. 75 (9) (2019) 6085–6105.
[21] S.R. Sahoo, B. Gupta, Real-time detection of fake account in twitter using machine-learning approach, in: Advances in Computational Intelligence and Communication Technology, Springer, 2021, pp. 149–159.
[22] Z. Chu, S. Gianvecchio, H. Wang, S. Jajodia, Detecting automation of twitter accounts: Are you a human, bot, or cyborg? IEEE Trans. Dependable Secure Comput. 9 (6) (2012) 811–824.
[23] Z. Yang, C. Wilson, X. Wang, T. Gao, B.Y. Zhao, Y. Dai, Uncovering social network sybils in the wild, ACM Trans. Knowl. Discov. Data (TKDD) 8 (1) (2014) 1–29.
[24] Z. Miller, B. Dickinson, W. Deitrick, W. Hu, A.H. Wang, Twitter spammer detection using data stream clustering, Inform. Sci. 260 (2014) 64–73.
[25] C. Xiao, D.M. Freeman, T. Hwa, Detecting clusters of fake accounts in online social networks, in: Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security, 2015, pp. 91–101.
[26] S. Kudugunta, E. Ferrara, Deep neural networks for bot detection, Inform. Sci. 467 (2018) 312–322.
[27] Y. Wu, Y. Fang, S. Shang, J. Jin, L. Wei, H. Wang, A novel framework for detecting social bots with deep neural networks and active learning, Knowl.-Based Syst. 211 (2021) 106525.
[28] Y. Feng, J. Li, L. Jiao, X. Wu, Towards learning-based, content-agnostic detection of social bot traffic, IEEE Trans. Dependable Secure Comput. 18 (5) (2020) 2149–2163.
[29] P. Wanda, H.J. Jie, DeepProfile: Finding fake profile in online social network using dynamic CNN, J. Inf. Secur. Appl. 52 (2020) 102465.
[30] P. Wanda, H.J. Jie, Deepfriend: finding abnormal nodes in online social networks using dynamic deep learning, Soc. Netw. Anal. Min. 11 (1) (2021) 1–12.
[31] T. Pourhabibi, K.-L. Ong, B.H. Kam, Y.L. Boo, Fraud detection: A systematic literature review of graph-based anomaly detection approaches, Decis. Support Syst. 133 (2020) 113303.
[32] D. Yuan, Y. Miao, N.Z. Gong, Z. Yang, Q. Li, D. Song, Q. Wang, X. Liang, Detecting fake accounts in online social networks at the time of registrations, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 1423–1438.

[33] A. Breuer, R. Eilat, U. Weinsberg, Friend or faux: graph-based early detection of fake accounts on social networks, in: Proceedings of the Web Conference 2020, 2020, pp. 1287–1297.
[34] X. Liang, Z. Yang, B. Wang, S. Hu, Z. Yang, D. Yuan, N.Z. Gong, Q. Li, F. He, Unveiling fake accounts at the time of registration: An unsupervised approach, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 3240–3250.
[35] N. Guan, D. Song, L. Liao, Knowledge graph embedding with concepts, Knowl.-Based Syst. 164 (2019) 38–44.
[36] X. Chen, S. Jia, Y. Xiang, A review: Knowledge reasoning over knowledge graph, Expert Syst. Appl. 141 (2020) 112948.
[37] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: Proceedings of ICLR, 2017, pp. 1–14.
[38] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, Adv. Neural Inf. Process. Syst. 30 (2017).
[39] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, in: Proceedings of ICLR, 2018, pp. 1–12.
[40] X. Wang, D. Bo, C. Shi, S. Fan, Y. Ye, S.Y. Philip, A survey on heterogeneous graph embedding: methods, techniques, applications and sources, IEEE Trans. Big Data (2022).
[41] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, P.S. Yu, Heterogeneous graph attention network, in: The World Wide Web Conference, 2019, pp. 2022–2032.
[42] S. Yoon, J.-H. Cho, D.S. Kim, T.J. Moore, F. Free-Nelson, H. Lim, Attack graph-based moving target defense in software-defined networks, IEEE Trans. Netw. Serv. Manag. 17 (3) (2020) 1653–1668.
[43] Y. Hei, R. Yang, H. Peng, L. Wang, X. Xu, J. Liu, H. Liu, J. Xu, L. Sun, Hawk: Rapid android malware detection through heterogeneous graph attention networks, IEEE Trans. Neural Netw. Learn. Syst. (2021).
[44] H. Yin, S. Yang, X. Song, W. Liu, J. Li, Deep fusion of multimodal features for social media retweet time prediction, World Wide Web 24 (4) (2021) 1027–1044.
[45] W. Ling, C. Dyer, A.W. Black, I. Trancoso, Two/too simple adaptations of word2vec for syntax problems, in: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, 2015, pp. 1299–1304.
[46] J.D.M.-W.C. Kenton, L.K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, 2019, pp. 4171–4186.
[47] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Proc. Syst. (NIPS) 30 (2017).
[48] B. Li, H. Zhou, J. He, M. Wang, Y. Yang, L. Li, On the sentence embeddings from bert for semantic textual similarity, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2020, pp. 9119–9130.
[49] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Adv. Neural Inf. Process. Syst. 33 (2020) 1877–1901.
[50] T. Jiang, S. Huang, Z. Zhang, D. Wang, F. Zhuang, F. Wei, H. Huang, L. Zhang, Q. Zhang, PromptBERT: Improving BERT sentence embeddings with prompts, 2022, arXiv preprint arXiv:2201.04337.
[51] L. Peel, Data driven prognostics using a Kalman filter ensemble of neural network models, in: 2008 International Conference on Prognostics and Health Management, IEEE, 2008, pp. 1–6.
[52] Z. Hu, Y. Dong, K. Wang, Y. Sun, Heterogeneous graph transformer, in: Proceedings of the Web Conference 2020, 2020, pp. 2704–2710.
[53] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, M. Tesconi, Fame for sale: Efficient detection of fake Twitter followers, Decis. Support Syst. 80 (2015) 56–71.
[54] S. Li, SybilFlyover: Heterogeneous graph-based fake account detection model on social networks, Knowl.-Based Syst. (2022) http://dx.doi.org/10.24433/CO.9860846.v1, https://www.codeocean.com/.
[55] B. Viswanath, A. Post, K.P. Gummadi, A. Mislove, An analysis of social network-based sybil defenses, ACM SIGCOMM Comput. Commun. Rev. 40 (4) (2010) 363–374.
[56] C. Zhang, D. Song, C. Huang, A. Swami, N.V. Chawla, Heterogeneous graph neural network, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 793–803.
[57] N. Chairatanakul, X. Liu, N.T. Hoang, T. Murata, Heterogeneous graph embedding with single-level aggregation and infomax encoding, Mach. Learn. (2022) 1–30.
[58] X. Song, J. Li, Q. Lei, W. Zhao, Y. Chen, A. Mian, Bi-CLKT: Bi-graph contrastive learning based knowledge tracing, Knowl.-Based Syst. 241 (2022) 108274.
[59] S. Yang, S. Verma, B. Cai, J. Jiang, K. Yu, F. Chen, S. Yu, Variational co-embedding learning for attributed network clustering, 2021, arXiv preprint arXiv:2104.07295.
[60] T. Cai, J. Li, A.S. Mian, T. Sellis, J.X. Yu, et al., Target-aware holistic influence maximization in spatial social networks, IEEE Trans. Knowl. Data Eng. (2020).