The Journal of Supercomputing (2023) 79:18667–18698
https://doi.org/10.1007/s11227-023-05376-z

SiMAIM: identifying sockpuppets and puppetmasters on a single forum-oriented social media site

Ying-Ho Liu[1] · **Chia-Yu Kuo**[1]

Accepted: 3 May 2023 / Published online: 17 May 2023

Abstract
With the Internet becoming indispensable in our lives, social media has become an integral part of our lives. However, with this has come the phenomenon of a single user registering multiple accounts (*sockpuppets*) to advertise, spam, or cause controversy on social media sites, where the user is called the *puppetmaster*. This phenomenon is even more evident on forum-oriented social media sites. Identifying sockpuppets is a critical step in stopping the above-mentioned malicious acts. The identification of sockpuppets on a single forum-oriented social media site has seldom been addressed. This paper proposes a *Single-site Multiple Accounts Identification Model* (*SiMAIM*) framework to address this research gap. We used Mobile01, Taiwan's most popular forum-oriented social media site, to validate SiMAIM's performance. SiMAIM achieved F1 scores between 0.6 and 0.9 on identifying sockpuppets and puppetmasters under different datasets and settings. SiMAIM also outperformed the compared methods by 6–38% in F1 score.

Keywords Sockpuppet · Puppetmaster · Social media site · Classification

1 Introduction

An increasing number of people rely on the Internet in their daily lives. A person may post varied types of posts on social media sites and interact with others. Social media sites fulfill the human need for communication. Two types of social media sites exist, namely, *individual-oriented social media sites* and *forum-oriented social media sites*. [1] The structure of an individual-oriented social media site is based

✉ Ying-Ho Liu
daxliu@gms.ndhu.edu.tw

Chia-Yu Kuo
610735001@gms.ndhu.edu.tw

[1] Department of Information Management, National Dong Hwa University, No. 1, Sec. 2, Da Hsueh Road, Hualien 97401, Taiwan, R.O.C.

on a user. The user usually owns a personal main page where he/she would provide personal information, such as his/her birthday, education status, work experience, and hobbies. The user may express his/her moods, feelings, or opinions on his/her site. This type of site includes Facebook, Twitter, LinkedIn, Instagram, Plurk, and Sina Weibo. In contrast, a forum-oriented social media site is structured in separate forums. Each forum has a topic. Users interact with each other by posting and replying to opinions on the forums on topics they are interested in. The purpose of user participation is to join the public conversation and therefore provide considerably limited personal information. Some examples of this type of site are Reddit, XDA-Developers, GameSpot, and Mobile01, as well as question-and-answer sites such as Quora, Yahoo! Answers, Stack Overflow, and Zhihu. In both types of social media sites, although some content can be viewed by guest accounts, actual participation, e.g., running a personal page and starting a conversation, requires an account.

Nowadays, social media has a significant influence. Because of the human thirst for information, messages on social media sites are constantly read by many people. However, malicious behaviors such as identity deception, bots, sockpuppets, fake news, and Sybil often attack. [2–4] Most negative behaviors relate to malicious accounts and false messages. It is sometimes difficult for people to distinguish the truth from the falsehood of many statements. If manipulated by people, false messages can spread everywhere on social media sites, causing considerable adverse effects. For example, in early 2020, unknown sources on social media began making fake reports regarding the supposed treatment of COVID-19. For instance, eating more coconut oil could treat the virus [5]. These fake reports broadly spread on many social media sites and instant message platforms. Sometimes, more severe events may ensue from a malicious message. In 2016, anonymous users of 4chan, a popular forum-oriented social media site, started to spread an allegation that former U.S. President Bill Clinton and his wife, Hillary Clinton, kidnapped and trafficked children through a pizza restaurant in Washington, DC [6]. The allegation lacked objective evidence but spread instantly across many social media sites. On December 4, 2016, a 28-year-old man walked into a Washington DC pizza store with an assault rifle. The young man claimed he wanted to investigate whether the allegation about the Clintons was true. The young man was finally arrested.

People also like to seek opinions from social media sites. Many forum-oriented social media sites provide forums for discussions on consumer products. Users can post their comments about a product on these forums and discuss the product with other users. Positive or negative comments on such forums considerably influence the users' intention to buy the products. Therefore, some people try to manipulate the comments on the sites, spreading words favorable to themselves or unfavorable to their opponents. A typical case took place in Taiwan. A leading global smartphone company hired a marketing company to post fake negative consumer comments about its competitor's smartphones in 2013 on Mobile01.com, the most popular forum-oriented social media site in Taiwan [7]. Although the smartphone company claimed it was an employee's behavior, the Fair Trade Commission in Taiwan eventually imposed a fine on this leading global smartphone company. In this case, the marketing company registered multiple accounts on Mobile01.com to spread fake negative comments. Multiple accounts registered by an entity, e.g., a person or a company, are called the *sockpuppets* of the entity, and the entity is the *puppetmaster*. Although this event occurred years ago, the sockpuppets have not disappeared on Mobile01.com. Sockpuppets are common on social media sites nowadays. Since a sockpuppet can be malicious, it is crucial to identify the sockpuppets and puppetmasters.

Figure 1 shows the relationship between users, accounts, sockpuppets, and puppetmasters. In the figure, four users are A, B, C, and D. Each user creates one or more accounts on social media site(s). For example, user A owns three accounts, and user B owns two. The accounts may exist on one or more social media sites. The three accounts held by user A are sockpuppets of user A, and user A is a puppetmaster. Similarly, users B and C are puppetmasters, each with two sockpuppets. User D is not a puppetmaster since user D only owns an account.

In the extant literature, many studies have aimed to identify sockpuppets on one or across several individual-oriented social media sites. The identification of sockpuppets and puppetmasters within a single forum-oriented social media site has seldom been addressed. To address this research gap, we propose a *Single-site Multiple Accounts Identification Model* (*SiMAIM*) framework. SiMAIM retrieves attributes from multiple accounts' profiles, text content, and social networks and constructs identification models using the retrieved attributes to identify sockpuppets and puppetmasters. The key contributions of this paper are as follows. First, this paper proposes the identification of sockpuppets and puppetmasters on a single forum-oriented social media site, which has seldom been addressed in the literature. Second, this paper presents a set of attributes covering account profiles, text content, and the social network connecting the accounts. Third, the proposed SiMAIM framework can be extended by adding new attributes and classification algorithms for different social media sites. Fourth, this paper evaluates the performance of SiMAIM on real datasets from Mobile01.com. Fifth, a performance comparison between SiMAIM and existing methods presents. Sixth,

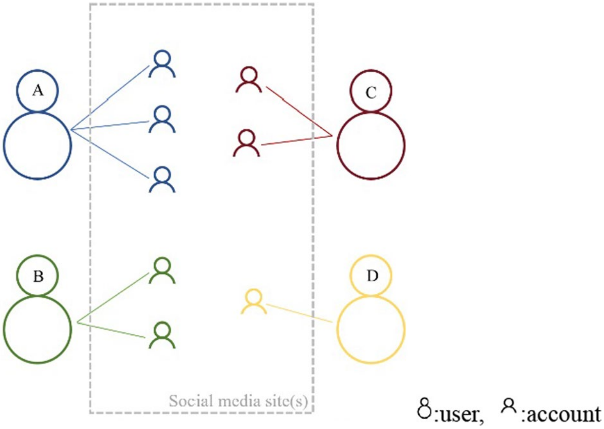


Fig. 1 Users and accounts

the experiments contain an ablation study reporting the effectiveness of different combinations of attributes.

The remainder of this paper is structured as follows. Section 2 contains a review of the literature on the identification of sockpuppets. We discuss the proposed method in detail in Sect. 3. In Sect. 4, we present the experiment results. Section 5 contains some concluding remarks.

2 Literature review

Identifying sockpuppets and puppetmasters on social media sites has been a challenging research topic, and its solution would have a wide range of applications. For example, in security, such identification can be used to identify malicious users on social media and prevent phishing, spamming, and identity theft by detecting the behavior of these malicious users. The above-mentioned identification can be used to improve privacy policies or prevent user identity leakage. Regarding referrals, it can help users create friend referral lists and recommend friends to users by linking information from different social networks.

Most of the literature focused on sockpuppet identification across social media sites. The existing studies have used three types of attributes to identify sockpuppets, namely, profile-based attributes (e.g., account name, nickname, gender, and place of residence), content-based attributes (e.g., posts published by users and replies to posts), and social network-based attributes (e.g., friends and followers) [8]. The existing studies can be categorized into two first-level categories: (1) studies on multiple social media sites and (2) studies on a single social media site. In addition, each category can be further divided into two second-level categories: (1) studies using a single type of attribute and (2) studies using multiple types of attributes. Next, we examine the vital research in this area. Section 2.1 presents studies on multiple social media sites. Section 2.2 offers studies on a single social media site. Section 2.3 gives a discussion.

2.1 Studies on identification across social media sites

Most studies have managed to find sockpuppets across social media sites. Sections 2.1.1 and 2.1.2 review studies using single and multiple types of attributes, respectively.

2.1.1 Studies using a single type of attribute across social media sites

Chung, Lin, Lin, and Cheng [9] used friends' personal information and social messages to identify users across social network sites. They proposed a two-stage clustering algorithm to generate social summaries for each person. Tan et al. [10] presented using the community structure to improve the effectiveness of mapping accounts. They used Manifold Alignment on Hypergraph (MAH) to sort the possible matching users in other social networks. Vosecky, Hong, and Shen [11] integrated their online friends into a context that made it easier for users to connect with their friends and improved the identification of friends across multiple social media sites. They also proposed to identify users by examining account content. Zhou, Liang, Zhang, and Ma [12] proposed the Friend Relationship-Based User Identification (FRUI) algorithm, a network structure-based identity recognition method, and applied it to Sina Weibo and Renren. Later, Qu et al. [13] proposed a similar approach. Zhang, Shu, Liu, & Wang [14] and Zhou et al. [15] modeled the user–user social network by graph neural networks. Zafarani and Liu [16] proposed the MOBIUS approach by adopting user behavior modeling, which involves identifying unique user behaviors to construct user features and using supervised learning to identify users. Motoyama and Varghese [17] proposed a system that uses user profiles to search and match individuals on social media sites. Zheng, Li, Chen, and Huang [18] proposed to build classification models, namely, decision tree, neural network, and support vector machine (SVM), based on the writing style (vocabulary, syntax, structure, and content). Hao, Zhou, Cheng, Huang, and Wu [19] used IP addresses to map online users' records (e.g., activities on social media sites) to offline users' records (e.g., check-in data of hotels).

2.1.2 Studies using multiple types of attributes across social media sites

Jain, Kumaraguru, and Joshi [20] proposed an identity search algorithm based on two types of attributes, content and social network. Their idea was implemented as an identification system Finding Nemo [21]. Considering the limited user information on social media sites, Zhang, Wang, Li, and Xiao [22] applied a classifier for user feature extraction and proposed the Social Link Identification (SIL) method to find the best matching user accounts with data from Sina Weibo and Renren. Nie et al. [23] proposed the dynamic core interests mapping (DCIM) algorithm, which considers the social network structure of users and the content of their posts, and analyzes the core topics of their posts. Goga et al. [24] collected the geographic location, timestamp of posts, and the writing style of users on Yelp, Flickr, and Twitter as the three main characteristics to identify users. A binary logistic regression classifier identifies puppet accounts across multiple social media sites. Xing, Deng, Wu, Xie, and Gao [25] observed that users tend to have similar display names across social media sites. Their proposed method retrieved the length, character, and letter features from the display names and combined features with text content to identify users. Zhong, Cao, Guo, and Nie [26] proposed CoLink, which considered profile-based attributes and social network-based attributes, to link users across social media sites.

2.2 Studies on identification on a single social media site

Furthermore, some studies have dealt with sockpuppets on a single social media site. Sections 2.2.1 and 2.2.2 review studies using single and multiple types of attributes, respectively.

2.2.1 Studies using a single type of attribute on a single social media site

Zhou et al. [27] observed that, after being blocked, a puppetmaster might create new sockpuppets to maintain previous social relationships. Their identification method considers time-series information to construct a time-series social network. Liu, Wu, Han, and Zhou [28] claimed that accounts with similar sentiment orientations are likely sockpuppets created by a puppetmaster. Their method first retrieves the sentiment orientations of accounts' comments. Then, a similar-orientation network is constructed where nodes represent accounts, and edges indicate the sentiment orientations of two accounts. Finally, community detection algorithms identify sockpuppets from the similar-orientation network. Two studies [29, 30] used sockpuppets' propagation characteristics and propagation structure to identify sockpuppets. The topological relation of posts and subsequent replies generates propagation trees. The propagation trees of sockpuppets and benign accounts differ in the properties of propagation trees, such as depth, width, and size of trees. Maity, Chakraborty, Goyal, and Mukherjee [31] retrieved two tweet features, i.e., the entropy of tweets and normalized retweets count, and profile-based features, such as verified or not, followers count, and friends count from accounts on Twitter. These features are combined with SVM, Logistic Regression, and random forest to predict sockpuppets on Twitter. Almishari and Tsudik [32] explored the linkability of reviews (and accounts) on a single site. They extracted each review's unigrams, diagrams, ratings, and categories. They used naïve Bayes classifier and symmetric Kullback–Leibler Divergence to link an anonymous review to an account on Yelp.

Wikipedia non-periodically blocks malicious accounts according to the accounts' editing activities. Some of these blocked accounts are sockpuppets. Based on the contents of the blocked accounts' talk pages, several studies intended to build a system for automatically identifying the sockpuppets on Wikipedia. For example, Yamak, Saunier, and Vercouter [33] proposed SocksCatch to detect sockpuppets. SocksCatch considered the characteristics of accounts' actions, such as account creation time, account first action time, and action occurrence time, to identify sockpuppets. Tsikerdekis and Zeadally [34] used non-verbal user behavior as input variables for SVM, random forest, and AdaBoost algorithms. Solorio, Hasan, and Mizan [35] extracted features representing an account's stylistic, grammatical, and formatting preferences to combine with SVM classifier to identify sockpuppets.

2.2.2 Studies using multiple types of attributes on a single social media site

Yu et al. [36] retrieved verbal features, such as text length and question frequency, and non-verbal features, such as the speaking time of each user and the number of comments per user per day from English Wikipedia. An adaptive feature selection process is used to select important features. Finally, an SVM model is constructed using the selected features to identify sockpuppets.

2.3 Discussion

Most previous studies focused on identifying sockpuppets on individual-oriented social media sites. Wikipedia is a particular case. However, the studies concerning Wikipedia used the content of accounts' pages, particularly the *talk pages*. The talk pages are also individual-oriented. Few studies have targeted the identification of sockpuppets and puppetmasters on a single forum-oriented social media site.

We have compiled the above-mentioned studies in Table 1, which presents the types of attributes used and the type of social media sites targeted in each study. Zafarani and Liu [16] used both types of social media sites. Hao, Zhou, Cheng, Huang, and Wu [19] and Almishari and Tsudik [32] used datasets from Yelp.com, which does not belong to these two types of sites. While previous studies did not consider all three types of attributes, our proposed SiMAIM framework used all three types of attributes to achieve the best recognition results.

Studies on identification across multiple sites usually assume that sockpuppets from different sites have similar profiles, contents, and social networks. However, a forum-oriented social media site usually contains minimal user profiles. Sockpuppets on a single site often deliberately differ from one sockpuppet's profile (display name, IP address, geographic location, timestamp of posts, gender, education, workplace, etc.) and writing style (vocabulary, syntax, structure, etc.) to others' profiles and writing style. Sockpuppets also tend to maintain different social networks on a single site. Therefore, these studies cannot effectively identify sockpuppets and puppetmasters on a single site.

Most studies on identification on a single site focus on individual-oriented social media sites. Those studies using profile-based attributes are usually not applicable to the identification on forum-oriented social media sites where profiles are limited and intentionally different from others. In addition, users are less motivated to engage in friendship on forum-oriented social media sites since discussions over specific topics are the leading cause of why people stay on the sites. To our knowledge, the only existing study (Liu, Wu, Han, and Zhou [28]) that performed identification on a single forum-oriented social media site used content-based attributes. We design novel attributes covering profiles, content, and social networks of accounts to complement literature. We also compared Liu, Wu, Han, and Zhou's method with SiMAIM in Sect. 4.5.

In short, considerable research has been conducted to find sockpuppets on one or multiple individual-oriented social media sites instead of identifying sockpuppets and puppetmasters on a single forum-oriented social media site. Therefore, we have focused on this research gap and hope to solve this critical problem. Moreover,

Table 1 Literature on the identification of sockpuppets

Site number	Attribute type number	Literature	Profile	Content	Social network	Individual or forum
Multiple sites	Single type	Chung, Lin, Lin, and Cheng [9]			✓	Individual
		Tan et al. [10]			✓	Individual
		Vosecky, Hong, and Shen [11]	✓			Individual
		Zhou, Liang, Zhang, and Ma [12]			✓	Individual
		Qu et al. [13]			✓	Individual
		Zhang, Shu, Liu, and Wang [14]			✓	Individual
		Zhou et al. [15]			✓	Individual
		Zafarani and Liu [16]	✓			Both
		Motoyama and Varghese [17]	✓			Individual
		Zheng, Li, Chen, and Huang [18]		✓		Forum
		Hao, Zhou, Cheng, Huang, and Wu [19]	✓			None
	Multiple types	Jain, Kumaraguru, and Joshi [20]		✓	✓	Individual
		Jain and Kumaraguru [21]		✓	✓	Individual
		Zhang, Wang, Li, and Xiao [22]	✓		✓	Individual
		Nie et al. [23]		✓	✓	Individual
		Goga et al. [24]	✓	✓		Individual
		Xing, Deng, Wu, Xie, and Gao [25]	✓	✓		Individual
		Zhong, Cao, Guo, and Nie [26]	✓		✓	Individual

Table 1 (continued)

Site number	Attribute type number	Literature	Profile	Content	Social network	Individual or forum
Single site	Single type	Zhou et al. [27]			✓	Individual
		Liu, Wu, Han, and Zhou [28]		✓		Forum
		Li, Zhou, Han, and Hu [29]	✓			Individual
		Li, Yuan, Zhou, Wang, and Hu [30]	✓			Individual
		Maity et al. [31]	✓			Individual
		Almishari and Tsudik [32]		✓		None
		Yamak et al. [33]	✓			Individual
		Tsikerdekis et al.[34]	✓			Individual
		Solorio et al. [35]		✓		Individual
	Multiple types	Yu et al. [36]	✓	✓		Individual

past studies usually used one or two types of attributes; SiMAIM uses three types of attributes to deliver more robust identification.

3 SiMAIM framework

Figure 2 shows the SiMAIM framework. SiMAIM consists of four modules: Attribute Extraction Module, Data Construction Module, Model Construction Module, and Sockpuppet Identification Module. The Attribute Extraction Module retrieves attributes concerning the profile, text content, and social network of accounts. The samples for constructing classification models are formed in the Data Construction Module. For example, the two accounts created by user1, the two created by user2, and the account created by user3 are used to construct the ten samples in the figure. The ten samples involve two positive samples (samples formed by accounts of the same user) and eight negative samples (samples formed by accounts of different users). Classification models are then built in the Model Construction Module with the constructed samples. The Model Construction Module is independent of classification algorithms; therefore, it can adopt any classification algorithms in practice.

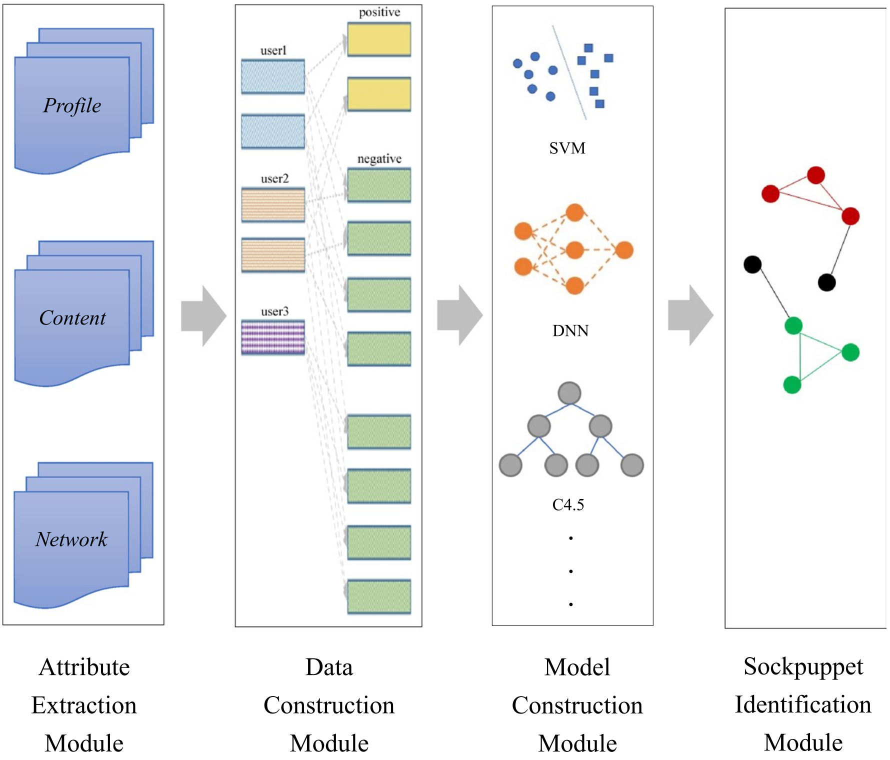


Fig. 2 SiMAIM framework

We used the C4.5 decision tree, random forest, naïve Bayes classifier, deep neural network, and SVM in the experiments. Finally, the Sockpuppet Identification Module carries out the identification of sockpuppets and puppetmasters. This module generates a network where a node denotes an account, and two accounts are linked if they are identified as sockpuppets by the classification models. Accounts consisting of a maximal clique are used to identify a puppetmaster. For instance, there are two maximal cliques, denoted by red nodes and green nodes, respectively, in the figure. We elaborate on these four modules in Sects. 3.2, 3.3, 3.4, and 3.5, while Sect. 3.1 presents definitions.

3.1 Definitions

Definition 1 An *account* is an identity on a social media site. A *user* is a real-world entity, e.g., a person or a company, who creates accounts. A *sockpuppet* is one of the accounts created by a user. The user who creates sockpuppets is a *puppetmaster*.

Definition 2 A *forum* on a social media site is a collection of conversations about a topic.

Definition 3 A *thread* is a conversation in a forum.

Definition 4 A *post* is an initial text that starts a thread, and a *reply* is the following text.

3.2 Attribute extraction module

The Attribute Extraction Module utilizes the profile, text content, and social network of accounts to generate attributes. We took the distribution of an account's posts and replies over a set of forum topics as the profile-based attribute. The similarity of the two attribute vectors represented the degree of similarity in the profile between the two accounts.

For generating the content-based attributes, we first tokenized all the posts and replies and performed linguistic processing to get a keyword set. Then, we calculated the TF-IDF [37] of each keyword in each post and each reply. The TF-IDF distributions are among the content-based attributes. Furthermore, the topic model Latent Dirichlet Allocation (LDA) [38] was used to obtain the distribution of all of an account's posts and replies on hidden topics, another content-based attribute to measure similarity in the content between accounts.

The accounts interact by replying to posts, forming a social network between the accounts. We proposed six social-network-based attributes to evaluate the similarity of social relationships between accounts.

We describe the profile-based, content-based, and social-network-based attributes in Sects. 3.2.1, 3.2.2, and 3.2.3, respectively.

3.2.1 Profile-based attribute

According to our observation, basic information, such as account name, registration date, login date, and login IP, is ineffective in identifying an account on a single forum-oriented social media site. Instead, an account's posts and replies usually focus on only a few topics. The topics an account is interested in are a characteristic of the account. We utilized the topic distribution of an account's posts and replies to construct the profile-based attribute. On a forum-oriented social media site, various forums exist. Each forum sets a topic. Suppose there are N forums on a site. The topic distribution is represented by a vector of dimension N. The ith element in the vector is the number of the account's posts and replies in the ith forum divided by the number of the account's total posts and replies. This vector is called the *topic distribution vector* (A_{v_Topics}), the profile-based attribute. For instance, there are 20 main forums on Mobile01.com, such as mobile phones, cameras, notebooks, and computers. The A_{v_Topics} of an account consists of 20 values (elements), each corresponding to the number of the account's posts and replies in a main forum divided by the number of the account's total posts and replies in all main forums.

If the topic distribution vectors of two accounts are similar, the probability that a puppetmaster creates the two accounts is relatively high. The cosine similarity measures the similarity of two topic distribution vectors. The cosine similarity of two vectors $\vec{q}$ and $\vec{d}$ is defined in Eq. 1.

$$\cos\left(\vec{q}, \vec{d}\right) = \frac{\vec{q} \cdot \vec{d}}{|\vec{q}|\left|\vec{d}\right|} \tag{1}$$

3.2.2 Content-based attributes

We used all accounts' posts and replies to generate three content-based attributes. Initially, all the collected posts and replies had to undergo tokenization and linguistic processing (stop words removal, normalization, lemmatization, and stemming) to generate a keyword set KS. Next, each account generates two documents. The first, *title assembled document*, consists of only the titles of the posts and replies published by the account. The second, *article assembled document*, consists of not only the titles but the text of the posts and replies published by the account. The *title assembled document* is a concise presentation of the subjects and events the account was interested in, and the *article assembled document* adds the details. Then, each document's TF-IDF of each keyword in KS was calculated. The two sets of TF-IDF reveal the keyword distribution of the posts and replies the account has published. The first set of TF-IDF is called *title TF-IDF vector* ($A_{v\_Title\_TF\text{-}IDF}$), and the second is *article TF-IDF vector* ($A_{v\_Article\_TF\text{-}IDF}$). These two vectors are the two content-based attributes.

The *topic distribution vector* defined in the previous section is generated with the topics specified by the social media sites. However, posts and replies may include more topics except the ones specified by social media sites. To further explore the

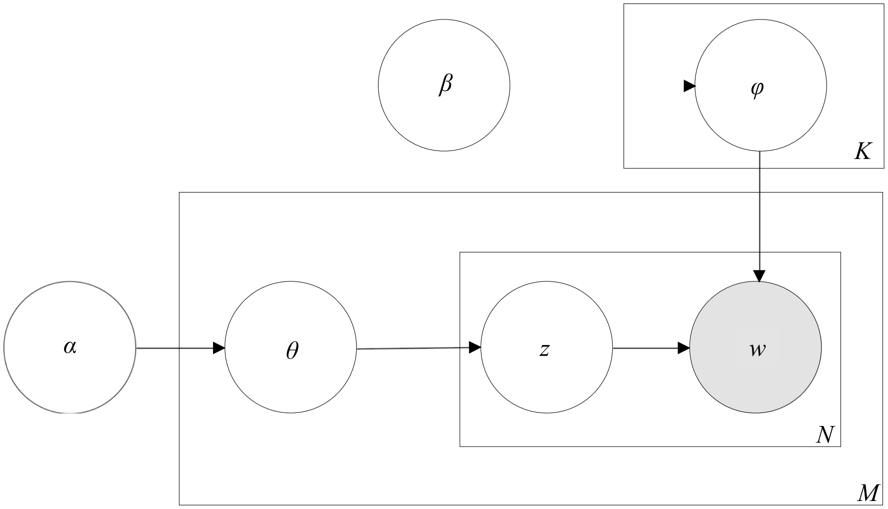


Fig. 3 Latent Dirichlet allocation [38]

hidden topics, Latent Dirichlet Allocation (LDA) [38] was utilized to retrieve the hidden topics. LDA is a topic model that identifies the hidden topics in the documents and retrieves the probability distribution over the hidden topics for each document. Figure 3 shows a schematic representation of the LDA model. M denotes the number of documents; K is the number of topics; N_i is the number of words in document i; and α and β are the parameters of the Dirichlet probability distribution. Furthermore, θ_i represents the topic distribution of document i; φ_k represents the word distribution of topic k; z_{ij} is the topic of the jth word in document i; and w_{ij} is the jth word in document i. The joint probability distribution is given in Eq. 2. SiMAIM first compiles the article assembled document of each account. The article assembled documents of all accounts are the input to the LDA, and M is the number of the article assembled documents, which equals the number of accounts. K is the number of hidden topics retrieved after solving the LDA. We also get θ_i, the probability distribution over the hidden topics for the article assembled document of account i. θ_i forms a *hidden topic distribution vector* ($A_{v\_Hidden\_Topics}$), the third content-based attribute. Each account gets its own hidden topic distribution vector. The similarities between two accounts on these three content-based attributes are also calculated using the cosine similarity.

$$\mathrm{p}\left(w_i, z_i, \theta_i, \varphi | \alpha, \beta\right) = \prod_{j=1}^{N} p(\theta_i|\alpha)p\left(z_{ij}|\theta_i\right)p(\varphi|\beta)p\left(w_{ij}|\varphi_{z_{ij}}\right) \tag{2}$$

3.2.3 Social network-based attributes

In social network-based identification, two accounts are considered to have a social interaction if the two accounts are involved in the same thread by publishing or

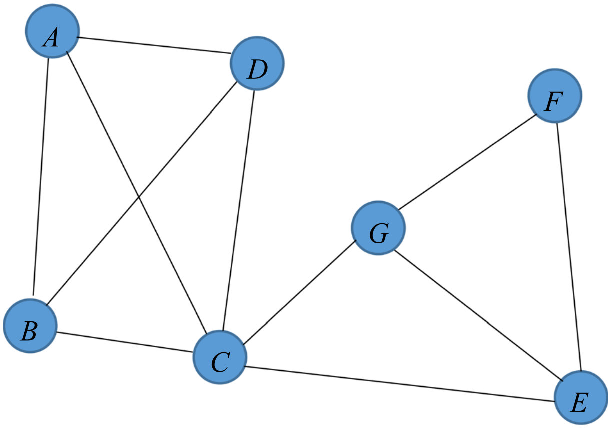


Fig. 4 A social network

replying to the post. In social networks, we treat each account as a node. An edge exists between the two nodes if social interactions between two accounts exist. For example, Fig. 4 shows seven accounts. Among them, accounts A, B, C, and D have social interactions with each other, which suggests that the four accounts mutually reply to at least one discussion string. Accounts C, E, and G also interact with each other. Accounts E, F, and G interact with each other. However, accounts C, E, F, and G do not have social interaction with each other, as the interaction between C and F does not exist.

We propose six social network-based attributes to represent the social relationship between accounts. The first attribute is *interactive presence attribute* ($A_{Interaction}$), a binary attribute that represents whether social interactions exist between two accounts. $A_{Interaction}$ is set to 1 if the two accounts have social interactions and 0 otherwise.

The second attribute is *interactive frequency attribute* ($A_{Frequency}$), which is the number of threads involving both accounts. The higher the $A_{Frequency}$, the larger the interactions between the two accounts. The third attribute, *interactive intimacy attribute* ($A_{Intimacy}$), is proposed to explore the interactions between the two accounts further. For each thread where both accounts are involved, the number of posts and replies published by either account in this thread is derived. Suppose two accounts appear in n threads simultaneously, i.e., $A_{Frequency}$ is n. Then, $A_{Intimacy}$ can be calculated using Eq. 3.

$$A_{Intimacy} = \frac{\sum_{i=1}^{n} k_i}{n} \tag{3}$$

Here, k_i denotes the number of posts and replies the two accounts published in the ith thread. A high $A_{Intimacy}$ value indicates that the two accounts interact frequently.

Suppose that m accounts exist. An account Acc has three numeric vectors, each dimension m. The first vector consists of the values of $A_{Interaction}$ of Acc to all m accounts. The second and the third vectors consist of the values of $A_{Frequency}$ and $A_{Intimacy}$ of account Acc to all m accounts, respectively. The three vectors are named *interactive presence vector* ($A_{v\_Interaction}$), *interactive frequency vector* ($A_{v\_Frequency}$), and *interactive intimacy vector* ($A_{v\_Intimacy}$), respectively. $A_{v\_Interaction}$, $A_{v\_Frequency}$, and $A_{v\_Intimacy}$ are the other three social network-based attributes. For evaluating the similarity between two accounts, $A_{Interaction}$, $A_{Frequency}$, and $A_{Intimacy}$ measure the direct interaction between the two accounts, while $A_{v\_Interaction}$, $A_{v\_Frequency}$, and $A_{v\_Intimacy}$ consider the interaction with all of the other accounts.

The above-mentioned ten attributes are listed in Table 2.

The similarities of the interactive presence attribute, interactive frequency attribute, and interactive intimacy attribute between the two accounts are the values of the attributes. The similarities of the interactive presence vector, interactive frequency vector, and interactive intimacy vector between the two accounts are calculated using cosine similarity.

3.3 Data construction module

Identifying sockpuppets is essentially a classification problem, which determines whether the same user creates two or more accounts. Samples for training a model and testing were required to perform a classification task. A sample was formed by the ten attributes' similarities between a pair of accounts. Note that the positive class indicates that the two accounts are sockpuppets, and the negative class is otherwise. Each sample has a class label. In detail, a pair of accounts the same user creates forms a positive sample, otherwise a negative sample. Figure 5 illustrates the

Table 2 Attributes

Attribute	Code	Description
Topic distribution vector	A_{v_Topics}	Distribution over the topics specified by the social media site
Title TF-IDF vector	$A_{v\_Title\_TF\text{-}IDF}$	Keyword distribution of the titles
Article TF-IDF vector	$A_{v\_Article\_TF\text{-}IDF}$	Keyword distribution of the posts and replies
Hidden topic distribution vector	$A_{v\_Hidden\_Topics}$	Distribution over the hidden topics
Interactive presence attribute	$A_{Interaction}$	Indicator of whether social interactions exist between two accounts
Interactive frequency attribute	$A_{Frequency}$	Number of threads where two accounts are involved
Interactive intimacy attribute	$A_{Intimacy}$	The ratio of the number of posts and replies to $A_{Frequency}$
Interactive presence vector	$A_{v\_Interaction}$	Set of $A_{Interaction}$ between an account and the other accounts
Interactive frequency vector	$A_{v\_Frequency}$	Set of $A_{\mathrm{Frequency}}$ between an account and the other accounts
Interactive intimacy vector	$A_{v\_Intimacy}$	Set of $A_{Intimacy}$ between an account and the other accounts

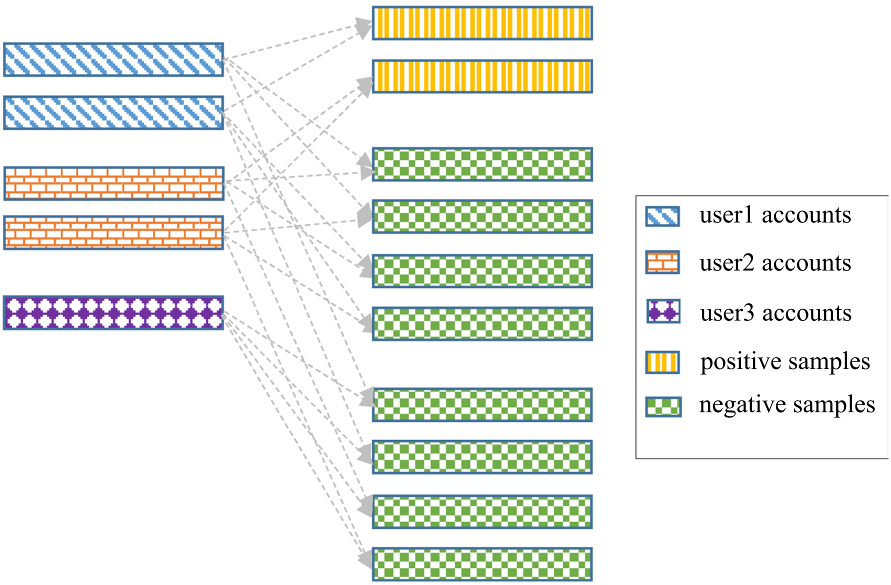


Fig. 5 Construction of samples

construction of samples. The two accounts created by user1 form the first positive sample, and the two accounts created by user2 form the second positive sample. A pair of accounts of different users forms a negative sample.

In the experiments, samples are further divided into training and test samples to evaluate the performance of SiMAIM. Classification models are constructed using training samples. Then, the classification models classify the test samples. Section 4.1 gives details about the training samples and the test samples used in the experiments. In the real world, social media sites can use previous data to build training samples and construct classification models. Then identify sockpuppets and puppetmasters in the upcoming data using the classification models.

3.4 Model construction module

In Model Construction Module, classification models are constructed using the training samples. As sockpuppet identification is independent of the classification algorithms, multiple classification algorithms are evaluated. We used five state-of-the-art classification algorithms in the experiments: the C4.5 decision tree, random forest, naïve Bayes classifier, deep neural network, and SVM. We will discuss each algorithm in brief next.

3.4.1 C4.5 decision tree (C4.5)

A decision tree is constructed in a top–down, iterative manner. In the C4.5 algorithm, an attribute that results in maximum information gain is selected as a tree node in each iteration. A leaf node is determined under certain conditions such that all training samples in the node belong to the same class. A pruning procedure can prune the tree to avoid overfitting.

3.4.2 Random forest (RF)

The random forest implements collaborative decision-making. Multiple decision trees are constructed to provide multiple intermediate decisions. The final decision is made by aggregating the intermediate decisions. The construction of multiple decision trees can use subsets of attributes.

3.4.3 Naïve Bayes classifier (NBC)

The naïve Bayes classifier is a popular classification algorithm that uses Bayes' theorem to determine the attribution of sample *x,* whose class is unknown. The predicted class of sample *x* is the class with the highest probability $p(y|x)$, where *y* indicates a class. The naïve Bayes classifier assumes that the attributes are independent of each other. This assumption accelerates the classification speed but may lower the classification accuracy.

3.4.4 Deep neural network (DNN)

In the last decade, deep learning techniques have been recognized as powerful learning models in many areas. These techniques are based on the neural network. A neural network is essentially a nonlinear regression. Between input and output variables, additional hidden variables, which form hidden layers, are added to model the complexity of the data distribution. A neural network with a hidden layer is a vanilla neural network. As only one hidden layer might not precisely capture the complexity of the data, we used a deep neural network, i.e., a neural network with more than one hidden layer.

3.4.5 Support vector machine (SVM)

SVM finds a hyperplane that separates the samples of different classes. The model construction is performed to maximize the distance between the training samples and the hyperplane. Classification is based on a test sample's position to the hyperplane. The kernel trick is used to find a nonlinear hyperplane. Support vector regression is a regression variant of the SVM.

3.5 Sockpuppet identification module

Classification models are constructed in the Model Construction Module. The Sockpuppet Identification Module utilizes the outcomes of the classification models to identify sockpuppets and puppetmasters. To do so, Sockpuppet Identification Module produces a "*sockpuppet identification network*." In the sockpuppet identification network, each account is a node. If the classification model determines that two accounts are sockpuppets created by a puppetmaster, there is a link between the corresponding two nodes. To find all the sockpuppets of a puppetmaster in the sockpuppet identification network, the Bron–Kerbosch algorithm [39] was adopted to find the maximal cliques in the network. A clique is a complete sub-network in a network, and a maximal clique is a clique that includes the largest possible number of nodes. The nodes of a maximal clique in the sockpuppet identification network stand for the sockpuppets of a puppetmaster. Note that more than one maximal clique may exist in the sockpuppet identification network. Each maximal clique indicates a puppetmaster.

4 Experiment results

We present the experiment results in this section. Section 4.1 explains the data collection and experiment setups. Section 4.2 introduces the measures for evaluating performance and a baseline method. Section 4.3 shows the results derived from various setups. Section 4.4 presents an ablation study, and Sect. 4.5 shows performance comparisons between SiMAIM and existing methods.

4.1 Data collection

This study retrieved texts and account information from 300 accounts on Taiwan's most famous forum-oriented social media site Mobile01. 150 out of the 300 accounts were sockpuppets suspended by the Mobile01 system administrator for malicious behaviors (attack on other accounts or specific products, self-advertising, causing controversy, etc.). Another 150 were benign accounts randomly selected. The sockpuppets were identified by the Mobile01 administrator with manual inspection or reported by other accounts, which was usually time-consuming. Forty-seven puppetmasters created the 150 sockpuppets. On average, a puppetmaster registered about three sockpuppets. The data collected included the basic information of each account, the content of all the posts and replies (title, content, and post time), and the relationship between the posts and the replies among accounts to infer the social network structure among the accounts.

After retrieving the data from Mobile01, a set of samples was constructed using the procedure described in Sect. 3.3. To evaluate the performance of SiMAIM, we compiled three datasets containing different numbers of training and test samples.

The collected data were inherently temporal. The initial posts and the following replies had timestamps. A cutting point in time, i.e., target time, was selected to compile a dataset. The sockpuppets suspended before the target time were combined with the same number (as the sockpuppets) of randomly selected benign accounts created before the target time to generate the training samples. The remaining accounts were used to generate the test samples. In this manner, the experiments simulated the real world, where site administrators use previous data to generate training samples and then construct classification models for identifying future sockpuppets and puppetmasters. The first dataset used 38 sockpuppets and 38 benign accounts to generate the training samples. In the second dataset, the training samples were generated by 78 sockpuppets and 78 benign accounts. In the third dataset, 115 sockpuppets and 115 benign accounts were used to generate the training samples. These three datasets differed in the size of training samples for examining SiMAIM's identification ability with different sizes of training samples.

4.2 Measures and the baseline method

We used recall, precision, and F1 score to evaluate the performance of SiMAIM. Let a maximal clique have n nodes (accounts) and the corresponding puppetmaster creates m sockpuppets; suppose that the intersection of the n accounts and the m sockpuppets contains s accounts. If both $\frac{s}{n}$ and $\frac{s}{m}$ are larger than or equal to a threshold δ, the puppetmaster is effectively identified. The recall is the number of the identified puppetmasters divided by the number of puppetmasters in the test samples. The precision is the number of identified puppetmasters divided by the number of maximal cliques retrieved from the sockpuppet identification network. The F1 score can be calculated using Eq. 4.

$$\mathrm{F1} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{4}$$

To show the effectiveness of classification algorithms, we used a baseline method that linearly aggregated the similarities derived from two accounts. For two accounts, S and T, the baseline method summed the ten attributes' similarities. The sum was denoted as $AttSum(S, T)$, as shown in Eq. 5, where cos(.,.) denotes the cosine similarity and $A_{Interaction}(S, T)$, $A_{Frequency}(S, T)$, and $A_{Intimacy}(S, T)$ are the values of attributes $A_{Interaction}$, $A_{Frequency}$, and $A_{Intimacy}$ between S and T, respectively. The higher the $AttSum(S, T)$ was, the higher the possibility that the same user created S and T.

$$\begin{aligned}
AttSim(S,T) = {} & \cos\left(S_{A_{v\_\text{Topics}}}, T_{A_{v\_\text{Topics}}}\right) + \cos\left(S_{A_{v\_\text{Title\_TF-IDF}}}, T_{A_{v\_\text{Title\_TF-IDF}}}\right) \\
& + \cos\left(S_{A_{v\_\text{Article\_TF-IDF}}}, T_{A_{v\_\text{Article\_TF-IDF}}}\right) \\
& + \cos\left(S_{A_{v\_\text{Hidden\_Topics}}}, T_{A_{v\_\text{Hidden\_Topics}}}\right) + A_{Interaction}(S,T) \\
& + A_{Frequency}(S,T) + A_{Intimacy}(S,T) \\
& + \cos\left(S_{A_{v\_\text{Interaction}}}, T_{A_{v\_\text{Interaction}}}\right) \\
& + \cos\left(S_{A_{v\_\text{Frequency}}}, T_{A_{v\_\text{Frequency}}}\right) + \cos\left(S_{A_{v\_\text{Intimacy}}}, T_{A_{v\_\text{Intimacy}}}\right)
\end{aligned} \tag{5}$$

For account S, the baseline method identified the top-k accounts with the highest AttSims with S. Then, S had links to these k accounts in the sockpuppet identification network. The identification of sockpuppets proceeded as discussed in Sect. 3.5.

Table 3 Recalls of the baseline method on the first dataset

Top-k	δ				
	0.1	0.2	0.3	0.4	0.5
10	1	1	0.904762	0.642857	0.619048
20	1	1	0.976190	0.595238	0.571429
30	1	1	0.976190	0.5	0.404762
40	1	1	0.904762	0.5	0.380952
50	1	1	0.880952	0.5	0.285714

Table 4 Recalls of SiMAIM with different algorithms on the first dataset

Algorithms	δ				
	0.1	0.2	0.3	0.4	0.5
NBC	0.928571	0.928571	0.857143	0.619048	0.547619
RF	1	1	0.952381	0.809524	0.785714
C4.5	1	1	0.952381	0.857143	0.833333
DNN	1	1	0.97619	0.809524	0.809524
SVM	1	1	0.97619	0.880952	0.809524

Table 5 Precisions of the baseline method on the first dataset

Top-k	δ				
	0.1	0.2	0.3	0.4	0.5
10	0.691462	0.691462	0.480595	0.128719	0.119017
20	0.781939	0.771369	0.319623	0.073759	0.046645
30	0.858818	0.675291	0.135505	0.05463	0.014891
40	0.907411	0.433605	0.074507	0.026194	0.006254
50	0.889885	0.221423	0.034048	0.00691	0.001438

Table 6 Precisions of SiMAIM with different algorithms on the first dataset

Algorithms	δ				
	0.1	0.2	0.3	0.4	0.5
NBC	0.575693	0.439232	0.255864	0.117271	0.072495
RF	0.83737	0.83737	0.719723	0.470588	0.460208
C4.5	0.848921	0.848921	0.717986	0.336691	0.316547
DNN	0.898477	0.888325	0.659898	0.239848	0.21066
SVM	0.631387	0.631387	0.452555	0.288321	0.237226

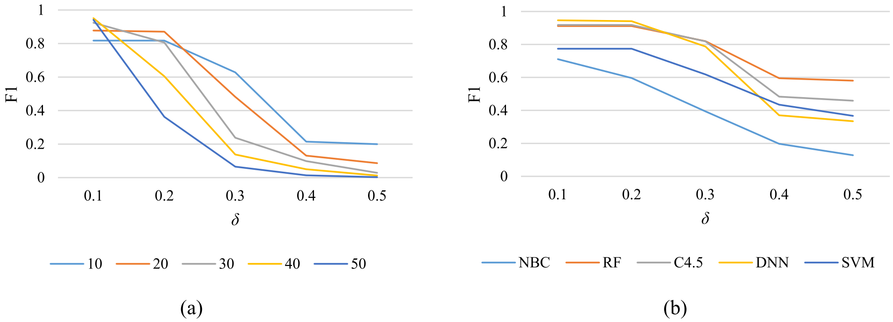


Fig. 6 F1 scores on the first dataset: **a** baseline method; **b** SiMAIM with different algorithms

Table 7 Recalls of the baseline method on the second dataset

Top-k	δ				
	0.1	0.2	0.3	0.4	0.5
10	1	1	0.969697	0.848485	0.848485
20	1	1	0.969697	0.636364	0.606061
30	1	1	0.969697	0.393939	0.363636
40	1	1	0.939394	0.424242	0.30303
50	1	1	0.818182	0.393939	0.30303

4.3 Experiment results

All of the experiments were performed on an IBM-compatible PC with an Intel Core i5 4570CPU (3.2 GHz) and 16-GB main memory, running on Windows 10 Professional 64-Bit. The algorithms were implemented using Python, Ruby, and Weka. Sections 4.3.1, 4.3.2, and 4.3.3 present the experiment results derived from the three datasets. Section 4.3.4 presents a discussion.

Table 8 Recalls of SiMAIM with different algorithms on the second dataset

Algorithms	δ				
	0.1	0.2	0.3	0.4	0.5
NBC	0.878788	0.878788	0.757576	0.545455	0.484848
RF	0.818182	0.818182	0.787879	0.69697	0.666667
C4.5	0.969697	0.969697	0.939394	0.878788	0.848485
DNN	0.969697	0.969697	0.969697	0.848485	0.818182
SVM	1	1	1	0.909091	0.878788

4.3.1 Experiment results of the first dataset

In the first dataset, 38 sockpuppets and 38 benign accounts were used to generate the training samples. This dataset contained the fewest training samples, providing the least decision-making material. Tables 3 and 4 present the recalls derived by the baseline method and SiMAIM with different classification algorithms, respectively. The recalls remained at a high level when δ was low. As δ increased, the recalls of the baseline method decreased, while the recalls of the algorithms maintained a high level except for NBC. Among the algorithms, the SVM outperformed the others. NBC performed the worst. The precisions presented similar statistics shown in Tables 5 and 6. The baseline method performed well only when δ was 0.1. The algorithms except the NBC achieved better precisions at any δ.

Figure 6a, b shows the F1 scores derived from the baseline method and algorithms. The baseline method of k equal to 10 had better F1 scores than the F1 scores of the other settings with different k. However, the algorithms performed

Table 9 Precisions of the baseline method on the second dataset

Top-k	δ				
	0.1	0.2	0.3	0.4	0.5
10	0.603561	0.60075	0.49672	0.209934	0.186504
20	0.80286	0.766898	0.437175	0.095321	0.067591
30	0.936499	0.702473	0.201187	0.052423	0.026508
40	0.984375	0.599071	0.098818	0.025422	0.006166
50	0.9956	0.396738	0.054129	0.012507	0.001687

Table 10 Precisions of SiMAIM with different algorithms on the second dataset

Algorithms	δ				
	0.1	0.2	0.3	0.4	0.5
NBC	0.518987	0.386076	0.246835	0.183544	0.158228
RF	0.842975	0.842975	0.760331	0.561983	0.545455
C4.5	0.774306	0.774306	0.684028	0.454861	0.451389
DNN	0.944762	0.775238	0.419048	0.205714	0.167619
SVM	0.5	0.5	0.440678	0.338983	0.338983

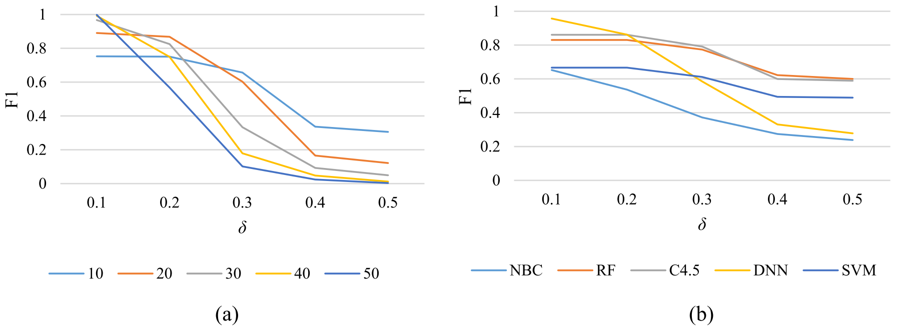


Fig. 7 F1 scores on the second dataset: **a** baseline method; **b** SiMAIM with different algorithms

Table 11 Recalls of the baseline method on the third dataset

Top-k	δ				
	0.1	0.2	0.3	0.4	0.5
10	1	0.933333	0.933333	0.733333	0.733333
20	1	1	0.933333	0.6	0.533333
30	1	1	0.8	0.6	0.4
40	1	1	0.6	0.4	0.2
50	1	0.733333	0.266667	0.066667	0

Table 12 Recalls of SiMAIM with different algorithms on the third dataset

Algorithms	δ				
	0.1	0.2	0.3	0.4	0.5
NBC	0.733333	0.733333	0.733333	0.6	0.533333
RF	1	1	1	1	1
C4.5	1	1	1	1	1
DNN	1	1	1	1	0.933333
SVM	1	1	1	1	0.933333

considerably better than the baseline method, particularly at a high δ. RF outperformed the other algorithms. C4.5, SVM, and DNN performed next to RF. C4.5, SVM, and DNN had a slightly higher recall but considerably lower precision than RF, leading to the lower F1 scores of C4.5, SVM, and DNN. NBC was not competitive at all, which might be attributed to its naïve presumption, i.e., all attributes are independent. The ten attributes used in SiMAIM were obviously not independent of each other.

Table 13 Precisions of the baseline method on the third dataset

Top-k	δ				
	0.1	0.2	0.3	0.4	0.5
10	0.551064	0.540426	0.465957	0.170213	0.170213
20	0.875	0.749031	0.347868	0.119186	0.057171
30	0.982967	0.709099	0.188705	0.083371	0.027342
40	0.999004	0.548188	0.071884	0.016129	0.003186
50	1	0.118597	0.002536	0.000085	0

Table 14 Precisions of SiMAIM with different algorithms on the third dataset

Algorithms	δ				
	0.1	0.2	0.3	0.4	0.5
NBC	0.487179	0.487179	0.435897	0.307692	0.307692
RF	0.857143	0.857143	0.8	0.8	0.8
C4.5	0.757143	0.757143	0.671429	0.585714	0.585714
DNN	0.960227	0.960227	0.522727	0.301136	0.215909
SVM	0.481481	0.481481	0.481481	0.462963	0.462963

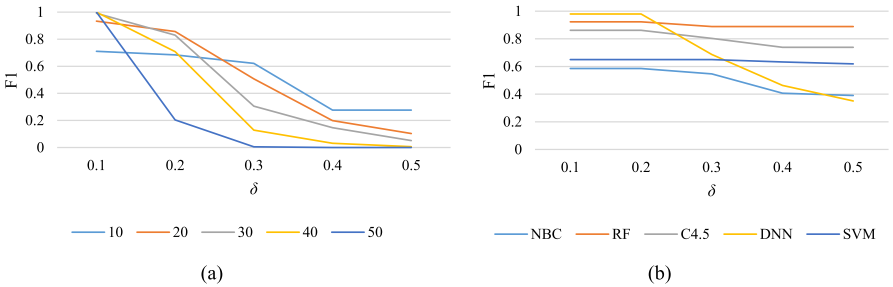


Fig. 8 F1 scores on the third dataset: **a** baseline method; **b** SiMAIM with different algorithms

4.3.2 Experiment results of the second dataset

In the second dataset, the training samples were generated by 78 sockpuppets and 78 benign accounts. This dataset evaluated SiMAIM's performance with more training material. Tables 7 and 8 present the recalls derived from the baseline method and different classification algorithms. SVM excelled at recalling the puppetmasters irrespective of the δ value. DNN, C4.5, and the baseline method with a k value equal to 10 also performed well regarding the recall. Regarding precisions presented in Tables 9 and 10, RF outperformed the other algorithms and the baseline method when δ increased gradually. C4.5 performed next to RF, while SVM and DNN had worse precisions at a high δ. Figure 7a, b shows the F1 scores derived from the baseline method and algorithms. Overall, RF and C4.5 exhibited the best performance. The baseline method had a high F1 score at a low δ; however, the F1 score declined significantly at a higher δ. Compared with the previous experiment, including a larger number of training samples helped achieve higher F1 scores.

4.3.3 Experiment results of the third dataset

In the third dataset, 115 sockpuppets and 115 benign accounts were used to generate the training samples. More training material was provided in this dataset. Tables 11 and 12 present the recalls derived from the baseline method and different classification algorithms. Because of the large number of training samples, most of the recalls derived by the baseline method and algorithms were high. However, when δ was high, the algorithms except NBC maintained perfect performance, but the baseline method's performance deteriorated significantly. As for precisions, shown in Tables 13 and 14, the baseline method's performance considerably worsened at a high δ. Instead, the performance of the considered algorithms was more stable than that of the baseline method. Note that RF performed consistently at every δ. Compared with the baseline method, all algorithms achieved better precision. Figure 8a, b shows the F1 scores of the baseline method and the algorithms. The F1 scores derived by the RF were as high as 0.9 at different δ. RF effectively identified the puppetmasters. The F1 scores in this experiment were significantly higher than those in the previous two experiments, which could be attributed to the inclusion of more training samples.

4.3.4 Discussion

The SiMAIM framework retrieved three types of attributes and used classification algorithms to identify puppetmasters. Compared with the baseline method, adopting classification algorithms led to relatively high recall, precision, and F1 score. The baseline method, which linearly aggregated the retrieved attributes, usually performed well at a low δ. However, the baseline method's performance declined significantly when δ increased. A high δ implied a stricter identification. Therefore, the baseline method was not a practical solution. In contrast, SiMAIM used classification algorithms to identify puppetmasters more effectively. In particular, at a high δ, all algorithms except NBC achieved considerably better identification results than the baseline method. At the same time, NBC's performance was comparable to the baseline method's performance.

If the social media sites' administrators desire to identify as many puppetmasters as possible, C4.5, DNN, and SVM algorithms are suggested because these algorithms deliver excellent recalls. The performance of RF was slightly worse than that of the above-mentioned three algorithms. When the number of training samples was large, e.g., the third dataset, the algorithms achieved a very high recall. The sites' administrators can keep their sites away from malicious behaviors by identifying many puppetmasters. If the precision of identification is a priority, RF and C4.5 are the better choices. Although DNN performs excellently at a low δ, its performance

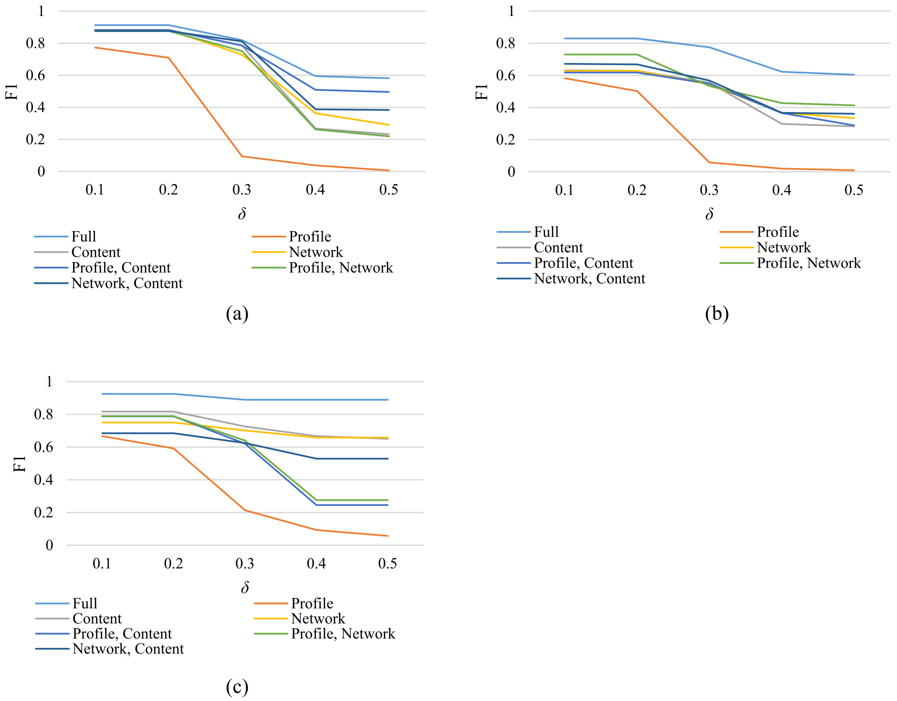


Fig. 9 F1 scores of ablation study: **a** first dataset; **b** second dataset; **c** third dataset

becomes lower than RF and C4.5 when δ is larger than 0.2. A high-precision identification prevents false alarms, which usually increases users' satisfaction. Finally, if the sites' administrators search for a balance between recall and precision, RF and C4.5 are still the better choices for their excellent F1 scores.

The greatest number of sockpuppets created by a puppetmaster is 6 in the collected datasets. SiMAIM recognizes the puppetmaster by identifying 5 out of the six sockpuppets because the five sockpuppets are in a maximal clique. All six sockpuppets discuss telecom companies very much. The sockpuppet not being identified mainly focuses on a single telecom company, while other sockpuppets concern more telecom companies. The sockpuppet not being identified also has fewer interactions with other sockpuppets. However, the puppetmaster is still recognized by identifying its five sockpuppets.

4.4 Ablation study

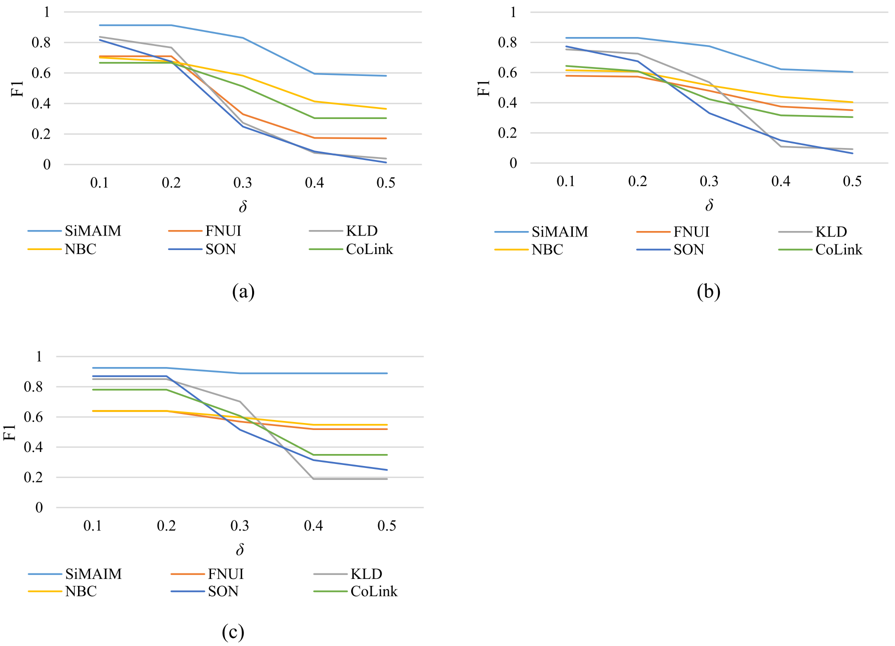
SiMAIM adopts three types of attributes, i.e., profile-based, content-based, and social network-based. The ablation study examined the performance of different combinations of attributes. We used RF algorithm to construct models. Figure 9 presents F1 scores obtained from different combinations on the three datasets. Using only profile-based attribute performed worst on the three datasets. This performance results from the limited profile. Using content-based attributes or social network-based attributes achieved better performance. On average, using social network-based attributes performed better than using content-based attributes. The results may come from the fact that sockpuppets usually deliberately adopt different writing styles writing style (vocabulary, syntax, structure, etc.) but involve the same conversion threads. Combining profile-based attribute with one of the other two types of attributes improved performance. The combination of content-based attributes and social network-based attributes delivered good and stable performance. Finally, using the complete set of attributes got the best performance.

4.5 A comparison with existing methods

We compared SiMAIM with four existing methods: FNUI [13], CoLink [26], SON [28], and Almishari & Tsudik [32].

FNUI constructs friendship networks of accounts from different social media sites. The friendship network of an account consists of the account's n-hop friends. Two accounts from different sites with similar structures of friendship networks belong to a user. FNUI uses social network-based attributes. In the experiment, we treated two accounts with social interaction as friends to construct the friendship network of each account.

CoLink adopts semi-supervised learning to identify a user's sockpuppets across social media sites. Accounts' profiles and relationships are retrieved, and SVM is



(a)

(b)

(c)

Fig. 10 F1 scores of existing methods: **a** first dataset; **b** second dataset; **c** third dataset

used in the semi-supervised learning process. Therefore, CoLink uses profile-based attributes and social network-based attributes. In our experiment, the social interactions between accounts form accounts' relationships.

SON thinks that a user creates two accounts from different social media sites if both accounts share similar sentiment orientation to topics based on emotional phrases extracted from posted comments, i.e., content-based attributes. A similar-orientation network is constructed by connecting accounts (nodes) with similar sentiment orientation. Then, community detection algorithms find sockpuppets in the similar-orientation network. SON takes social network-based attributes, too. In our experiment, we used HowNet to extract sentiment words and cosine similarity to calculate sentiment orientation similarity between two accounts.

Almishari and Tsudik extracted unigrams, diagrams, review ratings, and review categories from each review, i.e., content-based attributes. They adopted the naïve Bayes classifier (NBC) and symmetric Kullback-Leibler Divergence (KLD) to link an anonymous review to an account on Yelp. In the experiments, we performed account-to-account matches by using an account's posts and replies to generate the account's NBC model and KLD model.

Figure 10 presents the F1 scores from the compared methods on the three datasets. In the figure, NBC and KLD stand for Almishari and Tsudik's method with NBC and KLD, respectively. The F1 scores of RF algorithm are reported in the figure as SiMAIM's statistics. SiMAIM outperformed the existing methods, particularly at high δ. NBC, CoLink, and FNUI were next to SiMAIM. However, their F1 scores are under 0.4 on the first and second datasets, which is ineffective. KLD and SON did not perform well at high δ. The results show that the Kullback–Leibler Divergence may miss the similarity of accounts, and the sentiment orientation of text may be hard to interpret.

5 The conclusion and future work

In this paper, we proposed the SiMAIM framework for identifying sockpuppets and puppetmasters on a single forum-oriented social media site. SiMAIM consists of four modules: Attribute Extraction Module, Data Construction Module, Model Construction Module, and Sockpuppet Identification Module. Accounts' profiles, text contents, and social networks between accounts were used to generate attributes for building classification models to identify sockpuppets. Then, the sockpuppet identification network was constructed to identify the puppetmasters. Considering the nature of the forum-oriented social media sites, the proposed framework used the topic distribution of an account as the profile-based attribute. In addition to the topics specified by the social media sites, SiMAIM retrieved hidden topics from the text contents. The keyword distribution of the text contents generated content-based attributes as well. The involvement in the same thread was used to generate social network-based attributes. Classification models identified sockpuppets and then formed the sockpuppet identification network. The maximal clique in the network indicated a puppetmaster. The experiment results showed SiMAIM's effective identification.

In general, C4.5, DNN, SVM, and RF algorithms achieved good recall, while RF and C4.5 achieved good precision. RF and C4.5 were also good for F1 scores. The results of the ablation study showed that the social network-based attributes performed better than the content-based attributes, and the profile-based attribute performed worst. The combinations of different types of attributes improved performance. Using all three types of attributes delivered the best performance. Compared with four existing methods, SiMAIM outperformed the compared methods.

Different social media sites might generate different types of attributes according to the characteristics of the sites. SiMAIM can use various attributes, not limited to the ones used in this study. In addition, different algorithms have advantages over different performance metrics. SiMAIm can also integrate other algorithms. The flexibility of SiMAIM also facilitates its portability to different social media sites.

The posts and replies on forum-oriented social media sites nowadays may contain many images and photographs, which may contain important clues for identifying sockpuppets and puppetmasters. Some accounts even put all text contents in the images to evade text-based examination, as only images are present in the posts or replies. To deal with images and photographs is future work. In addition, SiMAIM identifies puppetmasters by detecting the maximal cliques. Two sockpuppets may not have similar topic distributions, similar text contents, and social interactions. The identification of a puppetmaster with such sockpuppets can be achieved by using sub-group detection algorithms such as community detection [40, 41]. We will also focus on this issue in the future. Finally, considering that malicious behaviors are events, finding event witnesses [42] on social media sites may help identify and process sockpuppets and puppetmasters. The integration of witness discovery into SiMAIM has a potential benefit.

Acknowledgements This research was partly supported by the Ministry of Science and Technology under Grant No. MOST 108-2410-H-259-051.

Author contribution Both authors contributed to the study's conception and design. Both authors performed material preparation, data collection, and analysis. Y-HL wrote the manuscript. Y-HL received BS, MS, and PhD degrees in Information Management from National Taiwan University, Taiwan, in 2000, 2003, and 2009, respectively. In August 2009, he joined the Department of Information Management at National Dong Hwa University and is now an associate professor. His papers have appeared in Data and Knowledge Engineering, Journal of Systems and Software, Pattern Recognition, Intelligent Data Analysis, Applied Intelligence, NTU Management Review, and Computer Vision and Image Understanding. His research interests include big data analysis, machine learning, and knowledge management.

Funding This research was partly supported by the Ministry of Science and Technology under Grant No. MOST 108–2410-H-259–051. The authors have no relevant financial or non-financial interests to disclose.

Data availability Raw data analyzed in this study are available on the Mobile01 social media site (https://www.mobile01.com/topiclist.php?f=587).

Declarations

Conflict of interest The authors have no competing interests to declare relevant to this article's content.

References

1. Garima Kakkar (2023) What are the different types of social media in 2023? Digital Vidya. https://www.digitalvidya.com/blog/types-of-social-media/. Accessed April 18 2023
2. Alharbi A, Hai D, Xun Y, Tari Z, Khalil I (2021) Social media identity deception detection: a survey. ACM Comput Surv 54(3):1–35. https://doi.org/10.1145/1122445.1122456
3. Paredes JN, Simari GI, Martinez MV, Falappa MA (2021) NetDER: an architecture for reasoning about malicious behavior. Inf Syst Front 23:185–201. https://doi.org/10.1007/s10796-020-10003-w
4. Cook DM, Waugh B, Abdipanah M, Hashemi O, Rahman SA (2014) Twitter deception and influence: issues of identity, slacktivism, and puppetry. J Inf Warf 13(1):58–71
5. Hilary Brueck (2020) 20 bogus claims about the coronavirus, including a fake marijuana cure, and false links between Bill Gates and 5G. INSIDER. https://www.businessinsider.com/wuhan-coronavirus-myths-red-meat-bleach-antibiotics-and-sesame-oil-2020-2. Accessed April 18, 2023
6. Pizzagate conspiracy theory (2016) https://en.wikipedia.org/wiki/Pizzagate_conspiracy_theory. Accessed April 18, 2023
7. Writergate "三星寫手門." (2013) https://zh.wikipedia.org/zh-hant/%E4%B8%89%E6%98%9F%E5%AF%AB%E6%89%8B%E9%96%80. Accessed April 18 2023
8. Xing L, Deng K, Wu H, Xie P, Zhao HV, Gao F (2019) A survey of across social networks user identification. IEEE Access 7:137472–137488
9. Chung CT, Lin CJ, Lin CH, Cheng PJ (2014) Person identification between different online social networks. IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 01
10. Tan S, Guan Z, Cai D, Qin X, Jiajun B, Chen C (2014) Mapping users across networks by manifold alignment on hypergraph. Proc AAAI Conf Artif Intell. https://doi.org/10.1609/aaai.v28i1.8720
11. Vosecky J, Hong D, Shen VY (2009) User identification across multiple social networks. International Conference on Networked Digital Technologies. https://doi.org/10.1109/NDT.2009.5272173
12. Zhou X, Liang X, Zhang H, Ma Y (2016) Cross-platform identification of anonymous identical users in multiple social media networks. IEEE Trans Knowl Data Eng 28(2):411–424
13. Qu Y, Xing L, Ma H, Wu H, Zhang K, Deng K (2022) Exploiting user friendship networks for user identification across social networks. Symmetry 14:110. https://doi.org/10.3390/sym14010110
14. Zhang W, Shu K, Liu H, Wang Y (2019) Graph neural networks for user identity linkage.
15. Zhou F, Wen Z, Zhong T, Trajcevski G, Xu X, Liu L (2020) Unsupervised user identity linkage via graph neural networks. IEEE Glob Commun Conf. https://doi.org/10.1109/GLOBECOM42002.2020.9322311
16. Zafarani R, Liu H (2013) Connecting users across social media sites: a behavioral-modeling approach. ACM SIGKDD Int Conf Knowl Discov Data Min. https://doi.org/10.1145/2487575.2487648
17. Motoyama M, Varghese G (2009) I seek you: searching and matching individuals in social networks. Int Workshop Web Inform Data Manag. https://doi.org/10.1145/1651587.1651604
18. Zheng R, Li J, Chen H, Huang Z (2006) A framework for authorship identification of online messages: Writing-style features and classification techniques. J Am So Inf Sci Technol 57(3):378–393
19. Hao T, Zhou J, Cheng Y, Huang L, Wu H (2022) A unified framework for user identification across online and offline data. IEEE Trans Knowl Data Eng 34(4):1562–1575. https://doi.org/10.1109/TKDE.2020.3000287
20. Jain P, Kumaraguru P, Joshi A (2013) @ i seek'fb. me': Identifying users across multiple online social networks. International Conference on World Wide Web
21. Jain P, Kumaraguru P (2012) Finding Nemo: searching and resolving identities of users across online social networks. https://arxiv.org/abs/1212.6147
22. Zhang Y, Wang L, Li X, Xiao C (2016) Social identity link across incomplete social information sources using anchor link expansion. Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp 395–408

23. Nie Y, Jia Y, Li S, Zhu X, Li A, Zhou BJN (2016) Identifying users across social networks based on dynamic core interests. Neurocomputing 210:107–115
24. Goga O, Lei H, Parthasarathi SHK, Friedland G, Sommer R, Teixeira R (2013) Exploiting innocuous activity for correlating users across sites. International Conference on World Wide Web
25. Xing L, Deng K, Wu H, Xie P, Gao F (2019) Behavioral habits-based user identification across social networks. Symmetry 11:1134. https://doi.org/10.3390/sym11091134
26. Zhong Z, Cao Y, Yong Cao M, Guo GM, Nie Z, Nie Z (2018) CoLink: an unsupervised framework for user identity linkage. Proc AAAI Conf Artif Intell. https://doi.org/10.1609/aaai.v32i1.12014
27. Zhou W, Wang J, Lin J, Li J, Han J, Songlin H (2019) A time-series sockpuppet detection method for dynamic social relationships. In: Li G, Yang J, Gama J, Natwichai J, Tong Y (eds) Database Systems for Advanced Applications: 24th International Conference, DASFAA 2019, Chiang Mai, Thailand, April 22–25, 2019, Proceedings, Part I. Springer International Publishing, Cham, pp 36–51. https://doi.org/10.1007/978-3-030-18576-3_3
28. Liu D, Wu Q, Han W, Zhou B (2016) Sockpuppet gang detection on social media sites. Front Comput Sci 10:124–135. https://doi.org/10.1007/s11704-015-4287-7
29. Li J, Zhou W, Han J, Songlin H (2019) Sockpuppet detection in social network via propagation tree. In: Rodrigues JMF, Cardoso PJS, Monteiro J, Lam R, Krzhizhanovskaya VV, Lees MH, Dongarra JJ, Sloot PMA (eds) Computational Science–ICCS 2019: 19th International Conference, Faro, Portugal, June 12–14, 2019, Proceedings, Part V. Springer International Publishing, Cham, pp 507–513. https://doi.org/10.1007/978-3-030-22750-0_44
30. Li J, Yuan C, Zhou W, Wang J, Songlin H (2020) Who are controlled by the same user? Multiple identities deception detection via social interaction activity (Student Abstract). Proc AAAI Conf Artif Intell 34(10):13853–13854. https://doi.org/10.1609/aaai.v34i10.7199
31. Maity SK, Chakraborty A, Goyal P, Mukherjee A (2017) Detection of sockpuppets in social media. ACM Conf Comput Support Coop Work Soc Comput. https://doi.org/10.1145/3022198.3026360
32. Almishari M, Tsudik G (2012) Exploring linkability of user reviews. In: Foresti S, Yung M, Martinelli F (eds) Computer security – ESORICS 2012: 17th European Symposium on Research in Computer Security, Pisa, Italy, September 10-12, 2012. Proceedings. Springer, Berlin, Heidelberg, pp 307–324. https://doi.org/10.1007/978-3-642-33167-1_18
33. Yamak Z, Saunier J, Vercouter L (2018) SocksCatch: Automatic detection and grouping of sockpuppets in social media. Knowledge-Based Syst 149:124–142
34. Tsikerdekis M, Zeadally S (2014) Multiple account identity deception detection in social media using nonverbal behavior. IEEE Trans Inf Forensic Secur 9(8):1311–1321
35. Solorio T, Hasan R, Mizan M (2013) A case study of sockpuppet detection inWikipedia. Workshop on Language in Social Media:59–68
36. Hang Y, Feng H, Liu L, Li Z, Li X, Lin Z (2021) Sockpuppet detection in social network based on adaptive multi-source features. In: Chang V, Ramachandran M, Muñoz VM (eds) Modern Industrial IoT, big data and supply chain: proceedings of the IIoTBDSC 2020. Springer Singapore, Singapore, pp 187–194. https://doi.org/10.1007/978-981-33-6141-6_19
37. Salton G, McGill MJ (1986) Introduction to modern information retrieval. McGraw-Hill College
38. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3:993–1022
39. Bomze IM, Budinich M, Pardalos PM, Pelillo M (1999) The maximum clique problem. In: Du DZ, Pardalos PM (eds) Handbook of combinatorial optimization. Springer, Boston, MA, pp 1–74
40. Baltsou G, Vakali TK, A, (2022) Local community detection with hints. Appl Intell 52:9599–9620. https://doi.org/10.1007/s10489-021-02946-7
41. Guo K, Wang Q, Lin J, Wu L, Guo W, Chao KM (2022) Network representation learning based on community-aware and adaptive random walk for overlapping community detection. Appl Intell 52:9919–9937. https://doi.org/10.1007/s10489-021-02999-8
42. Liu W, Cao J, Yang Y, Liu B, Gao Q (2022) Category-universal witness discovery with attention mechanism in social network. Inf Process Manag 59(4):102947. https://doi.org/10.1016/j.ipm.2022.102947

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.