Uncovering Malicious Accounts in Open Mobile Social Networks Using a Graph- and Text-Based Attention Fusion Algorithm

Yuting Tang, Dafang Zhang, Wei Liang, *Senior Member, IEEE*, Kuan-Ching Li, *Senior Member, IEEE*, and Keqin Li, *Fellow, IEEE*

***Abstract*—In recent years, open mobile social networks focused on socializing and dating purposes have gained widespread popularity, such as Soul, Tinder, Momo, and Tantan, among several others. These applications permit users to post, comment, and send private messages to other users without their consent, making communication accessible. However, this low-entry communication approach has also increased malicious user attacks. We delve into a comprehensive analysis of malicious accounts in open socializing and dating applications, revealing that the existing methods overlook hidden malicious signals within the user text-related information, thus resulting in poor detection performance. For such, we propose GraphTAM, a novel graph- and text-based multihead attention fusion network model for detecting such malicious accounts, consisting of modules that effectively combine nontext-related and text-related information, enhancing the accuracy and performance of detecting malicious accounts. We employ graph convolutional networks (GCNs) for nontext-related information to extract advanced representations of users, incorporating their attribute and social relationship features. Regarding text-related information, we employ a multihead attention model to identify suspicious patterns in users' posted articles, comments, and relevant behavioral statistics, so finally, we merge the advanced representations of nontext-related and text-related information using a multilayer perceptron to determine the maliciousness of an account. Data sets collected from SLink are utilized for the experimental evaluation and to compare the performance of the proposed model with the several state of the art algorithms. Experimental results show significant advantages in malicious account detection, where the F1 score achieves over 0.9, outperforming the existing methods that range between 0.6 and 0.85. Furthermore, the comparative experiments substantiate the critical role of text-related information in detecting malicious accounts in open socializing and dating applications.**

***Index Terms*—Graph convolutional networks (GCNs), malicious account detection, multihead attention fusion network, open mobile social networks.**

Manuscript received 20 February 2024; revised 15 May 2024; accepted 11 June 2024. Date of publication 24 June 2024; date of current version 25 September 2024. This work was supported by the National Natural Science Foundation of China under Grant 62072170 and Grant 61872138. *(Corresponding authors: Dafang Zhang; Wei Liang; Kuan-Ching Li.)*

Yuting Tang and Dafang Zhang are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410006, China (e-mail: yuting_tang@hnu.edu.cn; dfzhang@hnu.edu.cn).

Wei Liang is with the College of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201, China (e-mail: wliang@hnust.edu.cn).

Kuan-Ching Li is with the Department of Computer Science and Information Engineering, Providence University, Taichung 43301, Taiwan (e-mail: kuancli@pu.edu.tw).

Keqin Li is with the Department of Computer, State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

Digital Object Identifier 10.1109/JIOT.2024.3416556

I. Introduction

With the advancement of communication and software technologies, there are more and more types of online social networks, specifically mobile social networks for dating, such as Soul, Tinder, Momo [1], Tantan, and several others. Through available open software, making friends online has become a new trend for young people in recent years, attracting the attention of hundreds of millions of users. Privacy protection is also highly valued in various fields, such as mobile social media, IoT privacy security protection, and data privacy protection in MEC [2], [3]. However, these applications are different from the privacy-centric mobile social networks, as the users can communicate with each other without first becoming friends, significantly reducing the cost of establishing connections between the users. Therefore, they have become a target for malicious user attacks. Like conventional social media platforms, including Facebook, Google+, and Instagram, these open dating apps enable users to share posts, engage in discussions, and build social connections. The difference is that these apps can send messages and establish relationships without the other party's permission. Due to this low communication threshold, these dating apps have become a popular target for network attacks. For example, by creating fake malicious accounts, attackers can spread false junk information, post malicious advertisements [4], manipulate online bidding results, or organize prostitution. These malicious behaviors bring horrible experiences to the legitimate users, seriously threatening their property and personal safety. Therefore, it is urgent to research detecting malicious accounts on such dating apps to improve the user experience and reduce crime rates.

Researchers have conducted extensive investigations to detect malicious accounts on the typical social networks. This kind of work can be roughly divided into three types, all of which aim to defend against network attacks by directly identifying fake or malicious accounts controlled by attackers. The first approach employs machine learning techniques to construct a classifier [5] using the users' fundamental attribute features for legitimacy detection. The second approach entails

constructing a social connection graph [6] to assess legitimacy. The third method involves clustering user behaviors to discern patterns between the malicious and legitimate users [7], followed by making judgments based on the analysis. Unfortunately, these conventional methods are ineffective in detecting malicious accounts on popular open dating apps. The reasons are as follows.

1) For users with relatively loose social relationships in such available dating apps, using the graph-based detection methods alone is challenging to be effective.
2) Some advanced attackers are good at disguising themselves as legitimate users, not only by disguising their personal attribute characteristics but also by imitating legitimate users' behavior habits and patterns to achieve their illegal purposes. Therefore, machine learning methods based on user attribute characteristics or behavior also have limitations.

Through observation and analysis, we found that a prevalent attack method used by attackers is to publish malicious comments, false junk posts, or malicious activity advertisements, all of which are malicious activities conducted through the text information. Previous research has not undertaken a text-level analysis of such open dating apps. Based on the summary of prior work and the analysis of existing work challenges, our approach suggests a graph- and text-based multihead attention fusion network model for identifying malicious accounts on these open dating apps. The model mainly consists of a text-related information processing module, a nontext-related information processing module, and a multihead attention fusion module. The model uses graph-based and text-based features to improve the accuracy of malicious account detection, which is more effective than the traditional methods.

We utilized a relevant data set collected on SLink to compare the performance of our model against the several state of the art algorithms. Our model achieved an F1 score above 0.9, while the existing methods scored between 0.6 and 0.85. F1 score is a comprehensive metric for evaluating the models, and its specific description will be detailed in the experimental section. Furthermore, through the comparative experiments, we demonstrated that the contribution of textual information to the accuracy of malicious detection in open social networking applications is crucial.

Our key contributions are summarized as follows.

1) This investigation presents a pioneering effort to identify malicious accounts by analysing the textual data and social relationships within open social networking platforms.
2) The proposed approach suggests a graph- and text-based multihead attention fusion network model to analyse users' text and nontext related information. Specifically, we use graph convolutional network (GCN) to obtain advanced representations of users containing social relationships and their feature attributes, where we also use the multihead attention module to detect suspicious patterns in user behavior statistics and text information.
3) To conduct experimental analysis on a collected real data set to evaluate our model. Experimental outcomes demonstrate the superiority of our model over the state of the art algorithms, affirming its potential as a potent tool to detect malicious accounts in this software.
4) Through the comparative experiments, we demonstrate that the text-related information is pivotal in identifying malicious accounts in open social networking software.

The remainder of this work is organized as follows. We introduce the related work in Section II, relevant research background of this article in Section III, the notation, problem definition, and specific algorithm details in Section IV, and the experimental results, analysis, and discussions are depicted in Section V, and finally, the concluding remarks and future work are given in Section VI.

II. Related Work

In this section, we will briefly introduce the previous research on the traditional OSNs and open social networking applications and then discuss the earlier methods for detecting malicious accounts.

A. Research on OSN

Traditional OSNs, such as Facebook and Instagram have been studied extensively by scholars due to their early appearance and large user base. As early as 1998, Watts and Strogatz [8] proposed the "small world" network model, revealing the two characteristics of "high clustering" and "short path length" that most real social networks have. Leskovec et al. [9] studied the evolution of social networks, such as Flickr and LiveJournal, and found that the degree distribution of nodes in social networks follows a power-law distribution, and proposed two evolution models, the "rich-get-richer" and "network self-growth." Weng et al. [10] studied information diffusion on Twitter and found that the diffusion of information exhibits two modes: 1) "bursty" and 2) "crash," and a small number of nodes in the network contribute the majority of the diffusion. Fortunato and Barthélemy [11] proposed the concept of "optimal community structure in complex networks" and provided a more efficient method for community discovery. Suhara et al. [12] studied emotional propagation on Facebook and found that the emotional propagation in social networks exhibits two modes: 1) "emotional synchronization" and 2) "emotional bias."

For open social networking applications, such as Skout [13] and Momo [10] which have appeared and become popular rapidly, researchers have conducted in-depth research in this field because they have accumulated a large number of users and become new targets for attackers. Peng et al. [14] found through research that the men's tendency to make friends is with young women, while women consider the educational background and income of their potential friends. Zytko et al. [15] studied social networking applications from an unique perspective and found that people do not want to deceive their online friends because they are afraid of meeting them in real life. These studies have provided some inspiration for this article.

B. Malicious Account Detection Research

We categorize the previous research on malicious account detection into three categories: 1) feature-based; 2) graph-based; and 3) user-aggregated behavior-based methods.

Feature-based methods [7], [16] extract features from an user's personal profile, social graph, and behavior [17], then use machine learning techniques to identify malicious accounts. Most methods [7], [18] use supervised machine learning techniques, where a classifier is trained using prelabeled malicious and benign accounts and then used to classify the remaining accounts. For example, the Facebook's immunity system provides system support to manage many Facebook attack classifiers [4]. Wang and Xu [5] used LDA to obtain the topic distribution of textual data, combined with descriptive features to judge the maliciousness of an account. Nivas et al. [19] classified account types by integrating advanced feature selection and dimensionality reduction techniques. Khan et al. [20] used decision tree classifiers and oriented gradient histograms for the feature extraction and representation based on various user attribute information and then train on a deep convolutional network to detect the presence of anomalies. We believe that these methods are not applicable to the detection of malicious accounts in currently popular open social networking software because malicious accounts in these applications can easily imitate legitimate user features, such as personal profiles and social relationships to cover their tracks, which can reduce the effectiveness of these methods.

Graph-based methods [6] model OSNs as graphs with users as nodes and edges representing features that describe social relationships between the users (*e.g.,* follow and interact). The goal is to judge the maliciousness of an account by analysing the structural differences between the legitimate and malicious users in the established social graph. These methods mainly use random walk [21], community detection [22], or loop belief propagation (LBP) [23]. For example, Cao et al. [24] used the random walk to detect malicious accounts on Tuenti. Gong et al. [25] proposed a semi-supervised learning framework, SybiBelief, which uses LBP to detect malicious accounts. Recently, Wang et al. [23] proposed a collective classification framework based on the learning edge weights, achieving higher accuracy than the previous methods. Wanda and Jie [26] classified malicious vertices using link information of nodes because open social networking software has loose connectivity, these graph-based methods cannot achieve good results.

User-aggregated behavior-based methods mainly use the clustering methods to distinguish between the legitimate and malicious accounts. For example, Clickstream [27] and CopyCatch [28] pioneered clustering work on user-aggregated behavior in online social networks. Clickstream analysis identifies paired resemblances in HTTP requests made by social network accounts and categorizes accounts exhibiting comparable patterns. Leveraging the labeled data, the method distinguishes clusters as fake or legitimate. Clusters above the threshold of prelabeled fake accounts are classified as fake and below it as legitimate. However, its limitation is that it cannot be deployed on large OSNs like Facebook. CopyCatch is an internal system of Facebook that can detect fake synchronous likes. Its limitation is that the algorithm complexity increases exponentially with user behavior. The most representative work on the traditional OSNs is SynchroTrap [7], which borrows from the previous work on the zombie network detection and solves the limitations of the above two methods by identifying malicious accounts with synchronous behavior through hierarchical clustering. It has been successfully deployed on Facebook and has shown excellent performance. Research on these representative algorithms mainly focuses on the traditional OSNs. Due to the differences in functionality and design between the Web OSNs and mobile OSNs, the traditional OSN methods may not be effective for open mobile OSNs. These methods may not be able to detect malicious accounts that have limited behavioral patterns and are very similar to legitimate accounts, while some malicious accounts in open dating applications do not exhibit particularly frequent attack behaviors. Some other machine learning algorithms have also given us great inspiration, such as [29], while widely used nowadays in various fields [30], [31].

Two works closely related to this research are those proposed by Suarez-Tangil et al. [32], and He et al. [1]. The former uses profile features (PFs) to detect malicious users, as they first use statistical information, images, and other features to generate diverse characteristics from the user profiles, so then they use a support vector machine (SVM) as a classifier to identify and classify accounts. They did not consider text information related to users, but our experiments have shown that the text information is critical in detecting malicious accounts in open dating applications. On the other hand, the latter used statistical information from the user profiles and text-related information to detect malicious accounts in dating apps. Still, they did not consider the relationship features (RFs) among users, which is an essential factor reflecting user types in open dating apps.

Unlike the models mentioned above, our model not only considers users' textual information but also considers the relational information among users. By separately processing and integrating textual and nontextual information, we extract advanced representations of user features, enabling more accurate determination of user legitimacy. Specifically, for nontextual information, we employ GCNs to obtain advanced representations of accounts, including descriptive features and social RFs. For textual information, we utilize a multi-head attention model to capture information, such as posts, comments, and related behavioral statistics, each with varying attention weights reflecting potential suspicious patterns. Finally, the advanced representations of nontextual and textual information are merged, and a multilayer perceptron (MLP) is employed to determine account legitimacy. Compared to the previous methods, our model more comprehensively considers user-related information.

III. Background

A. Basic Features of Open Social Networking Apps for Dating

In this section, we use SLink as an example to describe the key design and basic functions of open social software

with the purpose of making friends. Dating apps, such as Soul, Tinder, Momo, and Tantan have similar functions and attributes, partially or entirely. When logging into SLink, the users must register a new account and create a personal profile. Users can upload videos, audio, and images, and publish posts, all of which can be tagged with location and found on the user's personal profile page. These profiles are public to all the users and can be viewed without establishing a friend relationship.

Users can also send messages to other users through the "match" feature in SLink. For example, the users can use the location-based matching function to find other users in the same city and send them messages to make friends. However, this location-based matching feature requires payment. We will first describe these unique designs in detail. Next, we will provide a concise overview of the traits exhibited by malicious accounts in dating apps with this feature.

B. Unrestricted Content Sharing

SLink's content-sharing policy is open, similar to Facebook and Weibo. For example, a post can be published on SLink as "@Tony, this place is fascinating!," which will notify the user Tony to see the post, even if Tony is not a follower or followee of the user. At the same time, other users can see the post on the plaza and make indiscriminate comments, even if the commenter is not a follower or followee of the user who posted the content. On SLink, when an user visits another user's homepage, they can see all the posts published on the public timeline, as well as all the comments on the posts, and can comment on these posts. In SLink, users can also send each other any message without any restrictions, such as text, images, audio, video, etc.

C. Malicious Account Activities in SLink

The open content sharing and pairing modes of social networking applications have enriched the channels for the user interaction but have also brought potential threats to the legitimate users. Through the statistical analysis of malicious accounts in our data set, we found that malicious attacks on open social networking applications can take two primary forms: location-based attacks account for 33.5%, content-based attacks account for 60.4%, and other attacks account for 6.1%. We will briefly describe these attacks using examples.

Location-Based Attacks: Malicious users can use location-based services to falsely check in at a particular location using a virtual IP address to achieve self-promotion or commercial interests [16]. For example, malicious users can create a fake location to attract benign users' attention or impersonate a well-known location to enhance their self-promotion. In SLink, malicious users can also disguise themselves as users from the target city to engage in fraud or malicious attacks on benign users.

Content-Based Attacks: Because of these applications' openness, users' posts can be indiscriminately viewed and commented on by other users, making communication based on content more vulnerable to malicious attacks. Through the statistical analysis of content-based attacks in the data set used in the experimental section of this article, we found that the content-based attacks can be divided into four main categories, as shown in Fig. 1: 1) 38.7% involved posting false comments to deceive legitimate users; 2) 32.4% involved acting as a distributor of spam messages by continuously posting advertisements in the application; 3) 21.3% involved committing financial crimes [33] similar to telecom fraud against legitimate users; and 4) 7.6% involved offering prostitution services [34] within the application. Observation and analysis of these attack behaviors reveal that most can be identified and judged from the textual content. This is primarily because malicious attacks often employ implicit language in their posts to evade detection, but when they attempt to attack through comments, their text content is more direct.

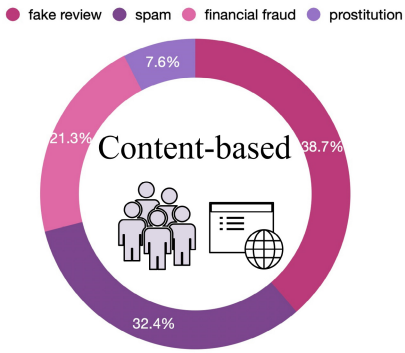

Fig. 1. Classification of content-based attacks. Fake reviews comprised 38.7% of the data, followed by spam at 32.4%, financial fraud at 21.3%, and prostitution services at 7.6%.

Based on the above analysis, we can see that the simplest way for malicious users to achieve large-scale dissemination of malicious content is to post content with malicious intent or to make malicious comments on targeted users' posts, both of which are the content-based attack methods. In addition, the data needed for location-based attack detection is often not public. Therefore, based on these characteristics, we will use posts and comments to identify and detect malicious accounts in SLink.

D. Potential Consequences of Malicious Behavior

Malicious accounts' complex and varied behaviors can lead to potential negative consequences for other legitimate users, platforms, and society.

For other legitimate users, malicious accounts may masquerade as legitimate ones, deceiving victims into providing personal information or money, resulting in financial losses. Malicious accounts may also engage in verbal attacks, sexual harassment, or other forms of online bullying against other users. Meanwhile, malicious accounts may steal personal information from other users and use it for illegal purposes. Interactions with malicious accounts can also leave legitimate users disappointed, frustrated, or depressed leading to emotional harm.

Malicious accounts can erode user trust in platforms leading to user attrition. Additionally, platforms must invest significant resources in identifying and combating malicious accounts, thus increasing operational costs. Ultimately, the behavior of

malicious accounts may damage the platform's brand image and reputation.

For society, malicious accounts can spread false information and rumors, disrupting social order. Meanwhile, malicious accounts can exacerbate social divisions by using online violence and hate speech. The behavior of malicious accounts can also violate societal morals and values.

Therefore, based on the harms as mentioned above, it is imperative to take adequate measures to combat and prevent malicious accounts.

IV. Methodology

This section will provide a comprehensive outline of the GraphTAM's architecture. This is a fine-grained user activity-based approach for detecting malicious accounts, which is primarily composed of three components: 1) a nontextual feature extractor (Section IV-C); 2) a textual feature (TF) extractor (Section IV-D); and 3) a classifier (Section IV-E). What sets this approach apart from the existing malicious account detection methods [7] is that it considers both the descriptive data and text data of users simultaneously.

A. Problem Definition

For the target research object, an open online social networking platform $S = (U, E, A)$, there are three fundamental types of information. U represents the set of users, E represents the relationships between the users, and A represents the set of user activities, such as public comments, likes, and posts.

For a given user set U, we have $U = (u_1, u_2, \ldots, u_N)$. N represents the total number of users in S, and the user set U reflects the basic profile of the user-visible to all the users on the online dating software.

Social relationships between the users are defined as a graph $E = (e_{ij})_{N\times N}$, which reflects the interaction between the users. e_{ij} is a directed link from the user i to the user j, and the weight of the edge e_{ij} quantifies the interaction between the users, which is represented by the total number of comments that the user i made to the user j. For example, if the user i made a total of five comments to the user j, the weight value of the edge e_{ij} is 5. The matrix E reflects the interaction between the users on the online social networking platform. Through E, we can understand the strength of the association between the users and quantify their interaction.

We have a set $A = \{s, r\}$ that represents the fine-grained activities of users, s represents the original public statements users generate, such as posts on the Weibo or tweets on Twitter, and r means the comments users receive on their original public statements. Precisely, s consists of tuples (p, t), where p means the original public speech text and t represents the timestamp at which the text was produced. Similarly, r consists of the tuples (p, p_r, u_r, t), where p, p_r, u_r, and t represent the text content of the original public statement, the text of the received comment, the commenter who commented, and the interaction timestamp, respectively. Therefore, for a given user u, their original posts s and received comments r can be described as $s_u = \{(p_1, t_1), \ldots, (p_{n_1}, t_{n_1})\}$ and $r_u = \{(p_1, p_{r_1}, u_{r_1}, t_{r_1}), \ldots, (p_{n_2}, p_{r_{n_2}}, u_{r_{n_2}}, t_{r_{n_2}})\}$, respectively. At this point, n_1 and n_2 represent the total number of original public statements and comments received, respectively. Based on the user's specific activity information, we can further analyse their behavior. For this study of the open online social networking platform, the primary objective of GraphTAM is to acquire a mapping function that connects the user's features to their respective labels.

TABLE I
Textual-Related Features

Category	Feature Name
Content-related features	Content's topic distribution
	Content shared between 0:00 and 4:00
	Content shared between 4:00 and 8:00
	Content shared between 8:00 and 12:00
	Content shared between 12:00 and 16:00
	Content shared between 16:00 and 20:00
	Content shared between 20:00 and 24:00
	Number of likes received
	Number of views by other users
Comments-related features	Comments' topic distribution
	Comments added between 0:00 and 4:00
	Comments added between 4:00 and 8:00
	Comments added between 8:00 and 12:00
	Comments added between 12:00 and 16:00
	Comments added between 16:00 and 20:00
	Comments added between 20:00 and 24:00

B. Algorithm Framework

Designing GraphTAM is challenging. First, it is not easy to judge whether these attributes are relevant because different attributes have large value ranges between them. Second, traditional malicious account detection algorithms are insufficient in identifying most malicious accounts, as they can resemble legitimate accounts in specific attributes, allowing legitimate users to mimic some attributes of fake accounts to evade detection. Therefore, these challenges need to be addressed in algorithm design.

In this research on the malicious account detection algorithms, it is of utmost importance to carefully choose the relevant features closely related to the final detection efficiency. Specifically, we divide user features into three categories: 1) self-attribute; 2) social relationship; and 3) behavioral. We further divide them into nontext and text features based on their reflective aspects, with specific feature descriptions shown in Tables I and II. Nontext features include user self-attribute features and social RFs, which reflect the overall description of the user. Text features include the text information of posted posts and received comments, reflecting the user's behavioral features. To process these two types of features separately, we designed the algorithm as shown in Fig. 2, which consists of three modules: 1) nontext information module; 2) text information module; and 3) decision maker module.

The input of the nontext information module is the account's descriptive features, namely the user's profile and a relationship matrix to quantify the strength of the relationship between the users. The account's descriptive eigenvector and relational matrix are the inputs of the GCN, which is widely used, for example, in the intelligent transportation systems. Reference [35], the GCN generates high-level representations

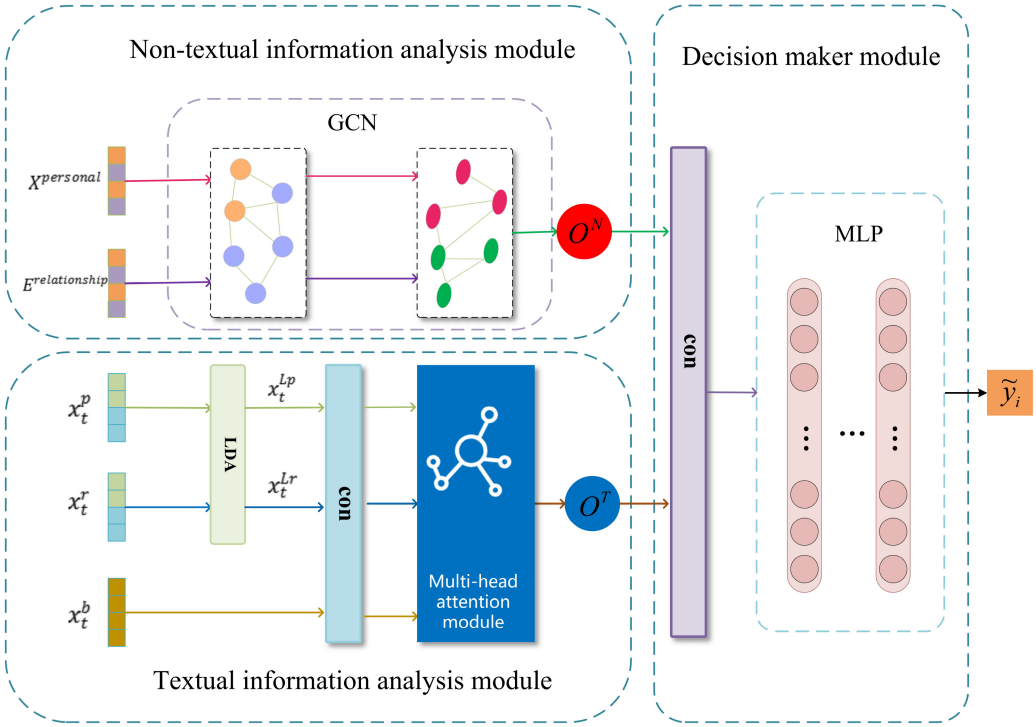

Fig. 2. Framework of GraphTAM. The framework consists of multiple modules, including a text-related information processing module, a nontext-related information processing module, and a decision-making module. It effectively combines nontext-related and text-related information, enhancing the accuracy and performance of malicious account detection.

TABLE II
NONTEXTUAL-RELATED FEATURES

Category	Feature Name
Profile Features	Gender
	Birthdate
	Registration time
	Constellation
	Hobbies
	Job
	Hometown
	Vip or not
	Signature
	Number of followers
	Number of followees
Statistical features about post	Number of posts with pictures
	Number of posts with text
	Number of posts with text and pictures
	Number of posts with audio

that include user self-attribute features and social RFs. Suarez-Tangil et al. [32] have shown that this information helps detect malicious users. The text information module sends behavior features related to text information, including posted posts and comments, respectively, to an MLP and two LDA models to get the relevant embedding features. By connecting the embedding features of each time step of behavior, posts, and comments, we use a series of combination vectors to represent the user's text features. Then, we use the multihead attention module to assign weights to the combination vectors to represent the user's text features. In the decision module, we aggregate the output of the nontext information module and the output of the text information module and use a decision-maker module to judge the legitimacy of the user. The model is trained by minimizing the loss function we defined. We introduce the GraphTAM's process in Algorithm 1.

Algorithm 1: Workflow of GraphTAM

Input: Data of users extracted from the open social networking apps for dating
Output: Prediction result of each user

Generate $X^{personal}$, $E^{relationship}$, x_t^p, x_t^r, x_t^b
Initialize parameters
for *each training iteration* **do**
  Sample a batch of training data
  $O^N = GCN\left(X^{\text{personal}}, E^{\text{relationship}}\right)$
  $x_t^{Lp} = LDA\left(x_t^p\right)$
  $x_t^{Lr} = LDA\left(x_t^r\right)$
  $x_t = \text{concat}\left(x_t^b, x_t^{Lp}, x_t^{Lr}\right)$ // input of multihead attention module
  $O^{\text{T}} =$ Multihead (T) // T results from the aggregation of x_t across all time steps t.
  $\tilde{y}_l = MLP(O^N, O^T)$
  Update GraphTAM parameters using cross-entropy loss for the batch of training data
end
for *each testing iteration* **do**
  Sample a batch of testing data
  Compute and save $\tilde{y}_l$ with trained parameters
end
return $\tilde{y}_l$ for all testing users

In this article's algorithm, the nontext and text information modules focus on different aspects of the user. The non-text information module provides user self-attribute and

social relationship-related features, while the text information module provides a fine-grained representation of the user's primary activities. Experimental results show that the algorithm can achieve better performance by combining these two parts.

C. Nontextual Information Analysis Module

To process nontextual information of user accounts and obtain advanced representations of user-specific and social RFs, we utilized the GCN algorithm, which is a type of deep learning method designed to operate on the graph-structured data. It extends the idea of convolutional neural networks (CNNs) [36] to effectively capture and utilize the information contained in the graph's node relationships [37].

For nontextual information, we applied GCN to capture advanced representations that include user-specific property features X^{personal} and social RFs. We consider a two-layer GCN with the following layer-wise propagation rule:

$$H^{(l+1)} = \sigma\left(\widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{1}$$

where $\tilde{A} = A + I_N$ is the adjacency matrix of the directed graph with added self-connections. I_N is the identity matrix, $\widetilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ and $W^{(l)}$ is a layer-specific trainable weight matrix. $\sigma(\cdot)$ denotes an activation function, such as the $\text{ReLU}(\cdot) = \max(0, \cdot)$. $H^{(l)} \in \mathbb{R}^{N\times D}$ is the matrix of activations in the l^{th} layer, $H^{(0)} = X$.

We take an $N \times D$ feature matrix X^{personal} and an $N \times N$ relationship matrix $E^{\text{relationship}}$ as input, where D is the number of descriptive features for each user. And we first calculate $\widehat{E} = \widetilde{D}^{-(1/2)}\widetilde{E}\widetilde{D}^{-(1/2)}$ in a preprocessing step, where $\tilde{E} = E + I_N$ is the adjacency matrix of the directed graph with added self-connections. I_N is the identity matrix, $\widetilde{D}_{ii} = \sum_j \tilde{E}_{ij}$. As a result, we obtained a new feature matrix H that fused user-specific property features and social RFs. The feature matrix h is given by

$$H = \widehat{E}\text{ReLU}\left(\widehat{E}XW^{(0)}\right)W^{(1)} \tag{2}$$

where $W^{(0)}$ is an input-to-hidden weight matrix for a hidden layer with feature maps. $W^{(1)}$ is a hidden-to-output weight matrix.

Based on the GCN operation described above, we obtained a new feature matrix $H \in \mathbb{R}^{N\times D}$ that includes user-specific property features and social RFs. We used $O^N(= H)$ to represent the output of the nontextual analysis model, which was connected with the output of the text analysis model for final predictions. The columns of the transposed matrix H^{T} are denoted by h_i, representing the new advanced representation of the ith user after processing by GCN.

D. Textual Information Analysis Module

Posting and commenting is a prominent and crucial behavioral activity for online dating apps. The text content that the users post can reflect the purpose of their behavior, which is one of the primary ways for the users to communicate with each other. Our experiments have also shown that text-related features are essential in detecting malicious accounts. Text-related features include user activity characteristics and text characteristics.

To obtain user activity characteristics, we use a multihead attention model to extract features that can represent user granular activities. Multihead attention [38] is a mechanism commonly used in transformer-based models for processing the sequential or graph-structured data. It improves the model's capacity to concentrate on various aspects of the input and capture complex relationships by applying the attention mechanism multiple times in parallel. Each attention head performs attention calculation and captures different relationships and dependencies in the input. This enables the model to pay attention to diverse input segments simultaneously, capturing both the local and global information.

We temporarily do not consider the text content for a given set of user activities A_u, but extract features that can represent user behavioral activities from the remaining tuple sequences. Specifically, we divide the entire data set's duration into a group of continuous time periods with fixed time intervals. In this article, our time interval is one day. For a day (24 h), we further divide every 4 h into a time period and tally the quantity of received posts and comments during each time period to form a vector. We use x_t^b to represent the text-related statistical information for a particular day, which is then combined with the processed text information features as inputs through the multihead attention model for further analysis.

For the text information that the users post and receive, i.e., posts and corresponding comments received, we extract the text information that the users post p_i and the received comment text information p_{r_i} from p_u and r_u, respectively. We use the vectors x_t^p and x_t^r to represent the text information posted and received at the time step t. They are used as inputs to the LDA model for further processing to obtain the topic features of the published posts x_t^{Lp} and the topic features of the received comment information x_t^{Lr}. LDA is a topic model that can produce topic distributions for given documents, initially proposed by Blei et al. [39].

We use the three sequential features obtained, x_t^b, x_t^{Lp}, and x_t^{Lr} to be processed in parallel by the multihead attention model, and finally generate the hidden representation O^{T} that integrates the three features. We use the matrix x_t as the input to the multihead attention model, as follows:

$$x_t = \text{concat}\left(x_t^b, x_t^{Lp}, x_t^{Lr}\right). \tag{3}$$

$X \in \mathbb{R}^{T\times d}$ denotes the combination of x_t in all the time steps where T is the number of total time steps and d is the dimension of the vector x_t.

The formalization of the multihead attention model is given by

$$Q = XW^Q, K = XW^K, V = XW^V \tag{4}$$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^{\text{T}}}{\sqrt{d}}\right)v \tag{5}$$

$$\text{Multihead}(Q, K, V) = \text{concat}(\text{head}_1, \ldots, \text{head}_n)W^o \tag{6}$$

where $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$, Q is the query matrix, K is the key matrix, and V is the value matrix. W^Q, W^K, W^V, and W^O are the projection weight matrices, and $W^Q, W^K, W^V \in \mathbb{R}^{d\times d}$ realize three different linear transformations to map X into different spaces. The matrix $W^O \in$

$\mathbb{R}^{nd\times d}$ controls the output scale. The output of the text-related information analysis module is given by

$$O^{\mathrm{T}} = \text{Multihead}\ (Q, K, V). \tag{7}$$

E. Decision Maker Module

The GraphTAM in this work utilizes a decision-making model for the final judgment. The decision maker is based on the outputs of both the nontextual analysis module and the text-related analysis module and utilizes an MLP model to determine whether the user is legitimate or malicious. The MLP model is a fundamental component of deep learning models and is given by

$$I_i = \text{concat}\left(O^N, O^{\mathrm{T}}\right) \tag{8}$$

$$o_i^l = \begin{cases} \text{softmax}\left(W_i^l I_i + b_i^l\right), l = 1 \\ \text{softmax}\left(W_i^l o_i^{l-1} + b_i^l\right), 1 < l \le L \end{cases} \tag{9}$$

$$\tilde{y}_l = o_i^l \tag{10}$$

where o_i^l, W_i^l, and b_i^l are the output vector, weight matrix, and bias vector of the lth fully connected layer. SoftMax is the nonlinear activation function that yields efficient computation. $\tilde{y}_l$ represents the predicted label of the user i. We use cross-entropy as the loss function, as follows:

$$L = \frac{-\sum_{i=1}^{N}\left(y_i \log(\tilde{y}_l) + (1-y_i)\log(1-\tilde{y}_l)\right)}{N} \tag{11}$$

where y_i is the actual label of the user i.

Once a set of parameters is provided, the L value can be obtained. We choose the parameter set that minimizes L to aid the decision maker. Then, utilizing the generated user data, the trained decision maker can identify malicious users.

V. Performance Evaluation

This section analyses the diversity of user features and evaluates our algorithm. Precisely, we assess the algorithm's effectiveness and examine each feature category's importance.

A. Experimental Setup

Data Set: We will conduct comparative experiments using a data set provided by a well-known open social networking app in China. This data set covers typical functionalities of open social networking apps, including but not limited to personal profile pages, establishment of social relationships, and generation of public content. Each account on this app has an unique ID, which can be used to retrieve the personal profile of each account. Additionally, posts made on this app are uniquely numbered, with each post containing information, such as the poster's ID, timestamp, content, total views, and number of likes. Furthermore, it includes information on each comment, including the commenter's ID, timestamp, and content. The data set comprises 500k verified legitimate users and 150k malicious users, all of whom are active users with at least seven published posts, excluding inactive users due to their limited impact. The entire data set is divided into training, validation, and testing sets in proportions of 50%, 10%, and 40%, respectively. It is noteworthy that, as described above, GraphTAM focuses on representative activities of social networking apps. Given the description of these activities in the preceding sections, it is evident that the other social networking apps also possess these representative activities. Therefore, GraphTAM can also be applied to the other social networking apps or other applications with similar activities.

Implementation Details: In the proposed algorithm, we use a two-layer GCN to obtain advanced representations of user-specific and social RFs, and LDA to retrieve topic distributions of posts and comments, respectively, is utilized. Following Blei et al. [39], we set topic number $K = 100$. The head number in the multihead attention model is 6. For the MLP model, hidden layers are set to 4 and hidden units are set to 16. We use AdaDelta as the optimizer. We used the parameters in [40] and got good results. For all these parameters, we tried different values, but the performance was not as good as the combination of the above parameters, or the improvement was negligible. Therefore, this section chooses to conduct comparative experiments using the parameters as mentioned above.

Evaluation Metrics: In this article, we use precision, recall, and F1-score to evaluate the algorithm's performance. The reason for using these three metrics is that they provide a comprehensive performance assessment, covering the accuracy, recall rate, and overall performance of the model to identify malicious accounts. Precision measures the proportion of instances identified by the model as malicious accounts that are truly malicious, focusing on the model's accuracy. Recall measures the proportion of malicious accounts successfully identified by the model out of the total number of malicious accounts, focusing on the model's recall rate. F1-score combines precision and recall and provides a comprehensive evaluation of the model's performance, focusing on balancing precision and recall to ensure an appropriate balance between the accuracy and recall rate. The combined use of these three metrics ensures a comprehensive evaluation of the model's performance, enabling researchers to understand the algorithm's performance better. The following is a formulaic definition of these three evaluation indicators:

$$\text{Precision} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \tag{12}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{13}$$

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{14}$$

where TP is the number of positive samples correctly identified, TN is the number of negative samples correctly identified, FP is the number of negative samples mistakenly identified as positive samples, and FN is the number of positive samples mistakenly identified as negative samples.

B. Comparison of Legitimate Users and Malicious Users

We conducted mining and analysis of the PFs of users in the data set to uncover hidden information. We selected the number of followers, the number of followings, the number of likes received, and the number of posts published by the users as the research objects. We used the t-test [41] to study and analyse the differences between the malicious and legitimate users in these four metrics. According to the

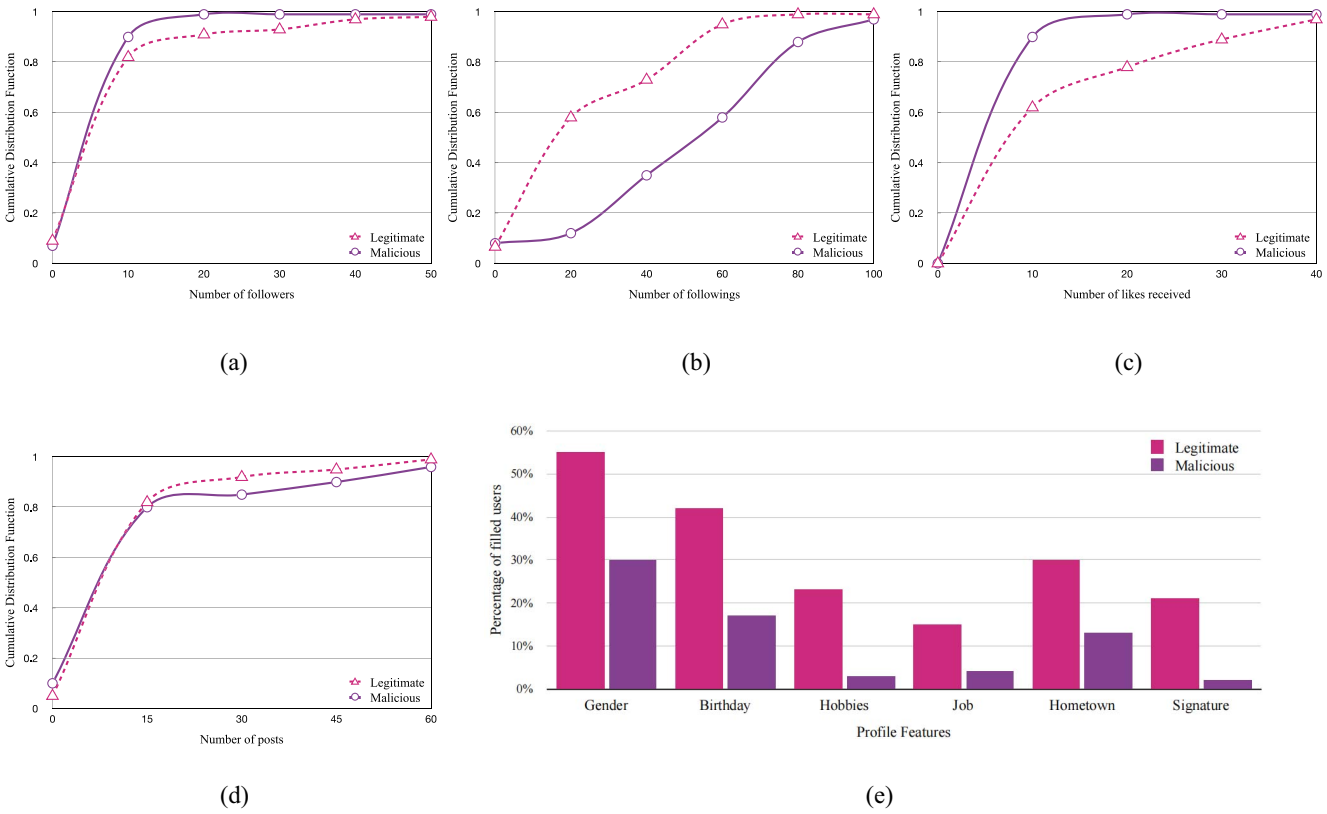

Fig. 3. Fig. 3(a)–(d) uses a cumulative distribution function to reflect the differences in these characteristics between legitimate and malicious users. Fig. 3(e) depicts the differences between malicious and legitimate accounts when filling out the information on the profile page.

content in [41], we calculated the corresponding p-values for each metric. If the p-value is below 0.05, it signifies a noteworthy distinction between the malicious and legitimate users in that metric. Through calculations, we found that the p-values for all the four metrics were less than 0.05, indicating significant differences between the malicious and legitimate users in these metrics. These differences can be attributed to the behavioral and purpose distinctions between the malicious and legitimate users. As depicted in Fig. 3, we use the cumulative distribution function to describe this difference. Fig. 3(a) reflects that the legitimate users tend to have more followers than the malicious users because they actively engage in social networks, thus accumulating more followers. Fig. 3(b) shows that the malicious users often follow more users to expand their influence and facilitate malicious activities. Fig. 3(c) indicates that the legitimate users receive more likes than the malicious users because their content is more popular, contrasting sharply with the malicious users. While malicious users' content may occasionally attract other users, it is perceived as malicious and therefore does not receive likes from others. Fig. 3(d) demonstrates that the malicious users post more than the legitimate users because they often increase their exposure and following by posting more frequently, thus facilitating their malicious activities. From these four figures, we can observe a consistent trend in these four measurement indicators.

Furthermore, we discovered that some optional fields in users' profiles are not necessary to fill out. We conducted a statistical analysis on the filling ratio of these fields. From Fig. 3(e), it is apparent that the malicious users provide limited personal information.

C. Comparison of Different Decision Maker Modules

Several algorithms can serve as classifiers for the decision-maker module of GraphTAM. We employed machine learning algorithms, such as MPL, CatBoost, SVM, linear regression (LR), and C4.5 decision tree (C4.5 DT), to evaluate the performance of the GraphTAM's decision layer.

The performance of these algorithms is shown in Table III. According to the evaluation metrics, it can be observed that the MLP achieves the highest score in terms of F1 score. The reason behind the outstanding performance of MLP lies in its core principles and workflow mechanisms. MLP possesses powerful nonlinear modeling capabilities and effective feature learning abilities. Through multilayer neural connections and the role of activation functions, MLP can better capture complex patterns and relationships between the features in the data, thereby enhancing the model's generalization ability and performance. In contrast, the other four models fail to match the performance of MLP due to limitations in their model complexity and feature representation capabilities. For example, although CatBoost is based on the gradient boosting decision trees, it may be constrained when handling high-dimensional and complex features. While performing well in some cases, SVM and logistic regression may have relatively

TABLE III
EVALUATON ON DIFFERENT DECISION MAKER MODULES

Classification	Precision	Recall	F1 Score
MLP	**0.921**	**0.893**	**0.907**
$CatBoost$	0.920	0.868	0.893
SVM	0.913	0.857	0.884
LR	0.905	0.890	0.897
$C4.5DT$	0.918	0.887	0.902

weaker capabilities in handling nonlinear separable problems. C4.5 decision trees, despite their excellent interpretability and ease of understanding, may have limitations in handling complex relationships and high-dimensional features.

In summary, the MLP demonstrates prominent performance on the malicious account detection task in this work, owing to its deep structure and flexible nonlinear modeling capabilities. Therefore, in subsequent experiments, we employ the MLP algorithm as the classification algorithm for the decision layer of the GraphTAM model.

D. Comparison With State-of-the-Art Approaches

To test the performance of our algorithm in detecting malicious accounts in open social apps, we compare it with the four best current malicious account detection methods.

1) *Weka+RF [18]:* This is a method proposed by Stringhini et al. based on the Weka model and random forest. The authors designed six features and used different features to detect spam according to different scenarios. The algorithm has been deployed on Twitter.
2) *PCA+RF [5]:* This is a machine learning-based malicious account detection algorithm proposed by Al-Qurishi et al. The algorithm first uses a PCA algorithm to process selected user features and then uses a random forest algorithm to classify account types.
3) *SynchroTrap:* Cao et al. [7] proposed a representative malicious account detection algorithm based on constructing an account-to-account graph using the user behavior and behavior time. The algorithm is deployed on Facebook and performs well. The algorithm filters out edges whose weights are less than the threshold and extracts connected users as communities to distinguish account maliciousness based on community density.
4) *Realguard:* Xia et al. [42] proposed a method to detect malicious accounts in privacy-centric mobile social network scenarios. This method is a combination of deep neural networks and random forests, using user profile information and text information from friend request messages to detect malicious accounts. This method has also been deployed on WeChat.

The algorithms mentioned above are currently the most advanced malicious account detection algorithms, some of which have already been deployed in industrial production (e.g., Realguard in WeChat and SynchroTrap in Facebook). Furthermore, they are all representative malicious account detection algorithms, such as PCA+RF which is feature-based; Stringhini which is graph-based; SynchroTrap which is behavior-based; and Realguard which is based on both features and text information. Therefore, to fairly evaluate the performance of our algorithm, we compared it with these existing advanced technologies.

TABLE IV
PERFORMANCE COMPARISON WITH EXISTING STATE-OF-THE-ART SOLUTIONS

Models	Precision	Recall	F1 Score
$Weka + RF$	0.710	0.577	0.637
$SynchroTrap$	0.768	0.576	0.658
$PCA + RF$	0.815	0.685	0.744
$Realguard$	0.847	0.798	0.822
GraphTAM	**0.921**	**0.893**	**0.907**

Table IV shows the experimental results. We can observe that the previous methods perform poorly on the SLink data set. Specifically, we observed that the previous methods cannot achieve an F1-score greater than 0.85. The recall rates of the Weka+RF and SynchroTrap algorithms are even lower than 0.6. In contrast, our algorithm performs significantly better than the other baseline methods, with an accuracy of 0.921, a recall rate of 0.893, and an F1-score of 0.907 indicating that our proposed algorithm demonstrates remarkable effectiveness in identifying malicious accounts within open social applications.

Through the comparative analysis, we summarize the reasons for the experimental results as follows.

1) We divided all account information into text-related information and nontext-related information and analysed them separately.
2) For text-related information, we not only analysed the content of the text but also considered the time-based statistical information generated by the text and used an attention mechanism to fully reveal the suspicious information hidden in the text that the previous methods ignored.
3) When considering nontext-related information, we considered user profile information and the relationship information between the accounts and fused them to obtain a high-level representation containing both types of information.

In contrast, the previous three comparison methods did not consider text-related information, and as shown in our experiments below, text information plays a crucial role in the detection of malicious accounts in open social apps. The final method did not consider the relationship between the accounts and the time-based statistical information related to the text, both of which are important parts of malicious account detection.

E. Ablation Study

Due to the excellent performance of our proposed algorithm, we conducted various experiments to investigate the impact of different categories of features on the algorithm's performance. Specifically, we divided the features into three categories: 1) TFs; 2) RFs; and 3) PFs. We tested all the six possible combinations of these three categories of features, either individually or in pairs.

TABLE V
PERFORMANCE UNDER DIFFERENT FEATURES COMBINATIONS

Features	Precision	Recall	F1 Score
PF	0.753	0.627	0.684
RF	0.662	0.586	0.622
TF	0.859	0.708	0.776
$PF + RF$	0.792	0.540	0.642
$PF + TF$	0.877	0.751	0.809
$RF + TF$	0.860	0.733	0.791
PF+RF+TF	**0.921**	**0.893**	**0.907**

The results of the experiments are shown in Table V. We found that using only one type of feature did not produce satisfactory results. However, combining two types of features led to improved results. This indicates malicious accounts can mimic legitimate user account features and behavior to evade detection.

From the experimental results, we also observed that using only the text-related features of an account can produce relatively good results. This suggests that most of the malicious behavior exhibited by such accounts is carried out through the text-related information, which is consistent with our previous analysis. Even if malicious users try to mimic legitimate user behavior to evade detection, their malicious intent will still be reflected in the text content they post. Adding either of the other two features or both to the text-based features can improve the algorithm's performance to a certain extent. Therefore, it can be proved that these three features are indispensable for the malicious account detection of open dating apps, and GarphTAM uses these three features.

VI. CONCLUSION

In this work, we primarily focus on detecting malicious accounts in open social networking apps. We systematically study heterogeneous feature data in these apps based on the real-world data sets and evaluate our approach on the SLink data set. The results show that our method outperforms the previous state of the art methods in precision and recall, achieving the best performance. The superior details of GraphTAM are analysed as follows.

1) GraphTAM comprehensively utilizes heterogeneous feature data by simultaneously leveraging user interaction text, traditional features, and user RFs, thereby exploiting different types of information to enhance the accuracy of malicious account detection.
2) The application of multihead attention networks. GraphTAM introduces a multihead attention network based on the heterogeneous information, effectively handling the differences between the text and non-text information, thus improving the model's detection capabilities.
3) Wide applicability, the heterogeneous information used by GraphTAM is publicly available to all the users, making this method suitable for detecting malicious accounts in other applications with similar features, demonstrating strong generality and scalability.
4) Evaluation of the importance of text features. Through the analysis experiments on several heterogeneous features, text features exhibit the best performance, further proving the importance of considering text information in open social networking apps, which is a crucial detail where our proposed algorithm outperforms existing methods.

Although we designed GraphTAM to detect malicious accounts in online dating apps, the algorithm still has certain limitations. It cannot achieve real-time detection of malicious behavior to achieve the goal of detecting malicious accounts.

Finally, we will continue to explore this topic to develop algorithms that can detect malicious attacks in real time. In the future, we will also strengthen collaboration with companies to explore integrating location information into malicious account detection models. The location information of online dating app users may contain essential clues to user behavior and identity. Analysing user activity in different geographical locations can better identify abnormal behavior or patterns. For example, if an account appears in two distant geographical locations within a short period, it may be a sign of malicious activity. Additionally, considering that the user location may affect their dating behavior, incorporating location information into the model may help improve detection accuracy. In exploring this field, privacy protection is crucial especially when dealing with sensitive information, such as personal account and behavioral data. The application of federated learning [43], [44] enables the model training to be conducted on the local devices rather than centralized on a central server, thereby aiding in safeguarding user privacy. In future research, we will focus on the user privacy protection and consider adopting new model architectures to enhance the algorithm robustness [45]. These technologies will help us to conduct more in-depth detection of malicious accounts in social networks.

REFERENCES

[1] X. He, Q. Gong, Y. Chen, Y. Zhang, and X. Fu, "DatingSec: Detecting malicious accounts in dating apps using a content-based attention network," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 5, pp. 2193–2208, Sep./Oct. 2021.
[2] J. Cai, W. Liang, X. Li, K. Li, Z. Gui, and M. K. Khan, "GTxChain: A secure IoT smart blockchain architecture based on graph neural network," *IEEE Internet Things J.*, vol. 10, no. 24, pp. 21502–21514, Dec. 2023.
[3] W. Liang, S. Xie, J. Cai, C. Wang, Y. Hong, and X. Kui, "Novel private data access control scheme suitable for mobile edge computing," *China Commun.*, vol. 18, no. 11, pp. 92–103, Nov. 2021.
[4] T. Stein, E. Chen, and K. Mangla, "Facebook immune system," in *Proc. 4th Workshop Soc. Netw. Syst.*, 2011, pp. 1–8.
[5] Y. Wang and W. Xu, "Leveraging deep learning with LDA-based text analytics to detect automobile insurance fraud," *Decis. Support Syst.*, vol. 105, pp. 87–95, Jan. 2018.
[6] B. Wang, Z. Le, and N. Z. Gong, "SybilSCAR: Sybil detection in online social networks via local rule based propagation," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2017, pp. 1–9.
[7] Q. Cao, X. Yang, J. Yu, and C. Palow, "Uncovering large groups of active malicious accounts in online social networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2014, pp. 477–488.
[8] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–442, Jun. 1998.
[9] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters," *Internet Math.*, vol. 6, no. 1, pp. 29–123, 2008.
[10] L. Weng, J. Ratkiewicz, N. Perra, B. Gonalves, and A. Flammini, "The role of information diffusion in the evolution of social networks," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2013, pp. 356–364.

[11] S. Fortunato and M. Barthélemy, "Resolution limit in community detection," *Proc. Nat. Acad. Sci.*, vol. 104, no. 1, pp. 36–41, 2007.
[12] Y. Suhara, Y. Xu, and A. Pentland, "DeepMood: Forecasting depressed mood based on self-reported histories via recurrent neural networks," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 715–724.
[13] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi, "Measuring user influence in twitter: The million follower fallacy," in *Proc. Int. AAAI Conf. Weblogs Soc. Media*, 2010, pp. 1–8.
[14] X. Peng, B. Ribeiro, C. Chen, B. Liu, and D. Towsley, "A study of user behavior on an online dating site," in *Proc. IEEE/ACM Int. Conf. Adv. Soc. Netw. Anal. Min.*, 2013, pp. 243–247.
[15] D. Zytko, S. A. Grandhi, and Q. Jones, "Impression management struggles in online dating," in *Proc. ACM Int. Conf. Support. Group Work*, 2014, pp. 53–62.
[16] B. Viswanath et al., "Towards detecting anomalous user Behavior in online social networks," in *Proc. 23rd USENIX Conf. Secur. Symp.*, 2014, pp. 223–238.
[17] S. Duan, C. Xia, X. Gao, B. Ge, H. Zhang, and K.-C. Li, "Multi-modality diversity fusion network with swintransformer for RGB-D salient object detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2022, pp. 1076–1080.
[18] G. Stringhini, C. Kruegel, and G. Vigna, "Detecting spammers on social networks," in *Proc. 26th Annu. Comput. Secur. Appl. Conf.*, 2010, pp. 1–9.
[19] T. S. Nivas, P. Sriramkrishna, S. S. K. Reddy, P. V. R. G. Rao, and R. S. P. Komali, "Fake account detection on Instagram using machine learning," *Int. J. Res. Eng., Sci. Manage.*, vol. 7, no. 5, pp. 24–26, 2024.
[20] S. S. Khan, A. Deo, K. K. Baraskar, A. Patel, A. Pathak, and A. K. Joshi, "Deep learning based IDS to detect anomaly over social networking site: Comprehensive review," in *Proc. Int. Conf. Integr. Comput. Intell. Syst. (ICICIS)*, 2023, pp. 1–7.
[21] J. Jia, B. Wang, and N. Z. Gong, "Random walk based fake account detection in online social networks," in *Proc. 47th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, 2017, pp. 273–284.
[22] L. Alvisi, A. Clement, A. Epasto, S. Lattanzi, and A. Panconesi, "SoK: The evolution of Sybil defense via social networks," in *Proc. IEEE Symp. Security Privacy*, 2013, pp. 382–396.
[23] B. Wang, J. Jia, and N. Z. Gong, "Graph-based security and privacy analytics via collective classification with joint weight learning and propagation," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2018, pp. 1–15.
[24] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro, "Aiding the detection of fake accounts in large scale social online services," in *Proc. USENIX Conf. Netw. Syst. Design Implement.*, 2012, pp. 1–14.
[25] N. Z. Gong, M. Frank, and P. Mittal, "SybilBelief: A semi-supervised learning approach for structure-based Sybil detection," *IEEE Trans. Inf. Forensics Security*, vol. 9, pp. 976–987, 2017.
[26] P. Wanda and H. J. Jie, "DeepFriend: Finding abnormal nodes in online social networks using dynamic deep learning," *Soc. Netw. Anal. Min.*, vol. 11, no. 1, p. 34, 2021.
[27] W. Gang, T. Konolige, C. Wilson, W. Xiao, and B. Y. Zhao, "You are how you click: Clickstream analysis for Sybil detection," in *Proc. USENIX Conf. Secur.*, 2013, pp. 241–256.
[28] A. Beutel, W. Xu, V. Guruswami, C. Palow, and C. Faloutsos, "CopyCatch: Stopping group attacks by spotting lockstep behavior in social networks," in *Proc. Int. World Wide Web Conf. Steer. Comm.*, 2013, pp. 119–130.
[29] X. Zhou, Y. Hu, J. Wu, W. Liang, J. Ma, and Q. Jin, "Distribution bias aware collaborative generative adversarial network for imbalanced deep learning in Industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 19, no. 1, pp. 570–580, Jan. 2023.
[30] Y. Li, W. Liang, L. Peng, D. Zhang, C. Yang, and K.-C. Li, "Predicting drug-target interactions via dual-stream graph neural network," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, early access, Sep. 8, 2022, doi: 10.1109/TCBB.2022.3204188.
[31] P. K. Roy, A. K. Tripathy, T.-H. Weng, and K.-C. Li, "Securing social platform from misinformation using deep learning," *Comput. Stand. Interfaces*, vol. 84, Mar. 2023, Art. no. 103674.
[32] G. Suarez-Tangil, M. Edwards, C. Peersman, G. Stringhini, A. Rashid, and M. Whitty, "Automatically dismantling online dating fraud," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1128–1137, 2020.
[33] M. Whitty, "The online dating romance scam: The psychological impact on victims—Both financial and non-financial," *SIAM J. Appl. Dyn. Syst.*, vol. 16, no. 2, pp. 176–194, 2017.
[34] K. Albury, J. Burgess, B. Light, K. Race, and R. Wilken, "Data cultures of mobile dating and hook-up apps: Emerging issues for critical social science research," *Big Data Soc.*, vol. 4, no. 2, 2017, Art. no. 205395171772095.
[35] C. Diao, D. Zhang, W. Liang, K.-C. Li, Y. Hong, and J.-L. Gaudiot, "A novel spatial-temporal multi-scale alignment graph neural network security model for vehicles prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 1, pp. 904–914, Jan. 2023.
[36] Y. Fan, B. Xu, L. Zhang, J. Song, A. Zomaya, and K.-C. Li, "Validating the integrity of convolutional neural network predictions based on zero-knowledge proof," *Inf. Sci.*, vol. 625, pp. 125–140, May 2023.
[37] W. Liang, J. Long, K. C. Li, J. Xu, and X. Lei, "A fast defogging image recognition algorithm based on bilateral hybrid filtering," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 17, no. 2, pp. 1–16, 2020.
[38] H. Wu et al., "Multi-head attention-based model for reconstructing continuous missing time series data," *J. Supercomput.*, vol. 79, pp. 20684–20711, Dec. 2023.
[39] D. M. Blei, A. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.
[40] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*.
[41] B. L. Welch, "On the comparison of several mean values: An alternative approach," *Biometrika*, vol. 38, nos. 3–4, pp. 330–336, 1951.
[42] Z. Xia, C. Liu, N. Z. Gong, Q. Li, Y. Cui, and D. Song, "Characterizing and detecting malicious accounts in privacy-centric mobile social networks: A case study," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2019, pp. 2012–2022.
[43] X. Zhou et al., "Digital twin enhanced federated reinforcement learning with lightweight knowledge distillation in mobile networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 10, pp. 3191–3211, Oct. 2023.
[44] X. Zhou et al., "Spatial–temporal federated transfer learning with multi-sensor data fusion for cooperative positioning," *Inf. Fus.*, vol. 105, May 2024, Art. no. 102182.
[45] X. Zhou et al., "Information theoretic learning-enhanced dual-generative adversarial networks with causal representation for robust OOD generalization," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Nov. 17, 2023, doi: 10.1109/TNNLS.2023.3330864.

Yuting Tang received the master's degree from the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, in 2016, where she is currently pursuing the Ph.D. degree.

Her research interests mainly include personalized recommendation and anomaly detection.

Dafang Zhang received the Ph.D. degree in applied mathematics from Hunan University, Changsha, China, in 1997.

He is currently a Professor with the College of Computer Science and Electronic Engineering, Hunan University. He was a Visiting Fellow with Regina University, Regina, SK, Canada, from 2002 to 2003, and a Senior Visiting Fellow with Michigan State University, East Lansing, MI, USA, in 2013. He has authored or co-authored more than 230 journal/conference papers and principal investigator for more than 30 large scale scientific projects. His research interests include dependable systems/networks, network security, network measurement, hardware security, and IP protection.

Wei Liang (Senior Member, IEEE) received the Ph.D. degree in computer science and technology from Hunan University, Changsha, China, in 2013.

He is currently a Professor with the School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China. He has authored or co-authored more than 140 journal/conference papers in high-ranked journals. His research interests include blockchain security technology, network security protection, embedded system and hardware IP protection, fog computing, and security management in wireless sensor networks.

Kuan-Ching Li (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of São Paulo, São Paulo, Brazil, in 2001.

He is a Distinguished Professor with the Department of Computer Science and Information Engineering, Providence University, Taichung, Taiwan, where he also serves as the Director of the High-Performance Computing and Networking Center. He published more than 400 scientific papers and articles and is the co-author or co-editor of more than 50 books published by well-known publishers. His research interests include parallel and distributed computing, big data, and emerging technologies.

Dr. Li is a Fellow of IET.

Keqin Li (Fellow, IEEE) received the B.S. degree in computer science from Tsinghua University, Beijing, China, in 1985, and the Ph.D. degree in computer science from the University of Houston, Houston, TX, USA, in 1990.

He is a SUNY Distinguished Professor with the State University of New York, New Paltz, NY, USA, and a National Distinguished Professor with Hunan University, Changsha, China. He has authored or co-authored over 1000 journal articles, book chapters, and refereed conference papers. He received several best paper awards from international conferences, including PDPTA-1996, NAECON-1997, IPDPS-2000, ISPA-2016, NPC-2019, ISPA-2019, and CPSCom-2022. He holds nearly 75 patents announced or authorized by the Chinese National Intellectual Property Administration. He is among the world's top five most influential scientists in parallel and distributed computing in terms of single-year and career-long impacts based on a composite indicator of Scopus citation database.

Dr. Li was the 2017 recipient of the Albert Nelson Marquis Lifetime Achievement Award for being listed in Marquis Who's Who in Science and Engineering, Who's Who in America, Who's Who in the World, and Who's Who in American Education for over twenty consecutive years. He received the Distinguished Alumnus Award from the Computer Science Department at the University of Houston in 2018. He received the IEEE TCCLD Research Impact Award from the IEEE CS Technical Committee on Cloud Computing in 2022 and the IEEE TCSVC Research Innovation Award from the IEEE CS Technical Community on Services Computing in 2023. He won the IEEE Region 1 Technological Innovation Award (Academic) in 2023. He is a Member of the SUNY Distinguished Academy. He is an AAAS and AAIA Fellow, and an ACIS Founding Fellow.