DatingSec: Detecting Malicious Accounts in Dating Apps Using a Content-Based Attention Network

Xinlei He, Qingyuan Gong, Yang Chen, *Senior Member, IEEE*, Yang Zhang, *Member, IEEE*, Xin Wang, *Member, IEEE*, and Xiaoming Fu, *Senior Member, IEEE*

Abstract—Dating apps have gained tremendous popularity during the past decade. Compared with traditional offline dating means, dating apps ease the process of partner finding significantly. While bringing convenience to hundreds of millions of users, dating apps are vulnerable to become targets of adversaries. In this article, we focus on malicious user detection in dating apps. Existing methods overlooked the signals hidden in the textual information of user interactions, particularly the interplay of temporal-spatial behaviors and textual information, leading to limited detection performance. To tackle this, we propose DatingSec, a novel malicious user detection system for dating apps. Concretely, DatingSec leverages long short-term memory neural networks (LSTM) and an attentive module to capture the interplay of users' temporal-spatial behaviors and user-generated textual content. We evaluate DatingSec on a real-world dataset collected from Momo, a widely used dating app with more than 180 million users. Experimental results show that DatingSec outperforms state-of-the-art methods and achieves an F1-score of 0.857 and AUC of 0.940.

Index Terms—Dating apps, malicious account detection, deep learning, attention mechanism, text analytics

1 Introduction

With the rapid development of the Internet and communication technologies, more and more people switch their traditional dating methods from offline to online. As a new type of social network, dating apps help millions of people find their interested partners. Famous dating apps like Tinder [1], Skout [2], and Momo [3], [4], [5] attract the attention of millions of people all around the world. Online interactions break the traditional geographical restriction between dating users. Therefore, users can communicate with others more easily. Compared with traditional offline dating methods, dating apps have accumulated a large number of users, which not only provide more dating choices but also introduce more risks of malicious attacks.

Similar to traditional social networks such as Facebook and Twitter, users can perform actions like sending posts, making comments, and building social connections with others in dating apps. Differently, it reduces the threshold for establishing communications between users since they are not required to be friends to start a conversation. Such a low barrier in starting a conversation makes dating apps vulnerable to potential adversaries. Sockpuppets can publish fraud opinions to deceive legitimate users [6], spammers keep sending advertisements about malicious activities [7], and prostitution services are also a threat for legitimate users in dating apps [8].

Malicious users are harmful to the user experience, privacy, even personal safety of legitimate users. Therefore, effectively detecting (and eventually removing) malicious accounts in dating apps can be of great benefit for improving user experience, providing better services, and reducing the probability of potential criminal cases. Previous work on defending against malicious attacks in social networks are mainly by three means. The first one is graph-based methods that consider users' social connections [9], [10], [11], [12], [13], [14], [15], [16], assuming that the social connections are limited between malicious users and legitimate users. The second one is machine learning-based methods [6], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], which extract different groups of features and build machine learning-based classifiers to detect malicious users. The third one is activity-based methods [28], [29], [30], [31], [32], [33], [34], which tends to detect malicious users via behavior patterns. However, applying existing methods for malicious user detection in dating apps has two challenges. First, the social connections are relatively loose in dating apps, which makes it harder for the graph-based methods to accurately detect malicious users. Second, the interactions between users are not analyzed at the textual level, which limits the representation ability for machine learning-based and activity-based methods.

In this paper, we are the first to introduce the textual information of user interactions for malicious user detection

- *Xinlei He, Qingyuan Gong, Yang Chen, and Xin Wang are with the School of Computer Science, Fudan University, Shanghai 200433, China, with the Shanghai Key Lab of Intelligent Information Processing, Fudan University, Shanghai 200433, China, and also with the Peng Cheng Laboratory, Shenzhen 518066, China. E-mail: {xlhe17, gongqingyuan, chenyang, xinw}@fudan.edu.cn.*
- *Yang Zhang is with the CISPA Helmholtz Center for Information Security, Saarland Informatics Campus, 66123 Saarbrucken, Germany. E-mail: zhang@cispa.de.*
- *Xiaoming Fu is with the Institute of Computer Science, University of Goettingen, 37073 Gottingen, Germany. E-mail: fu@cs.uni-goettingen.de.*

Manuscript received 12 Oct. 2019; revised 17 Nov. 2020; accepted 14 Jan. 2021. Date of publication 23 Mar. 2021; date of current version 1 Sept. 2021. (Corresponding author: Yang Chen.)
Digital Object Identifier no. 10.1109/TDSC.2021.3068307

in dating apps. It is common for malicious users to publish fraudulent posts or comments. Our observation in dating apps reveals that both posts and comments are important to detect malicious users since they may use implicit words in their posts but discuss more details in their comments (see Section 2.2). However, most of the previous work on malicious user detection overlooked the textual information of user interactions [15], [16], [29], [31], [35].

Based on the observation, we propose DatingSec, a content-based multi-factor attention network to detect malicious users in dating apps. We design a fusion mechanism to combine different pieces of information. Each part of DatingSec deals with the behavior or textual interactions of users. Multi-layer perceptron (MLP) and long short-term memory neural networks (LSTM) [36] are leveraged to summarize the statistical patterns and dynamic patterns like activities, posts, and comments generated by users. Then, the combination of LSTMs' encoding features will be fed into an attentive module to automatically detect the suspicious patterns behind activities, posts, and comments with different attention weights. We also study the contribution of each part of DatingSec in Section 4 and the results show that the textual interaction plays the most important role in detecting malicious users in dating apps.

In summary, we have made the following contributions:

- We are the first to investigate the textual information of user interactions under the context of malicious user detection in dating apps. Our experiment also shows that the textual information contributes the most to malicious user detection.
- We propose DatingSec, a content-based multi-factor attention framework that simultaneously models different aspects of users. Concretely, we use MLP to detect abnormal static properties and LSTM with an attentive module to reveal suspicious signals in users' dynamic behaviors as well as the textual information of user interactions.
- Evaluation performed in a real-world dataset collected from Momo demonstrates that DatingSec outperforms state-of-the-art methods and yields the best performance, which shows a great ability to accurately detect malicious users in dating apps.

Organization. The rest of this paper is organized as follows. In Section 2, we introduce basic functions and malicious activities in dating apps. The system architecture of DatingSec is presented in Section 3. We show the experimental evaluation results in Section 4. Section 5 lists the related work. We discuss several problems and conclude this paper in Sections 6 and 7, respectively.

2 Background and Data Collection

2.1 Basic Functions of Dating Apps

The emerging dating apps like Tinder [1], Skout [2], and Momo [3], [4], [5] provide users convenient ways to create basic profiles. Users can register accounts and fill in their profiles manually, or log in with existing popular social networks like Facebook or Twitter to automatically synchronize their profiles to the dating apps. Uploaded photos and videos, as well as text, can be found in users' profiles and viewed by their friends or nearby strangers with the help of location-based social services. Dating apps allow users to publish posts with location tags. It is a good way for users to know more about people nearby and find potential dating partners. Many dating apps also allow a user to request a list of users that are physically around her.

2.2 Malicious Attacks in Dating Apps

Convenient location-based services enrich the social interactions of users with the motivation of dating, whereas it also introduces potential threats at the same time. There are mainly two types of attacks in dating apps, i.e., location-based attacks and content-based attacks.

- *Location-Based Attacks.* With location-based services, malicious users may conduct fake check-ins at a certain venue to achieve commercial benefits and self-presentations [37], [38]. Furthermore, as pointed out by Xu *et al.* [38], attackers can create a false venue at a certain location, or just impersonate famous venues at a fake location to attract benign users to come. As users are allowed to request dating apps to return lists of nearby users, attackers may utilize these functions to conduct the trilateration attack [39]. Specifically, trilateration allows attackers to locate legitimate users by sending multiple requests to the dating app with different locations. It is a threat to not only privacy but also personal safety.
- *Content-Based Attacks.* Due to the open nature of dating apps, users' posts can be seen and commented by strangers, making the content communications vulnerable to malicious attacks. We randomly select 100 malicious users that have at least 5 posts from the Momo dataset and find about 74 percent of them are conducting malicious behaviors using contents. 13 percent of them are sockpuppets that publish fraudulent opinions to deceive legitimate users [29]. 20 percent are spammers that keep sending advertisements in dating apps. [7]. 8 percent are financial frauds, which are also common in dating apps [40]. 33 percent are recognized as prostitution service providers [8]. An interesting finding is that for sockpuppets, spammers, and financial frauds, most of their malicious behaviors can be observed from the posts. However, most of the prostitution service providers are identified by considering both posts and comments. It is reasonable since this type of malicious users tend to use implicit words to publish posts for escaping detection, but speak more straightforward when they comment to some certain users' posts. Such observation inspires us to take a joint consideration for posts and comments.

In this paper, we mainly focus on content-based attacks for two reasons. First, location-based attacks' detection requires the log data of users' historical locations or users' request history from dating apps, which is not available to the public. Second, as we discussed before, more than 70 percent of malicious users are conducting content-based attacks, which shows that content-based attacks are the major threat for dating apps like Momo.

2.3 Data Collection

Similar to Chen *et al.* [3] and Thilakarathna *et al.* [41], we conduct a data-driven study by using the data of Momo, a representative dating app in China. Founded in 2011 and went IPO on NASDAQ in December 2014, Momo has attracted more than 300 million monthly active users in total.[1] It supports typical functions of dating apps, including maintaining profile pages, building social connections, and generating public contents.

Each post on Momo has a unique post ID, which is an integer number and assigned in ascending order. If a new post has been published, the post ID will increase by one. It means that once we obtain the latest post ID, we can gather all previous posts by descending the post ID and query the API provided by Momo. Each post contains the post information including the post publisher's ID, post time, post content, the total number of views, and the number of likes of the post. At the same time, it also contains each comment's information like comment publisher's ID, comment time, and comment content. After collecting posts and comments data, we can get each user's personal profile by her user ID (publisher's ID in posts or comments). Note that collecting data by referring to users' IDs in mobile social apps is not unusual. Researchers have leveraged such methods to obtain data from Twitter [42], Whisper [43], and Foursquare [44].

We collected the data of all posts and comments that had been published on Momo for about two months. After that, we also collected their corresponding users' information. In total, we collected all the posts and their corresponding comments that were published from Jul. 14, 2016 to Sep. 15, 2016. The dataset consists of 240 million posts, 320 million comments, and 33 million users' profiles.

In the data we collected from Momo, each user's profile contains descriptive information like gender, age, registration time, job, biography, whether she is a premium user (paid for extra service), etc.

Note that a key named "deny code" is also shown on the user's data. We confirmed with Momo that the key was used to represent whether a user was malicious. The key would be set to 1 if this user was malicious, otherwise, the key would be set to 0.

Ethical Considerations: We have taken careful steps to ensure the ethical considerations of collecting and dealing with Momo's data. First, we only use the publicly accessible information for our study. Second, all users' identifiers have been anonymized to preserve privacy. Moreover, all data we collected are stored in an off-line server, which only permits authorized members to login. At last, our study was reviewed and approved by the Institute of Science and Technology, Fudan University.

3 System Design

In this work, we propose DatingSec, a content-based multifactor attention framework to detect malicious attackers in dating apps. We first formally define malicious user detection problem in dating apps, then we present the overview and detailed design of each component in DatingSec.

1. https://www.immomo.com/aboutus.html, accessed on October 1, 2019.

TABLE 1
Notations

Notation	Description
$\mathcal{G}$	The target dating app.
$\mathcal{V}$	The user set.
$\mathcal{E}$	The edge set.
u_i	A user i.
e_{ij}	An interaction from user i to user j.
$\mathcal{A}$	The activity set.
λ	The original user-generated contents.
γ	The interaction user-generated contents.
c	The textual contents.
t	The time step.
x	The features.
h_t^R	The embedding features from Bi-LSTM at time step t.
m_t	The input of attentive module at time step t.
$\hat{y}_i$	The predicted probability of user i being malicious.
y_i	The label of user i.

3.1 Problem Definition

For a target dating app $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$, there exist three basic kinds of information. $\mathcal{V}$ denotes the set of users, $\mathcal{E}$ denotes the social relationships between users, and $\mathcal{A}$ denotes the set of activities conducted by users. We summarize the notations introduced in this paper in Table 1.

For a given user set $\mathcal{V}$, we have $\mathcal{V} = \{u_1, u_2, \ldots, u_N\}$. N denotes the total number of users in $\mathcal{G}$. User set $\mathcal{V}$ reflects the basic information that users show on a dating app.

The social relationship between users can be defined as $\mathcal{E} = (e_{ij})_{N\times N}$. We use the social relationship $\mathcal{E}$ to represent the interactions between users. e_{ij} reveals the total number of interactions from user i to user j. Note that we consider the interactions between users as directed links. Therefore, e_{ij} and e_{ji} reflect different aspects about the interactions between user i and user j. More specifically, we use the weight of the edge e_{ij} to quantify the number of comments between user i and user j, since we consider comments as interactions between users. For instance, if user i makes 10 comments to user j in total, the value of e_{ij} will be 10. $\mathcal{E}$ measures the interactions among users in the dating app. By utilizing $\mathcal{E}$, we can differentiate the strong connections as well as the week connections between users and detect communities in the dating app.

For a given activity set $\mathcal{A}$, we have $\mathcal{A} = \{\lambda, \gamma\}$. λ denotes the original user-generated contents (original UGCs) like publishing a tweet on Twitter or sharing a post on Facebook. γ denotes the interaction user-generated contents (interaction UGCs) like receiving comments from others. Specifically, each element in λ can be represented as a tuple (c, t), where c and t denote the text and timestamp of the original content respectively. Similarly, we use a tuple (c, c_r, u_r, t_r) to represent a record in γ, where c, c_r, u_r, and t_r denotes the original text, received text, received text's publisher, and interaction timestamp respectively (r is the abbreviation of "receive"). Therefore, for a given user u, her original UGCs λ_u and interaction UGCs γ_u can be described as: $\lambda_u = \{(c_1, t_1), \ldots, (c_{n_1}, t_{n_1})\}$ and $\gamma_u = \{(c_1, c_{r_1}, u_{r_1}, t_{r_1}), \ldots, (c_{n_2}, c_{r_{n_2}}, u_{r_{n_2}}, t_{r_{n_2}})\}$, respectively. Where n_1 and n_2 are the total numbers of original UGCs and interaction UGCs,

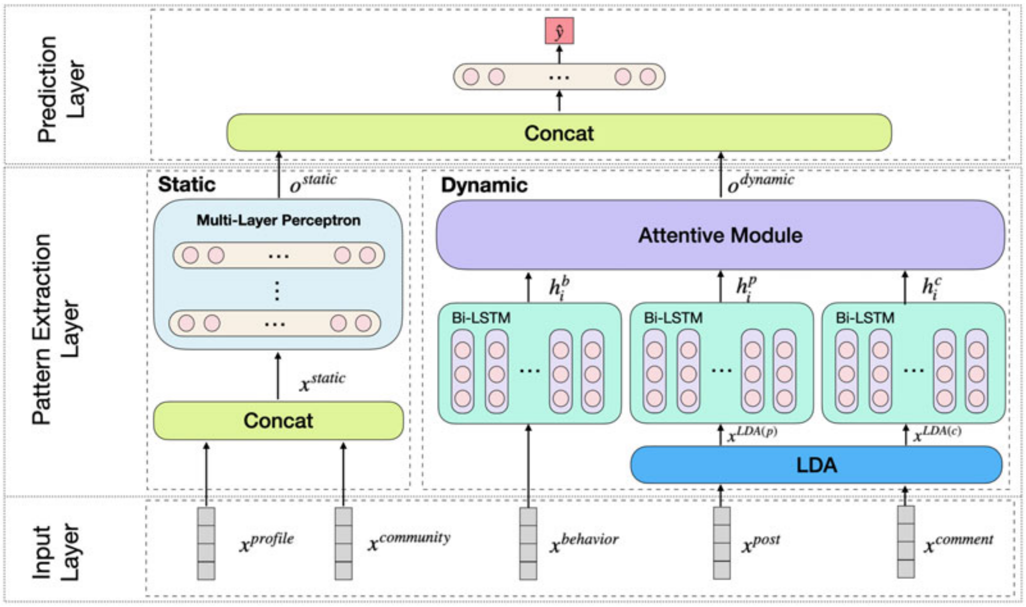


Fig. 1. Framework of DatingSec.

respectively. With the fine-grained activity information of users, we can conduct further analysis about them. Given the information of a dating app $\mathcal{G}$, the goal of DatingSec is to learn a mapping function from a user's features to her label.

3.2 Framework of DatingSec

As shown in Fig. 1, DatingSec consists of three basic layers: input layer, pattern extraction layer, and prediction layer. The input layer takes raw data as input and automatically extracts five types of features, which represent different aspects of activities of users in a dating app. Those features will be fed into the pattern extraction layer. The pattern extraction layer can be divided into a static part and a dynamic part to process the static features and the dynamic features, respectively. For the static part, profile features and community features will be concatenated first and fed into the MLP to capture high-level representations of these features. Such information is proved to be useful in detecting malicious users by Suarez-Tangil *et al.* [27]. Regarding the dynamic part, behavior features, textual features (posts), and textual features (comments) will be fed into three LSTMs respectively to generate their corresponding embeddings features. By concatenating the embedding features of behaviors, posts, and comments at each time step, we have a series of combined vectors to represent a user's dynamic patterns. After that, an attentive module is used to capture the interrelationship of different combined vectors. In DatingSec, the static part and the dynamic part focus on different aspects of users. The static part gives an overview of users' characteristics while the dynamic part provides a fine-grained representation for users' activities. DatingSec achieves better performance by combining the static part and the dynamic part (see Table 7). The prediction layer combines the outputs of the static part and the dynamic part in the pattern extraction layer. Then, a linear layer with softmax function is applied to get the final prediction. We train DatingSec by minimizing the loss between prediction posteriors and the corresponding labels for the training data.

3.3 Input Layer

In this layer, we focus on selecting suitable features to facilitate malicious user detection. Concretely, we extract five types of features from each user, i.e., profile features, community features, behavior features, textual features (posts), and textual features (comments). Details are listed in Table 2. The static features, such as profile features and community features, can give an overall description of a user. Meanwhile, dynamic features like behaviors, posts, and comments can reflect the dynamic patterns of users. Details of these features are shown as follows.

Profile Features. Profile features are extracted from each user's profile. Those features reflect the basic information and statistical behavior of a user. In detail, we use a 20-dimensional vector that includes features like gender, age, the total number of posts, comments, reading times of posts, and number of pictures/URLs hidden in posts. A user's profile features are denoted as $x^{profile}$.

Community Features. Communities exist in dating apps [41]. For each user in a specific community, three metrics are calculated: user's inner-community degree, user's inter-community degree, and the total number of users in this community. A user's community features are denoted as $x^{community}$.

The user's profile features and community features serve as static features, which will be concatenated first as x^{static} and fed into the static part of DatingSec as Fig. 1 shows. We leave it as our future work to study the dynamic changes of community features in dating apps.

Behavior Features. To capture the characteristics of the fine-grained activities of users, we extract features that can represent users' dynamic behaviors to form behavior features. Given the user's activities set $\mathcal{A}_u$, we have two tuple sequences λ_u and γ_u. Note that in the behavior features, we do not consider the textual contents. Instead, we put them into textual features (posts) and textual features (comments), and process them separately with Bilateral-LSTM (Bi-LSTM) models. Therefore, we first remove the textual contents. Then, we extract features that represent users'

TABLE 2
Feature Description of DatingSec

Type	Description
Profile features	Gender Ages Registration time Email (binary) Constellation (binary) Hobbies (binary) Job (binary) Company (binary) School (binary) Length of biography # hangout places Hometown (binary) Premium user (binary) Super-premium user (binary) # posts Signature length Nickname length # favorite books # favorite musics # favorite movies
Community features	Inner-community degree Inter-community degree # users in the community
Behavior features	Post day index Weekday (post) # posts # posts published in 0:00-6:00 # posts published in 6:00-12:00 # posts published in 12:00-18:00 # posts published in 18:00-24:00 # photos in posts # comments received in posts # likes received in posts # locations linked in posts # views by other users Comment day index Weekday (comment) # comments # comments received in 0:00-6:00 # comments received in 6:00-12:00 # comments received in 12:00-18:00 # comments received in 18:00-24:00
Textual features (posts)	Posts' topic distribution
Textual features (comments)	Comments' topic distribution

dynamic patterns from the rest of these two tuple sequences. After that, we split the entire time duration into a set of successive time intervals with a given time interval length (one day in our implementation). For each day, we collect features and form a 19-dimensional vector in this time interval (see Table 2). We split each day into 4 intervals by hours (0:00-6:00, 6:00-12:00, 12:00-18:00, 18:00-24:00) and count the numbers of posts published and the numbers of comments received in those intervals respectively. Behavior features, denoted as $x^{behavior}$, will be fed into a Bi-LSTM model first as Fig. 1 shows. After that, the three types of features will be combined together into the attentive module for further analysis.

Textual Features (Posts and Comments). Different from the previous study, we consider the user-generated contents including the original contents they generated and the interaction contents they received. We extract post contents c_i and comment contents c_{r_i} from λ_u and γ_u respectively, forming two feature vectors x^{post} and $x^{comment}$. They will be further processed by LDA models and generate textual features (posts) and textual features (comments), which can be denoted as $x^{LDA(p)}$ and $x^{LDA(c)}$. Note that the textual features (comments) are different from community features. While the community features only report the numerical results about community statistics, the textual features (comments) consider the textual contents generated by users, which are more informative to discover potential malicious signals. For dating apps where posting comments is one of the main communication types between users, the content posted by a user contains the motivation of the user's behaviors. Our evaluation also shows that the interactions between posts and comments are playing an important role in detecting malicious users (See Section 4).

3.4 Pattern Extraction Layer

To further capture the relationship of static features and temporal dependencies of dynamic features, we use an MLP and multiple Bi-LSTMs with an attentive module. We present the procedure of DatingSec in Algorithm 1.

Algorithm 1. The Workflow of DatingSec

```
   Input: Users' data extracted from the dating app
   Output: Prediction result of each user
   /*Input Layer*/
1  Generate $x^{profile}$, $x^{community}$, $x^{behavior}$, $x^{post}$, and $x^{comment}$, respectively
2  Initialize parameters
3  for each training iteration do
     /*Pattern Extraction Layer*/
4    Sample a batch of training data
     /*Static Part*/
5    $x^{static} = Concat(x^{profile}, x^{community})$
6    $o^{static} = MLP(x^{static})$
     /*Dynamic Part*/
7    $x^{LDA(p)} = LDA(x^{post})$
8    $x^{LDA(c)} = LDA(x^{comment})$
9    $h^b = BiLSTM(x^{behavior})$
10   $h^p = BiLSTM(x^{LDA(p)})$
11   $h^c = BiLSTM(x^{LDA(c)})$
12   $m_t = Concat(h^b, h^p, h^c)$ // input of the attentive module
13   $o^{dynamic} = Attentive(M)$ // $M$ is the combination of
     $m_t$ for all time step $t$
     /* Prediction Layer*/
14   Compute $\hat{y}$ using $o^{static}$ and $o^{dynamic}$
15   Update DatingSec's parameters for the batch of training
       data with cross-entropy loss
16 for each testing iteration do
17   Sample a batch of testing data
18   Compute and save $\hat{y}$ with trained parameters
19 return $\hat{y}$ for all testing users
```

First, the static features x^{static} will be concatenated by profile features $x^{profile}$ and community features $x^{community}$.

Then we feed x^{static} into the MLP as shown on the "Static" part of the pattern extraction layer in Fig. 1. For the dynamic features, x^{post} and $x^{comment}$ will be first handled by two distinct Latent Dirichlet Allocation (LDA) [45] models to generate sequential topic distribution features $x^{LDA(p)}$ and $x^{LDA(c)}$ respectively. Together with behavior features $x^{behavior}$, the three sequential features $x^{behavior}$, $x^{LDA(p)}$, $x^{LDA(c)}$ will be handled by three Bi-LSTMs and generate hidden states step by step. Each hidden state can be regarded as the current status of dynamic patterns. In every step i, the hidden states for behavior features, textual features (posts), and textual features (comments) are denoted as h_i^b, h_i^p, and h_i^c. They will be fed into the attentive module as shown on the dynamic part of the pattern extraction layer in Fig. 1. Finally, the static part generates the output o^{static} while the dynamic part generates the output $o^{dynamic}$. Those two outputs will be fed into the prediction layer to make the final prediction.

Latent Dirichlet Allocation (LDA). LDA is a topic model that can generate topic distributions for given documents. It was first proposed by Blei *et al.* [45]. We leverage LDA to take textual features (posts) x^{post} and textual features (comments) $x^{comment}$ as input and output the post topic features $x^{LDA(p)}$ as well as the comment topic features $x^{LDA(c)}$.

Multi-Layer Perceptron (MLP). MLP is one of the basic components of deep learning models since it can help to learn abstract features from data with different levels of representations [46]. For the static features x^{static}, an MLP is applied to capture the high-level representation among the static part. $x^{static} = Concat(x^{profile}, x^{community})$ is the static features concatenated by the profile and the community features. o^{static} is the output of static part as shown in Fig. 1 ($o^{static} = z^n$). MLP can be formalized as follows:

$$z^n = \begin{cases} \varphi(W^n x^{static} + b^n), & n = 1 \\ \varphi(W^n z^{n-1} + b^n), & 1 < n \leq N \end{cases}, \tag{1}$$

where z^n, W^n, and b^n are the output vector, weight matrix, and bias vector of the nth fully connected layer. $\varphi(\cdot)$ is the non-linear activation function of Rectified Linear Unit (ReLU) [47], which yields efficient computation. MLP is used to learn the overall representation of the numerical features extracted from profile and community. The representation will be concatenated with the output of the dynamic part for final prediction.

Bi-LSTM. We utilize Bi-LSTM to capture the temporal dependency of $x^{behavior}$, x_{LDA}^p, and x_{LDA}^c respectively since those features are sequential and temporally related. Bi-LSTM contains two parallel layers of LSTMs from both forward and backward directions. LSTM can be formalized as follows:

$$f_t^R = \sigma(W_f^R x_t^R + U_f^R h_{t-1}^R + b_f^R) \tag{2}$$

$$i_t^R = \sigma(W_i^R x_t^R + U_i^R h_{t-1}^R + b_i^R) \tag{3}$$

$$o_t^R = \sigma(W_o^R x_t^R + U_o^R h_{t-1}^R + b_o^R) \tag{4}$$

$$c_t^R = f_t^R \odot c_{t-1}^R + i_t^R \odot \tanh(W_c^R x_t^R + U_c^R h_{t-1}^R + b_c^R) \tag{5}$$

$$h_t^R = o_t^R \odot \tanh(c_t^R), \tag{6}$$

where $x^R = \{x^{behavior}, x^{LDA(p)}, x^{LDA(c)}\}$, f_t^R, i_t^R, o_t^R and c_t^R are the vectors of forget gate, input gate, output gate and cell state at time step t respectively. x_t^R and h_{t-1}^R are the input vector and the corresponding generated hidden state vector at time step t and previous time step $t-1$ in LSTM model respectively. The W terms denote the weight matrices for each current input x_t^R of each gate and cell. Similarly, the U terms denote the weight matrices for previous hidden state h_{t-1}^R and b terms denote the bias vectors of each gate and cell. The $\odot$ denotes the *Hardamard* product and the σ denotes the element-wise sigmoid function: $\sigma(x) = 1/(1+\exp(-x))$.

We obtain two hidden states $\overrightarrow{h}_t^R$, $\overleftarrow{h}_t^R$ at each time step t. By concatenating the forward and backward hidden states, we can obtain the overall hidden state: $h_t^R = [\overrightarrow{h}_t^R, \overleftarrow{h}_t^R]$. Hidden state h_t^R serves as the embedding features that contain information of behaviors, posts, and comments respectively. Following previous methods [22], [48], the last hidden state is taken as the output to be the representations of the whole sequences.

The motivation to introduce Bi-LSTM is to analyze the dependency between the constructed time-sequential user activities. We use three Bi-LSTM networks to deal with the descriptive information of users' behavior, posts they published, and comments they received separately. The textural vectors are constructed from the LDA model, making them different from the directly extracted numerical behavior features. To compensate for the separation, we utilize an attentive module over the three Bi-LSTMs to combine the three parts of outputs.

Attentive Module. Inspired by Transformer [49], which is a sequence model that based solely on attention mechanisms, we utilize an attentive module to detect malicious signals from the embedding features of behaviors, posts, and comments. The attentive module can automatically "focus" more on the suspicious time steps while giving less "attention" to the other time steps. The input of attentive module can be formalized as:

$$m_t = Concat(h_t^b, h_t^p, h_t^c), \tag{7}$$

where h_t^b, h_t^p, and h_t^c represent the hidden states of behavior features, textual features (posts) and textual features (comments) at time step t respectively. $M \in \mathbb{R}^{T \times d}$ denotes the combination of m_t in all time steps where T is the number of total time steps and d is the dimension of vector m_t. It first uses the input M to generate three matrices, i.e., the query matrix Q, the key matrix K, and the value matrix V, which can be formalized as:

$$Q = MW^Q \tag{8}$$

$$K = MW^K \tag{9}$$

$$V = MW^V, \tag{10}$$

where $W^Q, W^K, W^V \in \mathbb{R}^{d*d}$ are projection weight matrices that realize three different linear transformations to map M into different spaces. Note that the three weight matrices are initialized with random values and optimized by the gradient of training data. The key matrix K can be considered as a set of "template" patterns of a user while the value matrix V can be regarded as the corresponding "malicious" levels of such "template" patterns. Given the "real" patterns that a user may have from the query matrix Q, one intuition is that we can measure the similarity between "real" patterns and "template" patterns (Q and K). If they are very similar, the corresponding "malicious" level for Q and K would be similar as well. Following the intuition, the attentive module can be further formalized as:

$$Att(Q,K,V) = Softmax\left(\frac{QK^T}{\sqrt{d}}\right)V, \tag{11}$$

where d is the dimension of Q, K, and V to control the scale. Moreover, we apply multi-head attention to enhance the performance. Multi-head attention allows the model to consider the information from different representation subspaces at different positions. For example, some heads may focus on users' behavior patterns in a short period whereas some other heads may focus on users' topic patterns for long periods. For simplicity, we represent the attentive module as:

$$o^{dynamic} = Attentive(M), \tag{12}$$

where $o^{dynamic}$ represents the final output for the dynamic part.

3.5 Prediction Layer

The prediction layer combines the outputs from the static part as well as the dynamic part of the pattern extraction layer. It consists of a fully connected layer with a softmax function.

$$o_i = W_i \cdot Concat(o^{static}, o^{dynamic}) + b_i \tag{13}$$

$$\hat{y}_i = Softmax(o_i), \tag{14}$$

where o^{static} is the output vector of the static part, $o^{dynamic}$ is the output of the dynamic part. W_i and b_i are the weight matrix and the bias vector of this layer. o_i and $\hat{y}_i$ are the logits and posteriors, respectively. Cross-entropy loss is applied as our loss function, which can be formalized as follows:

$$\mathcal{L} = -\sum_{i=1}^{N}(y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)), \tag{15}$$

where y_i denotes the true label of a user and $\hat{y}_i$ denotes the predicted label of this user.

3.6 Summary

In this section, we discuss the detailed design for DatingSec. Compared with previous work, DatingSec is the first to consider the textual information of user interactions in malicious user detection. It takes two types of features into consideration, i.e., static features and dynamic features. As a static view of users, an MLP is applied to learn the overall representation from static features. Besides, to deal with the heterogeneous user behavior data, we leverage different methods and propose a synthetical system design to better extract potential patterns from malicious users. For the dynamic view of users, three Bi-LSTMs are applied to generate the embedding at each time step for dynamic features like behaviors, posts, and comments, respectively. We further calculate each user's "malicious" level by leveraging an attentive module. Combining the output of the static part and dynamic part, DatingSec will make a final prediction for each user.

4 Experiments

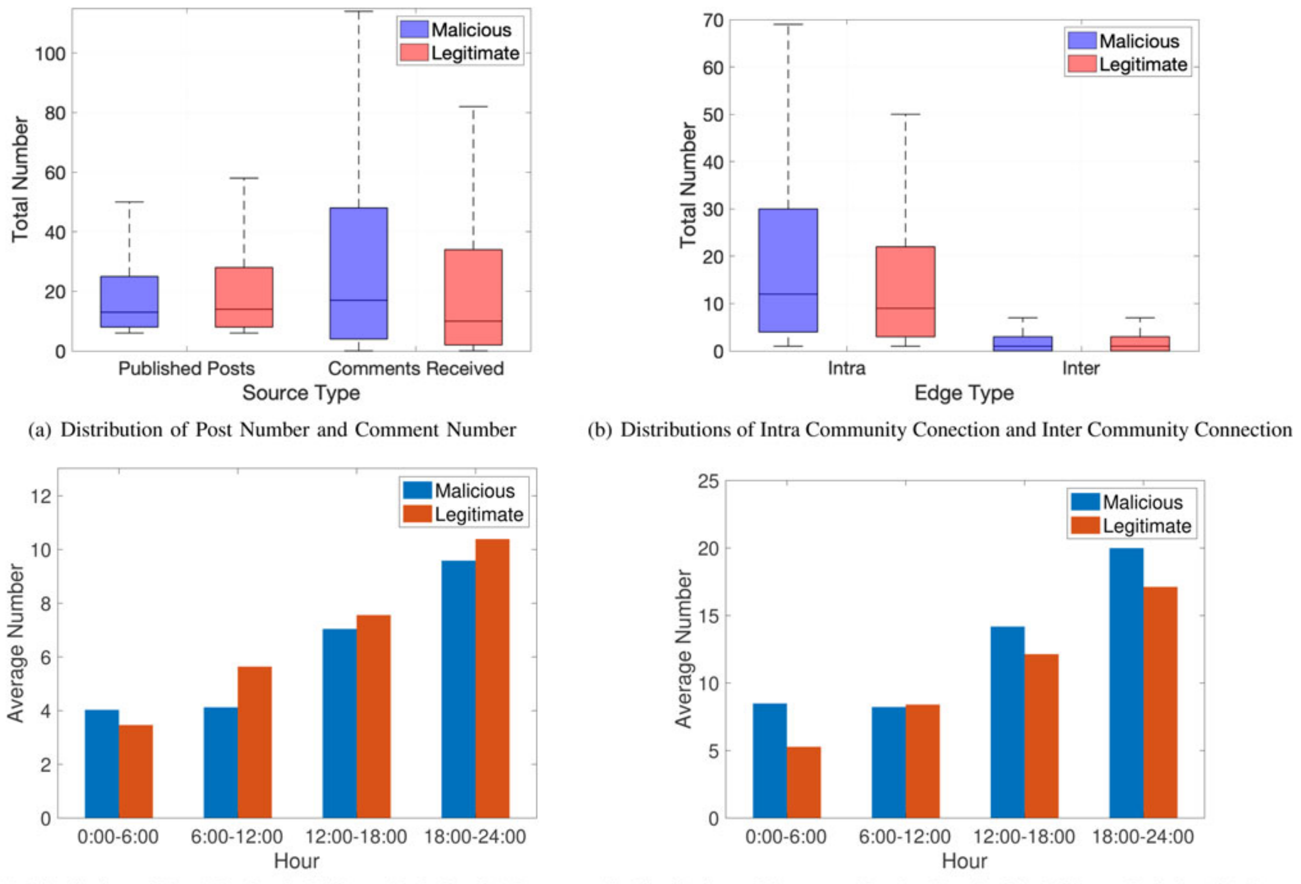
In this section, we choose Momo as a case study to evaluate DatingSec's performance. Note that DatingSec can be applied to various kinds of dating apps since the features concerned about by DatingSec cover representative activities generated by different dating apps.

To process the data, we first extract the largest weakly connected component (LWCC) from the dataset we collected from Momo. There are about 23 million users in the LWCC of Momo. Following the practice in [11], we do not consider inactive users, since their influence is limited.

Instead, we only select users from LWCC who had published at least 5 posts, which consists of 254,042 malicious users and 7,790,532 legitimate users. To evaluate DatingSec on different scales, we randomly select 10K, 20K, 50K, and 100K malicious users as well as the same numbers of legitimate users to form 4 datasets. Note that the dataset with 100K malicious users has the same order of magnitude as total malicious users that published at least 5 posts. For the dataset with 10K malicious users, we yield that it is a proper magnitude since Cao *et al.* [50], Gong *et al.* [22], and He *et al.* [51] also used similar numbers of user instances to train the corresponding deep learning models. Therefore, our evaluation has covered different magnitudes of data. For each dataset, we run the experiments using 5-fold cross-validation and report the average values as well as the standard deviations.

4.1 Comparison Between Malicious Users and Legitimate Users on Momo

We first conduct an analysis of user-generated content and the interactions between users from the dataset with 20,000 users, where half of them are malicious users. Fig. 2a depicts the distributions of published posts and comments received by malicious users and legitimate users. The left part shows that the distribution of posts published by these two groups of users are almost the same and legitimate users might be a little bit higher than malicious ones. However, from the right part, we can figure out that for malicious users, the median number of comments received is higher than that of legitimate users. In Fig. 2b, we show the distributions of intra-community connections (Intra) and inter-community connections (Inter). We run Louvain algorithm [52] to acquire the communities from the LWCC of Momo. Then for each user in our dataset, we calculate the number of edges that she connects with users in the same community as well as the number of edges that she connects with users in other communities. We can see from Fig. 2b that the average intra-community connections are one order higher than that of inter-community connection for both malicious users and legitimate users, which indicates that the users' connections are tighter inside the same community than with other



(a) Distribution of Post Number and Comment Number

(b) Distributions of Intra Community Conection and Inter Community Connection

(c) Distribution of Post Number in Different Periods of A Day

(d) Distribution of Comment Number Received in Different Periods of A Day

Fig. 2. Behavioral difference between malicious and legitimate users on Momo.

communities. Fig. 2c shows the average numbers of posts in hours for malicious users and legitimate users respectively. From Fig. 2c, we find that for legitimate users, they are more active than malicious users during 6:00-24:00. However, malicious users will publish more posts during 0:00-6:00. The reason behind this might be in two folds. First, the total number of posts generated at midnight periods is less than that of other periods in a day. Therefore sending a post at midnight means that their posts are more likely to be seen by other users. Second, for legitimate users who still using dating apps at midnight, they may be more likely to be attracted by malicious users in those periods. Hence sending posts at midnight may increase their attack success rate. Fig. 2d presents the numbers of comments received by malicious users and legitimate users in different periods of a day. Unlike Fig. 2c, malicious users received more comments than legitimate users in most of the periods of a day. We attribute this finding to the fact that malicious users are trained to conduct attacks on legitimate users. Therefore they are more skilled in attracting users' attention and luring legitimate users. In 18:00-24:00, both posts and comments reach the highest numbers across the whole day, which indicates that users on Momo are more active within this period.

4.2 Experimental Settings

Implementation Details. In our model, we utilize LDA [45] to represent the contents of posts and comments on the topic level. We first pre-train the LDA model with our selected 20,000 users where half of them are malicious. Note that LDA is an unsupervised algorithm so that we will not expose label information to the LDA model. Following Blei *et al*. [45], we set topics number $K = 100$. After 100 times of iteration (we have also tried different iteration times like 200 or 300 but received negligible performance improvement), we feed posts and comments into it and get topic distributions respectively. We set the number of hidden layers as 3 and the number of hidden units as 32 for the MLP. We employ 32-dimensional hidden units in both the forward and backward LSTMs and set the head number to 8 in the multi-head attention mechanism for the best performance. Adam [53] is applied as the optimizer with the learning rate of 0.001 and we set the mini-batch size to 100.

Evaluation Metrics. To systematically evaluate the performance of our proposed method, we use four metrics, i.e., Precision, Recall, F1-score, and AUC [54], which are widely used in classification tasks to evaluate the performance of models. Precision denotes the fraction of real malicious users among all classified malicious users, while recall is the fraction of correctly classified malicious users over the total amount of malicious users. F1-score is the harmonic mean of Precision and Recall. AUC measures the probability that the classifier will rank higher of a malicious user than a legitimate user when these two users are selected randomly. We also report (1) number of malicious users correctly classified, i.e., true positive (TP), (2) number of legitimate users correctly classified, i.e., true negative (TN), (3)

TABLE 3
Evaluation Results of Different Approaches (10K:10K)

Methods	*TP*	*TN*	*FP*	*FN*	*Precision*	*Recall*	*F1-score*	*AUC*
DatingSec	8144	9148	852	1856	0.905 ± 0.016	0.814 ± 0.031	**0.857 ± 0.011**	**0.940 ± 0.002**
LDA + MLP	7334	8169	1831	2666	0.800 ± 0.025	0.733 ± 0.031	0.765 ± 0.006	0.828 ± 0.006
PCA + RF	6643	8393	1607	3357	0.805 ± 0.010	0.664 ± 0.011	0.728 ± 0.009	0.804 ± 0.008
SybilBelief	5562	8084	1916	4438	0.744 ± 0.006	0.556 ± 0.014	0.636 ± 0.009	0.705 ± 0.008
SybilSCAR	6727	6027	3973	3273	0.629 ± 0.010	0.673 ± 0.011	0.650 ± 0.008	0.656 ± 0.014
GANG	8691	3374	6626	1309	0.567 ± 0.006	0.869 ± 0.014	0.687 ± 0.009	0.617 ± 0.011

number of legitimate users misclassified, i.e., false positive (FP), and (4) number of malicious users misclassified, i.e., false negative (FN) when comparing DatingSec with other methods.

4.3 Performance Against Existing Approaches

To demonstrate DatingSec's capability in detecting malicious users in dating apps, we compare our model with several representative approaches:

SybilBelief: Gong *et al.* [14] proposed SybilBelief, which was a semi-supervised malicious user detection system. SybilBelief took the whole social graph and a small set of known users (both malicious and legitimate users) as input. Then this algorithm propagated the known label information to the rest of this social graph to determine whether each user was malicious or legitimate.

SybilSCAR: Wang *et al.* [15] developed SybilSCAR, a structure-based algorithm to perform malicious user detection. SybilSCAR combined the advantages of Random Walk (RW)-based methods and Loop Belief Propagation (LBP)-based methods.

GANG: Wang *et al.* [16] proposed GANG, a guilt-by-association method on directed graphs. GANG used a pairwise Markov Random Field to capture the joint probability distribution of features extracted from the social graph.

LDA + MLP: Wang *et al.* [17] proposed an LDA-based text analysis method that combined structural data and used MLP to detect insurance fraud. To implement the algorithm, we feed the static features as well as the posts topics features to the MLP and tune the number of units of each layer for the best experimental result.

PCA + Random Forest: Al-Qurishi *et al.* [18] applied the principal component analysis (PCA) [55] to process selected user features, and fed the result into a Random Forest [56] classifier to identify malicious users in large-scale social networks.

Among the approaches, *SybilBelief*, *SybilSCAR*, and *GANG* are graph-based methods which leverage graph structure to detect malicious users. *LDA + MLP* is a machine learning-based method that considers both the static features and the textual information from posts. However, this method only considers the textual information in a static view. *PCA + Random Forest* is also a machine learning-based method but ignores the analysis for the dynamic features.

For *SybilBelief*, *SybilSCAR*, and *GANG*, we follow their default parameter settings. The maximum iteration time is set to 5, the prior probability of being legitimate for labeled legitimate users, labeled malicious users, and unlabeled users are set to 0.9, 0.1, and 0.5, respectively. For *LDA + MLP*, we set the number of hidden layers as 3 and the number of hidden units as 32 for the best experimental result. For *PCA + Random Forest*, the number of components in PCA is set to 10 for the best performance. For the Random Forest classifier, we set the maximum depth and number of trees to 5 and 100 at the beginning, and use grid search to find the best parameter setting and report the result.

Table 3 summarizes the results of comparisons between different methods in the dataset with 20,000 users in total. We apply 5-fold cross-validation and report the average as well as the standard deviation values. The best results are highlighted in bold. Note that we also evaluate different sizes of datasets and the performance of DatingSec is always the best and stable. The detail results are summarized in Tables 4, 5, and 6.

As shown in Table 3, DatingSec outperforms the other state-of-the-art methods and yields the highest F1-score of 0.857 and AUC value of 0.940, which confirms that our proposed system DatingSec can effectively detect malicious users in the dating app. The advantages of DatingSec are in two folds. On one hand, it takes textual information of posts and comments into consideration, which can reveal the suspicious signals hidden in textual information but ignored by some previous work. On the other hand, it utilizes the Bi-LSTMs to detect the malicious signals hidden in dynamic user activities, posts, and comments. After that, an attentive module has been added to automatically detect the suspicious signals hidden in users' dynamic features. We utilize

TABLE 4
Evaluation Results of Different Approaches (20K:20K)

Methods	*TP*	*TN*	*FP*	*FN*	*Precision*	*Recall*	*F1-score*	*AUC*
DatingSec	16655	17799	2201	3345	0.883 ± 0.011	0.833 ± 0.022	**0.857 ± 0.008**	**0.931 ± 0.002**
LDA + MLP	15051	16269	3731	4949	0.801 ± 0.022	0.753 ± 0.020	0.776 ± 0.001	0.838 ± 0.006
PCA + RF	13098	16838	3162	6902	0.806 ± 0.007	0.655 ± 0.008	0.722 ± 0.007	0.803 ± 0.008
SybilBelief	10835	16516	3484	9165	0.757 ± 0.007	0.542 ± 0.004	0.631 ± 0.003	0.703 ± 0.006
SybilSCAR	11411	14319	5681	8589	0.668 ± 0.003	0.571 ± 0.012	0.615 ± 0.007	0.666 ± 0.005
GANG	14857	10861	9139	5143	0.619 ± 0.004	0.743 ± 0.007	0.675 ± 0.005	0.657 ± 0.004

TABLE 5
Evaluation Results of Different Approaches (50K:50K)

Methods	*TP*	*TN*	*FP*	*FN*	*Precision*	*Recall*	*F1-score*	*AUC*
DatingSec	42620	44701	5299	7380	0.889 ± 0.018	0.852 ± 0.023	$\mathbf{0.870 \pm 0.004}$	$\mathbf{0.939 \pm 0.002}$
LDA + MLP	37647	41362	8638	12353	0.813 ± 0.010	0.753 ± 0.018	0.782 ± 0.005	0.859 ± 0.003
PCA + RF	32916	41958	8042	17084	0.804 ± 0.002	0.658 ± 0.007	0.724 ± 0.004	0.803 ± 0.003
SybilBelief	26878	41744	8256	23122	0.765 ± 0.003	0.538 ± 0.007	0.631 ± 0.005	0.705 ± 0.003
SybilSCAR	25446	39178	10822	24554	0.702 ± 0.002	0.509 ± 0.012	0.590 ± 0.008	0.674 ± 0.005
GANG	29604	35411	14589	20396	0.670 ± 0.004	0.592 ± 0.012	0.629 ± 0.009	0.671 ± 0.007

TABLE 6
Evaluation Results of Different Approaches (100K:100K)

Methods	*TP*	*TN*	*FP*	*FN*	*Precision*	*Recall*	*F1-score*	*AUC*
DatingSec	85813	90299	9701	14187	0.898 ± 0.011	0.858 ± 0.014	$\mathbf{0.878 \pm 0.003}$	$\mathbf{0.943 \pm 0.001}$
LDA + MLP	70989	88152	11848	29011	0.857 ± 0.009	0.710 ± 0.020	0.776 ± 0.009	0.868 ± 0.003
PCA + RF	65796	83947	16053	34204	0.804 ± 0.002	0.658 ± 0.004	0.724 ± 0.003	0.803 ± 0.002
SybilBelief	53110	84055	15945	46890	0.769 ± 0.002	0.531 ± 0.001	0.628 ± 0.002	0.706 ± 0.001
SybilSCAR	52173	79157	20843	47827	0.715 ± 0.003	0.522 ± 0.004	0.603 ± 0.002	0.688 ± 0.002
GANG	52899	77297	22703	47101	0.700 ± 0.002	0.529 ± 0.005	0.602 ± 0.003	0.678 ± 0.003

McNemar's test [57] to examine the performance difference of two classification algorithms and the results show that DatingSec is significantly different from any other state-of-the-art methods (p-value<0.01, McNemar's test). Concerning the metrics we use, DatingSec achieves the highest scores in both the F1-score and AUC.

We also evaluate DatingSec's performance for different percentages of malicious users. Concretely, we fix the number of legitimate users to 100K and vary the number of malicious users to conduct the experiment. The results are summarized in Table 8. Followed by previous work [58], [59], [60], we report the AUC value since it is threshold-independent and insensitive to label distributions. Our observation reveals that DatingSec performs stably with different percentages of malicious users.

4.4 Evaluation of Different Components

Since DatingSec achieves a promising performance, we want to figure out each component's contribution to the whole model.

As shown in Table 7, by using the static features or behavior features only, we can achieve an F1-score of 0.775 and 0.757, respectively. By combining these two feature sets, the F1-score could be improved by 3.7 percent (0.775 to 0.812). However, for malicious users, they can mimic the behaviors of legitimate users, study what they do and follow the rule. Therefore, the behavior patterns of malicious users might be fake and seem similar to legitimate users.

A more interesting finding is that we can only obtain an F1-score of 0.721 and 0.716 by using the textual features of posts or comments, which is even lower than the performance of using static features only. However, if we combine the textual features of posts and comments, we can achieve a more promising performance with an F1-score of 0.824, which is 10 percent higher than the best result by leveraging posts or comments only. As we have pointed out in Section 2.2, for some malicious users, using posts only is not enough for the detection since they may use implicit words to escape the detection. However, detailed information can be found in the comments. Combining the context of posts and comments, DatingSec detects malicious users more effectively.

If malicious users tend to conduct a series of bad behaviors in dating apps, they need to send messages and communicate with legitimate users. Even though they can mimic the normal patterns and pretend to be like legitimate users, their malicious purposes will be exposed in the contents they send. When a malicious user pretends to be a legitimate user, she may change the posting time and frequency to fit legitimate users' patterns. However, malicious users may show malicious intention in textual contents while hiding abnormal behaviors like posing time and

TABLE 7
Evaluation Results of Different Components in DatingSec

Features	*Precision*	*Recall*	*F1-score*	*AUC*
Static	0.842	0.718	0.775	0.852
Behavior	0.823	0.702	0.757	0.852
Static + Behavior	0.870	0.763	0.812	0.897
Textual Features (Posts)	0.725	0.716	0.721	0.786
Textual Features (Comments)	0.847	0.622	0.716	0.847
Textual Features (Posts + Comments)	0.808	0.841	0.824	0.904
Dynamic	0.858	0.838	0.847	0.924
All	0.905	0.814	**0.857**	**0.940**

TABLE 8
Evaluation Results of Different Percentages of Malicious Users

Malicious : Legitimate	*AUC*
1:1	0.943 ± 0.001
1:2	0.955 ± 0.001
1:5	0.954 ± 0.001
1:10	0.954 ± 0.001

TABLE 9
Evaluation Results of Attention Mechanism

Methods	*Precision*	*Recall*	*F1-score*	*AUC*
w/ Attention+Max Pooling	0.905	0.814	**0.857**	**0.940**
w/ Attention+Mean Pooling	0.849	0.850	0.850	0.925
w/o Attention+Max Pooling	0.831	0.867	0.848	0.924
w/o Attention+Mean Pooling	0.881	0.781	0.828	0.923

TABLE 10
Evaluation Results of Different Percentages of Malicious Users

Percentage of poisoned users	*Accuracy over poisoned users*
0%	0.885
1%	0.752
2%	0.727
3%	0.673
4%	0.698
5%	0.619
6%	0.326
7%	0.231
8%	0.025
9%	0.018
10%	0.016

posting frequency. Therefore, to accurately detect malicious users in dating apps, we should take the textual information of user interactions into consideration.

Moreover, if we utilize all dynamic features from behaviors, posts, and comments received, the overall F1-score can reach 0.847 (2.3 percent higher than using textual features of posts and comments received). And the final F1-score by combining static features will be 0.857, which indicates that static features also contribute effect to malicious user detection.

We randomly select 100 malicious users detected by DatingSec and manually identify the types of their malicious activities. 66 percent of the detected malicious users include sockpuppets (20 percent), spammers (14 percent), financial frauds (9 percent), and prostitution service providers (23 percent), are consistent with the malicious types identified by human eyes. The remaining 34 percent of malicious users are identified to conduct malicious activity not observed by human eyes previously. This verifies the usefulness of the sequential analysis on users' historical activities, which is able to find suspicious signals that are not sensitive to human observations.

4.5 Evaluation of Attention Mechanism and Pooling Methods

In this part, we evaluate the usefulness of attention mechanism and different pooling methods. As shown in Table 9 (we only report the average values due to the space limitation), for pooling methods like mean pooling or max pooling, using attention will improve the overall performance of the model since the attention mechanism can automatically detect the suspicious signals behind different features in the whole periods with different weights.

We also compare the effect of different pooling methods, i.e., max pooling and mean pooling. Results show that max pooling performs better than mean pooling. Compared with mean pooling, max pooling is more suitable for malicious user detection. If a user is malicious, it is not smart enough for her to conduct malicious behaviors all the time. Instead, she may conduct a lot of legitimate behaviors to hide her malicious behaviors deeper. If we use mean pooling, it may reduce the effect of her abnormal behaviors and make it harder for the detection.

4.6 Robustness Against Adversarial Attacks

In this paper, we also evaluate DatingSec's robustness against adversarial attacks, more specifically, data poisoning attack [61], [62], [63]. Data poisoning attack aims to add a *trigger* in each poisoned data sample. The trigger can be a specific data pattern in the selected attributes. In the data poisoning attack, the adversary poisons a subset of training data and manipulates their corresponding labels to mislead the detection system. Concretely, we select *Signature Length* and *Nickname Length* in the profile features and set them both to be 100 as our trigger. We evaluate DatingSec's performance with different percentages of poisoned data (malicious users with the trigger and labeled as legitimate users). Due to the space limitation, we only report the detection performance against poisoned data since the detection performance of normal data (both malicious and legitimate users) is almost the same as Table 3. The results are summarized in Table 10. Unlike previous work [63] whose classification accuracy over data poisoning attack drops significantly when there are as many as 5 percent poisoned data, DatingSec still performs well. When the percentage of poisoned users becomes even larger, the detection performance for poisoned users starts to drop, which is similar to previous work [61], [62]. Note that the malicious users we focus on in this paper are different from poisoned users. Malicious users conduct bad behaviors in the real world while poison users are manually generated by the adversary to fool the machine learning classifiers, which requires access to the model training process. Detecting and eliminating poisoned users is out of the scope of this paper. However, we also note that there are various studies [64], [65], [66], [67] focusing on detecting data poisoning attacks, which can be integrated with DatingSec to mitigate potential data poisoning attacks. For example, Neural Cleanse [66] can detect the minimum requirement of poison positions to mislead the model and filter out the potential triggers.

5 Related Work

In this section, we first introduce the related studies about dating apps in Section 5.1, then we discuss existing malicious user detection methods in Section 5.2.

5.1 Studies of Dating Apps

User Behavior. Since dating apps like Tinder [1], Skout [2], and Momo [3], [4], [5] are prevalent in our daily life, various studies about user behavior analysis have been conducted in this area.

Chen *et al.* [3] conducted a study of Momo's users. They considered both spatial and temporal aspects and investigated the potential of behavioral patterns for discovering different categories of users. Ma *et al.* [68] studied the location overlap in a dating app–happn. They found that the information of location overlap could help the user community reduce uncertainty since this kind of information could reflect the similarity between users. Hancock *et al.* [69] showed that deception about information was frequently observed in dating apps. Zytko *et al.* [70] conducted an interview study about impression management in dating apps. Their study revealed that, unlike previous work, people did not want to deceive their online dating partners since it might be discovered if they meet each other in reality. Xia *et al.* [71] have studied the correlation between users' online dating behaviors and various user attributes. Their study revealed that men tended to seek younger women whereas women would consider more about the income and education level of their dating partners.

Privacy and Security Issues. With the open nature of dating apps, privacy and security issues have also disturbed legitimate users and attracted the attention of many researchers.

Cobb *et al.* [1] studied user privacy and usage issues in dating apps. They used questionnaires and interviews to discuss the privacy settings of users on dating apps and explore the reasons behind them. Li *et al.* [4] considered the location privacy issues in dating apps. They studied the precision range of location service in dating apps like Skout and Momo and developed an incremental trilateration strategy to locate users. Hu *et al.* [72] took a study about malicious apps that masqueraded as dating apps to attract users. Real users in those fraudulent apps were lured to purchase premium services to chat with charming users. However, those charming users turned out to be scambots. Li *et al.* [5] discovered that location information leaked by users could be used to infer their demographic information like age, gender, and education level, even when they did not show that information in dating apps.

One work closely related to our paper is conducted by Suarez-Tangil *et al.* [27]. They utilized profile features to detect malicious users. More specifically, demographics, images, and descriptions from user profiles are used to form different types of features. After that, they used the support vector machine (SVM) as the final ensemble classifier to report the prediction results. However, their work did not take the dynamic features generated by users into consideration, which are known to be critical in malicious user detection in our evaluation for dating apps. Note that we have not taken this method into comparison since we are not able to acquire the information of the corresponding images from users.

5.2 Malicious User Detection

In previous work, three kinds of methods are mainly used to detect malicious users in social networks: graph-based methods, machine learning-based methods, and behavior-based methods. We discuss these three categories as follows.

Graph-Based Methods. Researchers tried to identify malicious accounts by leveraging the structures of social graphs. Some approaches are based on the assumption that social links between malicious accounts and legitimate accounts are limited. Cao *et al.* [10] introduced random walk strategies of social graphs to discover the malicious accounts in Tuenti, the largest OSN in Spain. Jia *et al.* [73] proposed SybilWalk, an updated random walk-based method that was more accurate and robust to label noise. Gong *et al.* [14] proposed SybilBelief, a semi-supervised learning framework to disseminate information from a known set of nodes using loopy belief propagation. SybilSCAR [15] and SYBILFUSE [74] combined the advantages of Random Walk (RW)-based methods and Loop Belief Propagation (LBP)-based methods to achieve higher accuracy. Wang *et al.* [16] proposed GANG, a guilt-by-association method on directed graphs. GANG used a pairwise Markov Random Field to capture the joint probability distribution of features extracted from the social graph. Wang *et al.* [35] developed a collective classification framework to detect malicious users by learning the edge weights and malicious scores simultaneously.

We argue that these methods do not fit dating apps like Tinder and Momo who show loose connectivity other than general OSNs like Facebook and Twitter. The sparse connectivity would render the graph-based method less effective.

Machine Learning-Based Methods. Various work utilize machine learning-based methods to identify malicious users in social networks. Zhu *et al.* [23] developed a supervised matrix factorization-based method using activity data in Renren. Zhang *et al.* [21] proposed an SVM-based method to detect malicious users considering location entropy-based metrics in Dianping, one of the dominant location-based social networks in China. Wang *et al.* [17] utilized LDA to extract topic distributions in the textual data, combined with descriptive features, and leveraged MLP to detect malicious patterns about users. Gong *et al.* [22] proposed DeepScan, a deep learning-based approach using the users' spatial-temporal data to uncover malicious accounts in Dianping. Al-Qurishi *et al.* [18] leveraged PCA to reduce the dimension of users' features and utilized a Random Forest classifier to detect malicious accounts on Twitter and YouTube. Yao *et al.* [26] uncovered a new type of malicious attack to help malicious accounts generate huge amounts of fake reviews by using recurrent neural networks. Kumar *et al.* [6] studied the sockpuppets among online communities and used a Random Forest classifier to identify them correctly.

Some previous work considers the textual information of users [17], [18], [26]. Wang *et al.* [17] and Al-Qurishi *et al.* [18] considered the textual information in a static view. Yao *et al.* [26] proposed a method to detect fake reviews generated by machine learning models based on their character distribution. However, a combined consideration between posts and comments is lacking.

Behavior-Based Methods. Another kind of work [28], [29], [30], [31], [32], [33] conducted a series of efforts in detecting malicious users with behavior-based methods. Viswanath *et al.* [30] applied PCA to detect principal components among users' behaviors as normal patterns and detected remarkable deviations as abnormal patterns generated by malicious users on Facebook. Zheng *et al.* [29] proposed a three-stage scheme that considered different periods of user behavior to detect elite sybil users in Dianping. Cao *et al.* [31] developed SynchroTrap, a malicious account detection system that can cluster users according to the similarity of their behavior. However, for malicious users, they can mimic real users and follow their behavioral patterns thus bring more difficulty in detection.

To the best of our knowledge, we are the first who apply textual information of interaction contents and attention mechanisms to malicious user detection in dating apps.

6 Discussion and Limitation

6.1 Implementation in Other Dating Apps

We evaluate DatingSec on Momo since it is a famous dating app and the ground truth label is known to us. Still, DatingSec can also be implemented in other dating apps with only a little extra effort. For example, in various dating apps such as Tinder, Match, and OkCupid, users have their own profile pages that share similar information [27], which can be used to extract the profile features. Since users in those apps can have communications with other users they are interested in, the community features can be extracted from the interactions between users. With the help of detailed information from communications, the behavior features can be extracted. The content published by users can be considered as post contents in DatingSec while the received comments are able to represent the comment contents. Note that the feature dimension of these apps may be different from Momo. However, only a limited amount of effort is needed to fit DatingSec into other dating apps.

6.2 Spoofing DatingSec

Malicious users may mislead the detection of DatingSec by mimicking the normal pattern of legitimate users. We acknowledge that some of the features generated by users can be manipulated with relatively low costs, such as the profile features. However, it will take higher costs for the malicious users to spoof DatingSec by manipulating the dynamic features. Since DatingSec takes a period of users' data into consideration, it may cost more time for the malicious users to carefully forge normal patterns as legitimate users. Moreover, as we discussed in Section 2.2, malicious users may conduct behaviors like requesting money, sending unwanted advertisements, sharing spamming messages, and conducting prostitution services. Those behaviors may require the malicious users to publish specific content to achieve their goals, which can be detected by the dynamic part of DatingSec. Note that there is still some potential to further improve the detection performance of DatingSec. At the moment we only leverage publicly-visible information for detection. In the future, if we could further collaborate with the service providers of the online dating apps, additional information such as the clickstream and back-end activities of users can be incorporated and the detection performance can be enhanced. We leave it as the future work to further improve the performance of DatingSec.

6.3 Retraining

Users' behaviors may change over time in dating apps, some legitimate users may become malicious users in the future. Since DatingSec is trained using a period of users' data, it should be stable for a short time. Still, retraining is helpful to make DatingSec up-to-date. One practical solution is to retrain DatingSec periodically by setting a reasonable time interval. Second, if DatingSec failed to detect a pre-configured amount of malicious users reported by legitimate users, retraining would be activated as well.

6.4 Limitation

We realize that DatingSec might also misclassify users (FN cases and FP cases). We check our 10K:10K experiment and randomly select 100 FN cases (malicious users who are misclassified as "benign" users) to manually verify. We find that 77 of them cannot be verified as "malicious" by the human eye as well. This may be due to two reasons. First, we only collect 9 weeks of data to perform the detection, however, the malicious users may behave benignly in this period. Second, these users may conduct location-based attacks that cannot be detected by DatingSec. For example, 5 out of the 77 users keep sending posts with different cities' names, which indicates that these users might conduct location-based attacks on Momo. Note that in the left 23 FN cases, 8 users are related to prostitution service. 15 users are spammers but only a very small portion of their posts are related to advertisements. We leave it as our future work to filter out such users more accurately. DatingSec might also misclassify benign users as malicious (FP cases). In the real-world deployment, the service provider of a dating app like Momo can run DatingSec alongside other graph-based detection systems [35], [74] and determine the malicious accounts by jointly considering the outputs of multiple systems to reduce the false positive rate (FPR).

7 Conclusion

In this paper, we focus on malicious user detection in dating apps. Compared with previous work, we are the first to leverage the textual information of user interactions into malicious user detection in dating apps. We propose DatingSec, a content-based multi-factor attention network, which considers two types of features, i.e., static features and dynamic features. To deal with the heterogeneous user behavior data, we leverage different methods and propose a synthetical system design to better extract potential patterns from malicious users. Using the real data collected from Momo, the evaluation shows that DatingSec outperforms other comparison methods and achieves the best performance. An important finding is that for all feature sets, the textual features (posts and comments) perform the best (see Table 7), which demonstrates the necessity of considering the textual information of user interactions in malicious user detection. Note that our system is based on publicly-accessible information. Therefore, it can be used by not only the dating app service providers but also third-party application providers to detect malicious users.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62072115, No. 71731004, No. 61602122, and No. 61971145), the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie Grant No. 824019, and the project "PCL

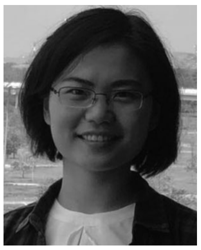
Future Greater-Bay Area Network Facilities for Large-scale Experiments and Applications (LZC0019)".

References

[1] C. Cobb and T. Kohno, "How public is my private life?: Privacy in online dating," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 1231–1240.
[2] R. Xie, Y. Chen, S. Lin, T. Zhang, Y. Xiao, and X. Wang, "Understanding skout users' mobility patterns on a global scale: A data-driven study," *World Wide Web*, vol. 22, no. 6, pp. 2655–2673, 2019.
[3] T. Chen, M. A. Kaafar, and R. Boreli, "The where and when of finding new friends: Analysis of a location-based social discovery network," in *Proc. Int. Conf. Weblogs Soc. Media*, 2013, pp. 61–70.
[4] M. Li *et al.*, "All your location are belong to us: Breaking mobile social networks for automated user location tracking," in *Proc. Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2014, pp. 43–52.
[5] H. Li, H. Zhu, S. Du, X. Liang, and X. S. Shen, "Privacy leakage of location sharing in mobile social networks: Attacks and defense," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 4, pp. 646–660, Jul./Aug. 2018.
[6] S. Kumar, J. Cheng, J. Leskovec, and V. S. Subrahmanian, "An army of me: Sockpuppets in online discussion communities," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 857–866.
[7] L. E. LeFebvre, "Swiping me off my feet: Explicating relationship initiation on Tinder," *J. Soc. Pers. Relationships*, vol. 35, no. 9, pp. 1205–1229, 2018.
[8] 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, pp. 1–11, 2017.
[9] B. Viswanath, A. Post, P. K. Gummadi, and A. Mislove, "An analysis of social network-based sybil defenses," in *Proc. ACM SIGCOMM Conf.*, 2010, pp. 363–374.
[10] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro, "Aiding the detection of fake accounts in large scale social online services," in *Proc. Symp. Netw. Syst. Des. Implementation*, 2012, pp. 197–210.
[11] Z. Yang, C. Wilson, X. Wang, T. Gao, B. Y. Zhao, and Y. Dai, "Uncovering social network sybils in the wild," *ACM Trans. Knowl. Discov. Data*, vol. 8, no. 1, pp. 2:1–2:29, 2014.
[12] A. Mohaisen, A. Yun, and Y. Kim, "Measuring the mixing time of social graphs," in *Proc. ACM Internet Meas. Conf.*, 2010, pp. 383–389.
[13] C. Liu, P. Gao, M. K. Wright, and P. Mittal, "Exploiting temporal dynamics in sybil defenses," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 805–816.
[14] N. Z. Gong, M. Frank, and P. Mittal, "SybilBelief: A semi-supervised learning approach for structure-based sybil detection," *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 6, pp. 976–987, Jun. 2014.
[15] B. Wang, L. Zhang, 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.
[16] B. Wang, N. Z. Gong, and H. Fu, "GANG: Detecting fraudulent users in online social networks via guilt-by-association on directed graphs," in *Proc. Int. Conf. Data Mining*, 2017, pp. 465–474.
[17] 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, 2018.
[18] M. Al-Qurishi, M. S. Hossain, M. A. AlRubaian, S. M. M. Rahman, and A. Alamri, "Leveraging analysis of user behavior to identify malicious activities in large-scale social networks," *IEEE Tran. Ind. Inf.*, vol. 14, no. 2, pp. 799–813, Feb. 2018.
[19] J. Song, S. Lee, and J. Kim, "CrowdTarget: Target-based detection of crowdturfing in online social networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 793–804.
[20] T. Stein, E. Chen, and K. Mangla, "Facebook immune system," in *Proc. 4th Workshop Social Network Syst.*, 2011, pp. 1–8.
[21] X. Zhang, H. Zheng, X. Li, S. Du, and H. Zhu, "You are where you have been: Sybil detection via Geo-location analysis in OSNs," in *Proc. IEEE Global Commun. Conf.*, 2014, pp. 698–703.
[22] Q. Gong *et al.*, "DeepScan: Exploiting deep learning for malicious account detection in location-based social networks," *IEEE Commun. Mag.*, vol. 56, no. 11, pp. 21–27, Nov. 2018.
[23] Y. Zhu, X. Wang, E. Zhong, N. N. Liu, H. Li, and Q. Yang, "Discovering spammers in social networks," in *Proc. AAAI Conf. Artif. Intell.*, 2012, pp. 171–177.
[24] A. Ramachandran, N. Feamster, and S. Vempala, "Filtering spam with behavioral blacklisting," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2007, pp. 342–351.
[25] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design and evaluation of a real-time URL spam filtering service," in *Proc. IEEE Symp. Secur. Privacy*, 2011, pp. 447–462.
[26] Y. Yao, B. Viswanath, J. Cryan, H. Zheng, and B. Y. Zhao, "Automated crowdturfing attacks and defenses in online review systems," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 1143–1158.
[27] G. Suarez-Tangil, M. Edwards, C. Peersman, G. Stringhini, A. Rashid, and M. T. Whitty, "Automatically dismantling online dating fraud," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 1128–1137, Jan. 2020.
[28] K. Thomas, C. Grier, D. Song, and V. Paxson, "Suspended accounts in retrospect: An analysis of Twitter spam," in *Proc. ACM Internet Meas. Conf.*, 2011, pp. 243–258.
[29] H. Zheng *et al.*, "Smoke screener or straight shooter: Detecting elite sybil attacks in user-review social networks," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2018, pp. 1–15.
[30] B. Viswanath *et al.*, "Towards detecting anomalous user behavior in online social networks," in *Proc. 23rd USENIX Secur. Symp.*, 2014, pp. 223–238.
[31] 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.
[32] G. Wang, T. Konolige, C. Wilson, X. Wang, H. Zheng, and B. Y. Zhao, "You are how you click: Clickstream analysis for sybil detection," in *Proc. 22nd USENIX Secur. Symp.*, 2013, pp. 241–256.
[33] K. Thomas, F. Li, C. Grier, and V. Paxson, "Consequences of connectivity: Characterizing account hijacking on Twitter," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2014, pp. 489–500.
[34] J. Huang, G. Stringhini, and P. Yong, "Quit playing games with my heart: Understanding online dating scams," in *Proc. Int. Conf. Detection Intrusions Malware, Vulnerability Assessment*, 2015, pp. 216–236.
[35] 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.*, 2019, pp. 1–15.
[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[37] G. Wang, S. Y. Schoenebeck, H. Zheng, and B. Y. Zhao, "Will check-in for badges: Understanding bias and misbehavior on location-based social networks," in *Proc. Int. Conf. Weblogs Soc. Media*, 2016, pp. 417–426.
[38] F. Xu *et al.*, "Understanding motivations behind inaccurate check-ins," in *Proc. ACM Human-Comput. Interact.*, 2018, pp. 188:1–188:22.
[39] N. P. Hoang, Y. Asano, and M. Yoshikawa, "Your neighbors are my spies: Location and other privacy concerns in GLBT-focused location-based dating applications," in *Proc. Int. Conf. Commun. Technol.*, 2017, pp. 851–860.
[40] M. T. Whitty and T. Buchanan, "The online dating romance scam: The psychological impact on victims - Both financial and non-financial," *Criminol. Criminal Justice*, vol. 16, no. 2, pp. 176–194, 2016.
[41] K. Thilakarathna, S. Seneviratne, K. Gupta, M. A. Kaafar, and A. Seneviratne, "A deep dive into location-based communities in social discovery networks," *Comput. Commun.*, vol. 100, pp. 78–90, 2017.

[42] M. Cha, H. Haddadi, F. Benevenuto, and P. K. Gummadi, "Measuring user influence in Twitter: The million follower fallacy," in *Proc. Int. Conf. Weblogs Soc. Media*, 2010, pp. 10–17.
[43] G. Wang, B. Wang, T. Wang, A. Nika, H. Zheng, and B. Y. Zhao, "Whispers in the Dark: Analysis of an anonymous social network," in *Proc. ACM Internet Meas. Conf.*, 2014, pp. 137–150.
[44] Y. Chen, J. Hu, Y. Xiao, X. Li, and P. Hui, "Understanding the user behavior of foursquare: A data-driven study on a global scale," *IEEE Trans. Computat. Soc. Syst.*, vol. 7, no. 4, pp. 1019–1032, Aug. 2020.
[45] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.
[46] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[47] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 807–814.
[48] Y. Suhara, Y. Xu, and A. S. 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.
[49] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Annu. Conf. Neural Inf. Process. Sys.*, 2017, pp. 1–15.
[50] B. Cao *et al.*, "DeepMood: Modeling mobile phone typing dynamics for mood detection," in *Proc. ACM Conf. Knowl. Discov. Data Mining*, 2017, pp. 747–755.
[51] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 173–182.
[52] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Statist. Mechanics: Theory Exp.*, vol. 2008, 2008, no. 10, Art no. P10008.
[53] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–15.
[54] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.
[55] K. Pearson, "LIII. on lines and planes of closest fit to systems of points in space," *Philos. Mag.*, vol. 2, no. 11, pp. 559–572, 1901.
[56] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[57] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947.
[58] M. Backes, M. Humbert, J. Pang, and Y. Zhang, "walk2friends: Inferring social links from mobility profiles," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 1943–1957.
[59] A. Pyrgelis, C. Troncoso, and E. D. Cristofaro, "Knock knock, who's there? Membership inference on aggregate location data," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2018, pp. 1–15.
[60] I. Hagestedt *et al.*, "MBeacon: Privacy-preserving beacons for DNA methylation data," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2019, pp. 1–15.
[61] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "BadNets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47230–47244, 2019.
[62] Y. Yao, H. Li, H. Zheng, and B. Y. Zhao, "Latent backdoor attacks on deep neural networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2019, pp. 2041–2055.
[63] L. Truong *et al.*, "Systematic evaluation of backdoor data poisoning attacks on image classifiers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 788–789.
[64] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2018, pp. 8011–8021.
[65] B. Chen *et al.*, "Detecting backdoor attacks on deep neural networks by activation clustering," in *Proc. Workshop Artif. Intell. Saf. Co-located AAAI Conf. Artif. Intell.*, 2019. [Online]. Available: http://ceur-ws.org/Vol-2301/paper_18.pdf
[66] B. Wang *et al.*, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *Proc. IEEE Symp. Secur. Privacy*, 2019, pp. 707–723.
[67] X. Xu, Q. Wang, H. Li, N. Borisov, C. A. Gunter, and B. Li, "Detecting AI trojans using meta neural analysis," in *Proc. IEEE Symp. Secur. Privacy*, 2021, pp. 1–18.
[68] X. Ma, E. Sun, and M. Naaman, "What happens in happn: The warranting powers of location history in online dating," in *Proc. ACM Conf. Comput. Supported Cooperative Work*, 2017, pp. 41–50.
[69] J. T. Hancock, C. L. Toma, and N. B. Ellison, "The truth about lying in online dating profiles," in *Proc. Annu. ACM Conf. Human Factors Comput. Syst.*, 2007, pp. 449–452.
[70] D. Zytko, S. A. Grandhi, and Q. Jones, "Impression management struggles in online dating," in *Proc. Int. Conf. Supporting Group Work*, 2014, pp. 53–62.
[71] P. Xia, B. F. Ribeiro, C. X. 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. Mining*, 2013, pp. 243–247.
[72] Y. Hu *et al.*, "Dating with scambots: Understanding the ecosystem of fraudulent dating applications," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 3, pp. 1033–1050, Jun. 2021.
[73] J. Jia, B. Wang, and N. Z. Gong, "Random walk based fake account detection in online social networks," in *Proc. Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw.*, 2017, pp. 273–284.
[74] P. Gao, B. Wang, N. Z. Gong, S. R. Kulkarni, K. Thomas, and P. Mittal, "SYBILFUSE: Combining local attributes with global structure to perform robust sybil detection," in *Proc. IEEE Conf. Commun. Netw. Secur.*, 2018, pp. 1–9.

Xinlei He received the BS degree from the School of Computer Science, Fudan University, in 2017. He is working toward the graduate degree at the School of Computer Science, Fudan University. He has been a research assistant with the Mobile Systems and Networking (MSN) group since 2014. His research interests include machine learning, data mining, and user behavior analysis and modeling. He has been a visiting student with the University of Göttingen and the Southern University of Science and Technology, in 2018.



Qingyuan Gong received the PhD degree in computer science from Fudan University, in 2020. She is now working as a postdoc with Fudan University. Her research interests include network security, user behavior analysis and computational social systems. She has published referred papers in *IEEE Communications Magazine*, ACM TWEB, IEEE TCSS, Springer WWW Journal, ACM CIKM, and ICPP. She has been a visiting student with the University of Göttingen, in 2015 and 2019, also at the University of Chicago in 2018.

Yang Chen (Senior Member, IEEE) received the BS and PhD degrees from the Department of Electronic Engineering, Tsinghua University, in 2004 and 2009, respectively. He is an associate professor with the School of Computer Science, Fudan University, and leads the Mobile Systems and Networking (MSN) group at Fudan. From April 2011 to September 2014, he was a postdoctoral associate with the Department of Computer Science, Duke University, where he served as senior personnel in the NSF MobilityFirst project. From September 2009 to April 2011, he has been a research associate and the deputy head of Computer Networks Group, Institute of Computer Science, University of Göttingen, Germany. He visited Stanford University (in 2007) and Microsoft Research Asia (2006–2008) as a visiting student. He was a Nokia Visiting Professor at Aalto University in 2019. His research interests include online social networks, Internet architecture, and mobile computing. He serves as an associate editor-in-chief of *the Journal of Social Computing*, an associate editor of *IEEE Access*, and an editorial board Member of the *Transactions on Emerging Telecommunications Technologies* (ETT). He served as a OC/TPC Member for many international conferences, including SOSP, SIGCOMM, WWW, IJCAI, AAAI, ECAI, DASFAA, IWQoS, ICCCN, GLOBECOM, and ICC.

Yang Zhang (Member, IEEE) received the PhD degree from the University of Luxembourg, in November 2016. He is a faculty member at CISPA Helmholtz Center for Information Security, Germany. Previously, he was a group leader at CISPA. His research interests include the intersection of privacy and machine learning. Over the years, he has published multiple papers at top venues in computer science, including WWW, CCS, NDSS, and USENIX Security. His work has received NDSS 2019 Distinguished Paper Award. He has served in the technical program committee of USENIX Security 2021, ACM CCS 2021 2020 2019, WWW 2021 2020, AAAI 2021, RAID 2020, ICWSM 2020, and PETS 2021 2020.

Xin Wang (Member, IEEE) received the BS degree in information theory and MS degree in communication and electronic systems from Xidian University, China, in 1994 and 1997, respectively, and the PhD degree in computer science from Shizuoka University, Japan, in 2002. He is a professor with Fudan University, Shanghai, China. His research interests include quality of network service, next-generation network architecture, mobile Internet, and network coding. He is a distinguished member of CCF.

Xiaoming Fu (Senior Member, IEEE) received the PhD degree in computer science from Tsinghua University, Beijing, China, in 2000. He is a full professor of computer science with the University of Goettingen. He was then a research staff with the Technical University Berlin until joining the University of Goettingen, Germany, in 2002, where he has been a professor in computer science and heading the Computer Networks Group since 2007. His research interests include network architectures, protocols, and applications. He is currently an editorial board member of IEEE Communications Magazine, the *IEEE Transactions on Network and Service Management*, and Elsevier *Computer Communications*, and has served on the organization or program committees of leading conferences such as INFOCOM, ICNP, ICDCS, MOBICOM, MOBIHOC, CoNEXT, ICN and COSN. He is IEEE Communications Society distinguished lecturer, a fellow of IET, and member of the Academia Europaea.