CSCD: A Cyber Security Community Detection Scheme on Online Social Networks

Yutong Zeng[1], Honghao Yu[1], Tiejun Wu[2], Yong Chen[1], Xing Lan[2], and Cheng Huang[1(✉)]

[1] School of Cyber Science and Engineering, Sichuan University, Chengdu, China
opcodesec@gmail.com
[2] NSFOCUS Technologies Group Co., Ltd., Beijing, China

Abstract. Online social networks (OSNs) are playing a crucial role in daily life, cyber security guys such as hackers, cyber criminals, and researchers also like to communication and publish opinions. Their discussions and relations can provide unprecedented opportunities for researcher to develop better insights about those accounts' activities in communities, which could be helpful for different purposes like cyber threat intelligent hunting and attack attribution. In this paper, we propose a scheme for cyber security community detection named CSCD on OSNs. We present a social relevance analysis method by building an ego network from one seed account. Through multidimensional analysis, features organized into four categories are taken into consideration and a recognition model is used to detect security-related accounts. Then we construct the social network, consisting of detected accounts, and propound a pruning strategy to remove weak relationships between accounts on the basis of edge features. An unsupervised overlapping community detection model is applied to unearthing potential communities. To evaluate our proposed scheme, we utilize Twitter as the platform to construct datasets. The recognition model achieves an accuracy up to 95.1%, and the community detection model obtains the best performance comparing to other former algorithms.

Keywords: online social network · community detection · cyber security · social network analysis

1 Introduction

Online social networks, where users can share information on different topics, follow any other people unidirectionally and keep track of the hottest trends, have exploded in popularity over the past few years. At the same time, there are lots of security researchers and hackers active on OSNs. OSNs have been utilized as a new platform to conduct malicious behaviors, including spreading malware

S. Goel et al. (Eds.): ICDF2C 2022, LNICST 508, pp. 135–150, 2023.
https://doi.org/10.1007/978-3-031-36574-4_8

[1], phishing [6], spamming [20,22], social engineering attack [7], and vulnerability disclosure [2,9]. As a consequence, OSNs have been preferred subjects to many security-related studies among researchers. To extract valuable information from OSNs, monitoring security-related accounts may provide a good solution. Comparing to relying on security-related keywords to identify vulnerabilities in posted content, keeping tabs on accounts such as security experts can indicate new security threats claimed to be unknown before. Therefore, how to detect such accounts automatically in a large scale has been a research hotspot.

In terms of Twitter, there have been a few recently published studies on detecting security-related accounts such as spammers, hackers or security researchers. Many methods have been proposed to combat the increasing number of these malicious accounts [20,22] in order to curb the risks brought by them. Meanwhile, some cyber security professionals, who are active users on OSNs, are spreading hacker tutorials, exchanging knowledge, providing their own insights on security incidents and etc. Monitoring these accounts has proven to be a good source of information for many purposes.

However, most current researches pay attention to detecting security-related accounts on OSNs individually rather than discovering security groups by a community-based approach, or analyzing properties within related communities. We lack basic insights into the characteristics of cyber security ecosystem on OSNs.

In this study, we propose CSCD, a scheme for cyber security community detection, and conduct an empirical study of discovered community. In summary, our contributions are as follows:

- We design a recognition model to differentiate security-related accounts from unrelated ones through extracting features from accounts' profile, behavior, time sequence of posts and content, reaching an accuracy rate of 95.1%.
- We propose a pruning strategy to construct the social network using edge features including the relevance of interaction, and the similarity of content, friends and followers, with the aim of removing the weak relationship between accounts.
- We put forward a three-staged security-related community detection scheme on OSNs, which can identify overlapping cyber security communities automatically starting from one seed account. We also present the application, evaluation of CSCD, and a case study of predicted community on Twitter.

The rest of the paper is organized as follows. Section 2 provides related work. Research goal is presented in Sect. 3. Section 4 details the methodology and different stages in CSCD. Subsequently, our results are described in Sect. 5. We conclude this paper and provide an outlook for future work in Sect. 6.

2 Related Work

2.1 Security-Related Account Recognition on OSNs

With the rapid growth of OSNs, there has been a tremendous rise in the number of malicious activities, which has been an urgent task to develop an effi-

cient detection system that can identify security-related accounts, such as spammers, compromised accounts, hackers and etc. Ellaky et al. [8] divided malicious accounts into four categories: Cloned account, Sock Puppets account [21], Sybil account [19] and Bot account [20]. However, most studies focus on the detection of malicious account. There has been few researches on the automated detection of hackers, security researchers or cyber criminals, who can provide security insights and do not engage in malicious activities directly. Aslan et al. [3] proposed the first fully automated classifier for such accounts. Though their best-performing model reached an accuracy rate of 97%, the dataset in their study was relatively small, which only contains 424 accounts. Mahaini et al. [12] presented a more mature detection framework with a systematic dataset construction method and a richer set of features to group security accounts into different categories on the basis of the study by Aslan et al. The detection of such accounts is also the focus of our study.

2.2 Security-Related Community Detection on OSNs

Community detection is a growing field of interest in the area of social network applications. Yang et al. [16] proposed the most common definition of community which is a subset of vertices that are densely connected between them and sparsely with the other nodes of the network. Chakrabo et al. [5] underlined that community detection is approached through two phases: unveiling the community structure from the network and evaluating the goodness of the result through several measure metrics. Community detection algorithms have been used in the discovery of security-related sub-communities. Lingam et al. [11] created a weighted signed Twitter network graph focused on behavioral similarity and confidence values as weighted edges. They proposed two algorithms, where the former recognized social botnet communities with malicious behavioral similarities, while the latter reconstructed and detected social botnet communities more accurately in presence of many forms of malicious behaviors. Results of their experiments demonstrated the validity of their algorithms than previous work.

In summary, most previous studies have focused on the detection of malicious accounts. There are relatively few studies on the detection of security-related communities on social networks. For community detection in this area, most studies only considered topological information without nodes' content or simply adopted cluster algorithms according to the similarity of nodes without the relationship on social networks.

3 Research Goal

We decide to conduct an experiment about how to detect cyber security community and provide an empirical analysis on how cyber security accounts are assembled and active on OSNs, considering the lack of past studies investigating connections in cyber security community and activities of inner accounts. To specify our work, we set some separate RQs for our study:

- **RQ1**: Can we propose a scheme which helps us detect cyber security community efficiently?
- **RQ2**: Can we study the social structure of these inner accounts on the basis of the data on OSNs?
- **RQ3**: What topics can we extract from the communication of these account on OSNs?
- **RQ4**: Can we identify key accounts in each sub-community?

For **RQ1**, which is also the basis of our work, we explain how we construct CSCD and how we get such accounts belonging to the specific field in the next section. To study **RQ2**, we use unsupervised overlapping method to identify communities established on the constructed network of collected accounts. Then we figure out social structures of these sub-communities. With the purpose of solving **RQ3**, natural language processing (NLP) is applied to analyzing behavior of accounts in sub-communities. For **RQ4**, we use PageRank algorithm in order to indicate key accounts in communities.

4 Methodology

4.1 Overview of Proposed Scheme

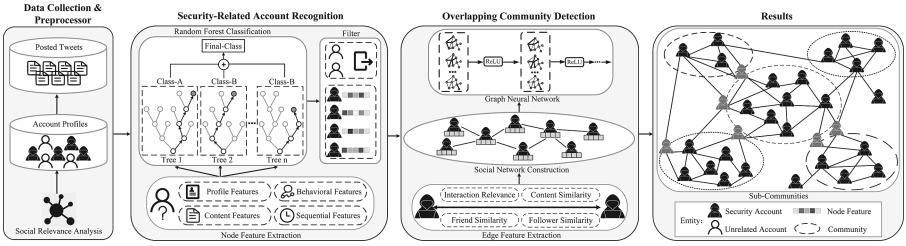

Fig. 1. The framework of proposed scheme named CSCD

Figure 1 presents the methodology adopted to achieve cyber security community detection. In this section, detailed description for each component of CSCD will be provided in following sections. A seed account is required to conduct our data collection at the beginning of CSCD. After selecting some accounts which are influential on security-related fields, we can extend other accounts by social relevance analysis. This collection method will bring in a number of accounts as well as relationships between accounts, especially after iteration. It is obvious that security accounts will also interact with unrelated account such as their family members, friends or celebrities. To remove such unrelated accounts from the dataset collected in previous step, we extract features in four categories to distinguish security accounts from unrelated ones. In addition, edge features in four categories are taken into consideration during the course of social network

construction. To solve the problem that community detection needs to deal with rich non-Euclidean graph data, we adopt GNN since the rapid development of GNN in graph mining technology.

4.2 Data Collection and Pre-processing

Data Collection. The first step of our approach is data collection. There are many varied accounts active on online social networks while security-related accounts only account for a small proportion, so we need to design a method to make sure we can collect security-related accounts as efficiently as possible. Twitter is selected as the platform to illustrate our scheme design. By utilizing the social relevance and homogeneity between accounts, we can find a great number of similar accounts. Therefore, we manually select some accounts which are influential and active in the field on Twitter as seed accounts based on below searching rules:

- Searching for accounts with a number of followers, which also engaged in hot issues and events related to cyber security.
- Searching for accounts belonging to renowned security companies or agencies.
- Searching for security topics and collecting accounts which engaged in these discussions and posted amount of content with good quality.

After selecting seed accounts manually, the crawler will collect their profile data and tweets data. Profile data, including an account's friends, followers, created time and description, is the basic information. Tweets data, consisting of tweet text and an account's interactive records, can reveal topics the user interested in and accounts it interacted with. Then we extract interactive accounts through records such as mention, favorite, reply, quote and retweet in tweets data. These interactive accounts, together with friends and followers of the seed account, form the ego network of the seed after deduplication. Also, for each extended account, our crawler will collect its profile data and tweets.

Pre-processing. We apply pre-processing techniques for each tweet before tweets are available to our model. We translate all tweet texts into English to facilitate the subsequent study. There is a problem that the content of retweets crawled through Twitter API is incomplete. Therefore, we need to restore the content of retweets. Links, usernames, emojis and punctuations in tweets are removed by regular expressions. After that, tweets are tokenized and lowercased, with lemmatization applied to each word in order to represent the inflected forms of a word as a single word. In the end, stopwords are deleted.

4.3 Security-Related Account Recognition

In this section, we propose a node feature extraction strategy from four categories. Since the aim of the proposed framework is to improve the security community detection performance, node features are available to the recognition model to narrow the collected Twitter data.

Table 1. Node features in four categories

Feature	#No	Description	Feature	#No	Description
Profile Features			**Content Features**		
Screen Name	#01	The number of alphabetic characters in screen name	Readability & Difficulty	#14	Lexical diversity
	#02	The number of numeric characters in screen name		#15	Flesch-Kincaid score
	#03	The number of capitalization in screen name		#16	SMOG index
Social Information	#04	The number of friends	Keywords Score	#17	Prototypical words
	#05	The number of followers		#18	Weirdness score
	#06	The ratio of friends and followers		#19	TF-IDF
Account Settings	#07	The presence of location	**Behavioral Features**		
	#08	The presence of URL	Tweeting Habits	#20	The number of posted tweets
Sequential Features				#21	The average number of hashtags
Time Sequence	#09	The average interval between tweets		#22	The average number of URLs
	#10	The standard deviation between tweets intervals		#23	The average number of mentions
	#11	The fraction of tweets posted in recent week		#24	The average number of replies
	#12	The minimum interval between tweets	Source Diversity	#25	The distinct sources of tweets
	#13	The maximum interval between tweets			

Node Feature Extraction. Referring to the previous study in security-related accounts detection [3], which has proven to be effective, 25 features organized into four categories are summarized in Table 1.

Profile Features. Profile features are extracted from accounts' profile provided by Twitter, including ID, username, created time, description, followers, friends, statuses count, location and website URL. These profiles are set up by their owners to leave an impression on others, so we extract features to encode the information conveyed by users.

Behavioral Features. After posting a tweet, the account's timeline can display the source of the tweet and whether the tweet is retweeted from another account. At the same time, user can interact with others on posted tweets by commenting, mentioning, retweeting, replying, quoting and favoriting. Therefore, there are six different behavioral features extracted in this study, where the first five features present the user's tweeting habits while the last one reflects the user's equipment using habits. We use Margalef's index to calculate the source diversity of tweets, which is expressed by the following formula:

$$\gamma_{SD} = \frac{m_S - 1}{\ln K} \tag{1}$$

where K is the statuses count and m_s is the number of distinct sources in tweets.

Content Features. Six metrics of features including lexical diversity, Flesch-Kincaid grade level score, SMOG (Simple Measure of Gobbledygook) index, and probability of some keywords extracted through three different techniques (prototypical words [15], weirdness score [10] and TF-IDF), are based on the content of tweets. The lexical diversity is a measure of how many different words appear in a document. The Flesch-Kincaid grade level and SMOG index, which measure the text difficulty through sentence and word length, are popular readability formulas. Keywords from above three keywords extraction techniques is helpful to

identify security-related accounts from normal ones. An account u is assigned a score for each keyword kw in the collection which is computed as follows:

$$kw\_score(kw, u) = \frac{|kw|}{\sum_{w \in W_u} |w|} \tag{2}$$

where $|kw|$ is the number of times that the keyword kw is issued by account u, and W_u is the set of all words issued by u.

Sequential Features. Sequential features are extracted from time sequence of accounts' posted tweets. These features depend on the interval of accounts' tweet time (in seconds) and the fraction of tweets belonging to the recent week (out of all tweets), presenting how active accounts are on Twitter.

Account Recognition Model. We pick the Random Forest as our recognition model, which has received increasing attention due to the outstanding classification results in many fields. The Random Forest yields reliable classification results using predictions derived from an ensemble of decision trees [4].

4.4 Overlapping Community Detection

Social Network Construction. Most of the existing literature on OSNs conduct community detection on graphs established on following relationships, but following relationships may do not accurately reflect associations between accounts. We propose a construction method to remove weak relationships existing between accounts from multiple dimensions. We consider social relationships and model OSN as an undirected graph $G = (V, E)$, where each node in V corresponds to an account in the network, and each edge in E corresponds to a bilateral undirected social relationship based on following and friendship. Then we introduce edge features to provide a more detailed description of the edge relationships from four categories.

Interaction Relevance. On OSNs, accounts can interact with each other by various methods as we mention above. These behaviors can reflect the strength of association between accounts, which can be reckoned as the strength of the edge. We adopt the number of interaction behaviors between accounts as the interaction feature IR.

Content Similarity. We extract the feature considering the phenomenon of homogeneity, which refers to the tendency of interconnected accounts on OSNs to be similar. Therefore, content similarity feature based on accounts' tweets are captured by means of NLP. We choose some feature words to present the content of an account using the mutual information(MI). The mutual information of two words is calculated as follows:

$$MI(w_i, w_j) = \frac{f(w_i, w_j)}{f(w_i) + f(w_j) - f(w_i, w_j)} \tag{3}$$

where $f(w_i, w_j)$ is the simultaneous occurrence frequency of w_i and w_j in one tweet. $f(w_i)$ is the frequency of w_i appearing in tweets. Each word gets a score

ws which is calculated by TF-IDF. An account's content can be presented as a set of words: $WS = \{w_1 : ws_1, w_2 : ws_2, ..., w_n : ws_n\}$. The content similarity between two accounts is calculated by cosine similarity algorithm, expressed as follows:

$$CS(v_i, v_j) = \frac{WS_{v_i} \cdot WS_{v_j}}{\|WS_{v_i}\| \cdot \|WS_{v_j}\|} \tag{4}$$

Friend and Follower Similarity. Twitter allows bidirectional social relationships between accounts through following, which forms a network around the account. By studying the similarity of friends and followers between two accounts, we are able to understand the overlap between accounts in terms of social structure. The follower and friends similarity are computed as follows:

$$Fo(r)Sim(v_i, v_j) = \frac{\left|Fo(r)List_{v_i} \cap Fo(r)List_{v_j}\right|}{\left|Fo(r)List_{v_i} \cup Fo(r)List_{v_j}\right|} \tag{5}$$

We construct a relationship between accounts on the basis of these edge features extracted from above four dimensions. We remove edges that do not satisfy one of the following conditions: (1)*whether there is interaction between accounts, i.e., IR > 0;* (2)*whether the content between the accounts is similar and overlap is existing in friends or followers to some extent, i.e., CS > 0 and (FoSim > 0 or FrSim > 0).* Under above conditions, edges which proved the existing of relationships between accounts are retained, and by contrast edges maintaining weak relationships are removed. These limitations guarantee the strong connection between accounts.

Overlapping Community Detection Model. It is intuitive to be aware that each account can belong to multiple communities, whose number is potentially unlimited. However, it is of great difficulty to classify an account into a literal community and confirm the number of communities in the network explicitly because of the lack of ground truth. Confronted with above problems, we divide the whole network into different communities depending on combining connections and similarities between accounts. Referring to the approach proposed by Shchur et al. [17] to detect overlapping communities, we fuse the power of GNN with the Bernoulli-Posson probabilistic model. The structure of our model is illustrated below.

Given the undirected Graph G and the matrix of node features X, A is the adjacency matrix and $F \in \mathbb{R}_{\geq 0}^{N \times C}$ is the affiliation matrix, where C is the number of communities and N is the number of nodes. For our model architecture, the same as mentioned in [17], deeper models didn't lead to better results, so we just adopt a 2-layer Graph Convolutional Network(GCN) to receive parameters:

$$F := GCN_\theta(A, X) = ReLU(\widehat{A}ReLU(\widehat{A}XW)W) \tag{6}$$

where $\widehat{A} = \widetilde{D}^{-1/2}\widetilde{A}\widetilde{D}^{-1/2}$ is the normalized adjacency matrix, $\widetilde{A} = A + I_N$ is the adjacency matrix with self loops, and $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$ is the diagonal degree matrix of $\widetilde{A}$.

The negative log-likelihood of the Bernoulli-Poisson is optimized in the model in order to avoid the problem that the real-world sparse graph makes a large contribution to the loss. The balanced model is expressed as follows:

$$\mathcal{L}(F) = -\mathbb{E}_{(u,v)\sim P_E}\left[\log(1 - exp(-F_u {F_v}^{\mathrm{T}}))\right] + \mathbb{E}_{(u,v)\sim P_N}\left[-F_u {F_v}^{\mathrm{T}}\right] \quad (7)$$

where P_E and P_N denote uniform distributions over edges and non-edge, respectively.

To get the optimal affiliation matrix, the neural network parameter $\theta^\star$ is used to minimize the balanced negative log-likelihood:

$$\theta^\star = \underset{\theta}{argmin}\, \mathcal{L}(GNN_\theta(A, X)) \quad (8)$$

We can get the result of predicted communities through the obtained community affiliation matrix F.

5 Evaluation and Case Study

5.1 Experiment Design

We designed several experiments and conducted a case study to validate our methodology and answer RQs set before on Twitter. We began with a group of seed accounts to assess our account extension strategy. To evaluate features we extracted above, we compared the results over different baseline models. The community detection model was compared with other overlapping community detection algorithms over ground-truth datasets. The intuition that combining the graph with nodes' content leads to better results was verified by measuring the goodness of metrics. Besides, the result of community detection was shown by the means of visualized adjacency matrix to prove the validity of the overlapping community detection. Above processes responded to **RQ1** and **RQ2**. With the purpose of examining that CSCD truly detected security-related communities and solved **RQ3** and **RQ4**, an empirical study was provided to take deep insights on a sample community.

In order to evaluate the performance of the unsupervised community detection algorithm, we adopted the following metrics to measure the results of our experiments: Normalized Mutual Information, Modularity, Conductance, Density, and Clustering Coefficiency. The first metric NMI [14] was used for evaluating model performance on ground-truth datasets, while the other four were applied to testify our intuition on Twitter dataset. Except for the Modularity, we calculated other three metrics for each predicted community and averaged the value.

We used a Modularity calculation method for overlapping community proposed by Shen et al. [18]. Conductance measures the fraction of total edge volume that points outside the community, expressed by the following formula:

$$Conductance = \frac{\sum_{u\in C, v\notin C} A_{uv}}{\sum_{u\in C, v\in C, v\neq u} A_{uv} + \sum_{u\in C, v\notin C} A_{uv}} \quad (9)$$

Density is built on the intuition that good communities are well connected, which is calculated as follows:

$$Density(C) = \frac{2|E|}{|V|(|V|-1)} \tag{10}$$

Clustering coefficient, which is based on the premise that communities are manifestations of locally inhomogeneous distributions of edges for the reason that nodes with common neighbors are more likely to be connected with each other, defined by the following formula:

$$ClustCoef(C) = \frac{1}{|V|} \sum_{i,k_i>1} \frac{2t}{k_i(k_i-1)} \tag{11}$$

where t_i is the difference between the number of edge connections in the egocentric network to which the node belongs and the degree of the node.

5.2 Dataset

We firstly selected 10 seed accounts manually, which belong to hackers or security researchers, as the initial crawl task. All crawling tasks are iterated only once, i.e., only the extended accounts of the seed are crawled. This eventually resulted in a collection of 30,469 Twitter accounts, where accounts extended from each seed ranged from 482 to 5439. There were 10,467,239 tweets obtained from collected accounts under the limitation that only 600 latest tweets were crawled.

After collecting accounts data, we needed to create a labeled dataset as a ground truth dataset in order to train the recognition model. We selected randomly 5,138 active accounts, and all the selected accounts were manually labeled by cyber security experts. After inspecting the description and posted tweets of each account, we obtained 4,058 security-related accounts and 1,080 unrelated ones. To validate of our community detection model, we chose Facebook datasets with ground truth provided by Mcauley et al. [13].

5.3 Experiment and Results

Account Recognition. We extracted 574 node features where 552 features were words generated from top 200 words in three keywords techniques after deduplication. Supervised learning was used in order to detect the security-related accounts. Table 2 compares the measure metrics over different baseline machine learning models. We finally chose the Random Forest as a part of CSCD because of its performance, obtaining an accuracy rate of 95.1%. Comparing to previous work [12], though we had a better performance in our experiments, the results we obtained were based on different datasets.

Table 2. Account recognition performance comparison over different models

	ACC	FPR	Precision	Recall	F-measure	MCC	AUC
SVM	0.491	0.062	0.938	0.099	0.178	0.190	0.545
LogisticRegression	0.622	0.013	0.987	0.331	0.495	0.413	0.663
DecisionTree	0.802	0.028	0.971	0.666	0.791	0.654	0.821
GradientBoosting	0.928	0.011	0.989	0.880	0.932	0.862	0.934
RandomForest	0.951	0.002	0.997	0.915	0.955	0.906	0.956

Community Detection. We used a dataset with 5,439 accounts extended from one seed account and four ground-truth datasets in our experiment. After detecting accounts in the extended dataset through our trained security-related recognition model, we retained 3,193 accounts and removed other unrelated accounts. We firstly constructed a graph with 65,062 edges. After removing some weak association, we retained 64,970 edges in our constructed social network.

Table 3. Community detection performance comparison over different methods

	Nodes	Edges	SNMF(%)	NNSED(%)	DANMF(%)	BigCLAM(%)	GNN(%)
Dataset I	66	2145	25.9	28.0	27.8	13.2	**42.2**
Dataset II	159	12561	19.3	33.4	41.1	32.0	**52.0**
Dataset III	227	25651	10.4	14.5	19.4	7.5	**35.8**
Dataset IV	755	284635	16.1	10.2	20.9	4.1	**43.8**

Table 4. Metrics comparisons of community detection over different inputs

	Modularity	AvgCond	AvgDens	AvgClustCoef
A	0.1942	0.3628	5.835×10^{-2}	1.634×10^{-3}
X	0.2166	0.2975	4.001×10^{-2}	6.901×10^{-4}
$A + X$	**0.2216**	**0.3855**	$\mathbf{6.077} \times 10^{-2}$	$\mathbf{1.827} \times 10^{-3}$

Table 3 showed how well different algorithms recover the ground-truth communities, and our model obtained the best performance competing against other methods. We also tried to use the adjacency matrix A, node features X and a combination of both A and X as input, respectively. After several attempts with different candidate values, the community number was set to seven. We measured the results under different inputs, shown in Table 4. We got a better result when we used $A + X$ as the input. Figure 2 shows the visualized adjacency matrix where the lines and columns are nodes of the whole graph after community division with the input of $A + X$. We excluded nodes that did not belong to any community in the figure. The cell at the intersection of the line

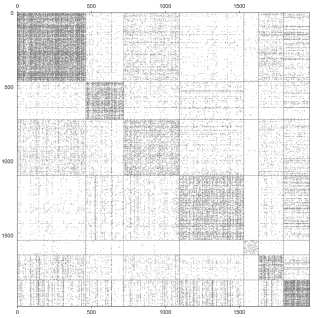

Fig. 2. Visualized adjacency matrix

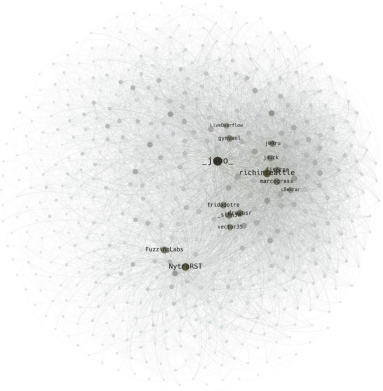

Fig. 3. The structure and key accounts of the sample community

and column is displayed using a contrasted color. Blocks on the diagonal where nodes are densely connected represent a community. We have detected cyber security communities and presented the social structure of these inner accounts through CSCD, which well answered the **RQ1** and **RQ2**.

5.4 Case Study of a Sample Sub-Community

In this subsection, we took a sample sub-community which we found above to make an empirical study on it in order to make our predicted communities more convincing. This sub-community contained 253 accounts.

Key Accounts. There is a group of active accounts which are pivotal in the community such as opinion leaders. Although they are part of the community, for the reason that these individuals have special skills, knowledge or other characteristics, they are able to have some influence on others. We utilized the PageRank algorithm to tap into key accounts in the selected sub-community. Figure 3 shows the identified key accounts and the topological structure of the community. After inspecting these accounts manually, most of these accounts were security researchers or hackers with high impact in this community.

Hashtag Analysis. Hashtag, which is a collection of characters preceded by the (#) symbol, intends to organize a discussion for a particular event. Users can join the discussion of a specific topic through hashtags which they are interested in and express their opinions on. We extracted the top trendy hashtags and excluded some meaningless hashtags from the content of the sub-community, shown dynamically in Fig. 4. We can know the property of the community and

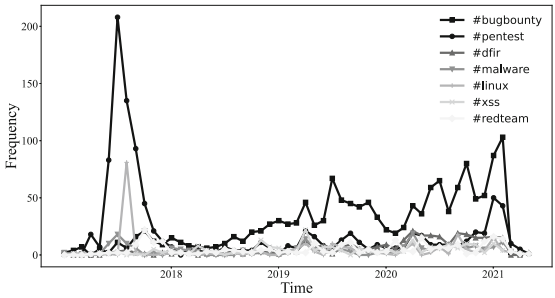

Fig. 4. Hashtags evolution over time

Table 5. Top topics words in extracted topics

Topic	Words
Daily Life	time support nice job awesome live love week fun happy read talk community wait
Vulnerability Mining	file xss cve txt bypass api target rce php injection ssrf ip bug bounty http
Pentest	windows security linux enroll team malware privilege tool penetration vulnerability
Vulnerability Mining	cyber fuzzing tickets team facebook labs free join session software read channel
Web	web google xss application exploit data dns bug post log4j bounty cve blog resources
Penetest	suite tool alert burp cve bug javascript bounty payload server src bypass poc

topics which they were interested in at different times through analyzing the evolution of hashtags.

Topic Analysis. In this part, we concentrated on analyzing topics that security-related communities usually talked about. We applied the LDA-based topic modeling algorithm to analyze the corpus of the whole sub-community. LDA regards a document as a bag of words and assume k topics spread across all m documents in a corpus. We used the timeline of an account as a document and need to adjust the k manually to get the best performance. After several attempts and optimizing by coherence, we set the k of our model to 6. Table 5 shows the extracted topics of the selected community.

We can acknowledge that this community consist of web security enthusiasts according to the above analysis. They focus on the vulnerability mining and penetration test. As we can see from Fig. 2, the community excavated from the network has been proved to be equipped with intensive connections inside. The

topic analysis goes a step further, showing a uniformity of community content. We can have a comprehensive understanding of other communities with the similar analysis method, which solved the **RQ3** and **RQ4**.

6 Conclusion and Future Work

6.1 Conclusion

In summary, we demonstrated the efficiency of our account extension method in the experiment, which extended a large scale dataset containing significant quantities of security-related accounts through social relevance analysis. We trained and compared the recognition model over the features extracted from four categories. Our results showed that the Random Forest is the machine learning model with the best performance, reaching an accuracy rate of 95.1%. Besides, the recognition model helped us narrow the scope of subsequent community detection. An edge construction strategy was propounded to remove weak edges by analyzing relationships between accounts through multiple dimensions. To solve existing problems that an account can belong to multiple communities, we adopted an unsupervised overlapping community detection model. We proposed CSCD combining the content with the graph structure. Additionally, an empirical study was provided on the ecosystem of the security-related community.

6.2 Future Work

We acknowledge that our collected data only accounts for a small proportion of the whole OSN, but it can be solved through multiple iterations. The influential or key accounts in communities may be a good entry of a new iteration. There are also some problem existing in our community detection model. The challenge of unknown number of communities hasn't been solved well. Many community detection algorithms still need to define the number of communities manually, which also exists in our study. The model doesn't consider the strength of edges between accounts so we didn't use a weighted network. CSCD can be applied to community detection on other specific groups or other online social network platforms as well. Besides, the content of detected communities can be a good data source for cyber threat intelligence.

Acknowledgment. This research is funded by the National Key Research and Development Program of China (No. 2021YFB3100500), CCF-NSFOCUS KunPeng Research Fund (No. 202105).

References

1. Ab Razak, M.F., Anuar, N.B., Salleh, R., Firdaus, A.: The rise of "malware": bibliometric analysis of malware study. J. Netw. Comput. Appl. **75**, 58–76 (2016)
2. Alves, F., Bettini, A., Ferreira, P.M., Bessani, A.: Processing tweets for cybersecurity threat awareness. Inf. Syst. **95**, 101586 (2021)

3. Aslan, Ç.B., Sağlam, R.B., Li, S.: Automatic detection of cyber security related accounts on online social networks: Twitter as an example. In: Proceedings of the 9th International Conference on Social Media and Society, pp. 236–240 (2018)
4. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
5. Chakraborty, T., Dalmia, A., Mukherjee, A., Ganguly, N.: Metrics for community analysis: a survey. ACM Comput. Surv. (CSUR) **50**(4), 1–37 (2017)
6. Djaballah, K.A., Boukhalfa, K., Ghalem, Z., Boukerma, O.: A new approach for the detection and analysis of phishing in social networks: the case of Twitter. In: 2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS), pp. 1–8. IEEE (2020)
7. Egele, M., Stringhini, G., Kruegel, C., Vigna, G.: Towards detecting compromised accounts on social networks. IEEE Trans. Dependable Secure Comput. **14**(4), 447–460 (2015)
8. Ellaky, Z., Benabbou, F., Ouahabi, S., Sael, N.: A survey of spam bots detection in online social networks. In: 2021 International Conference on Digital Age & Technological Advances for Sustainable Development (ICDATA), pp. 58–65. IEEE (2021)
9. Huang, S.Y., Ban, T.: Monitoring social media for vulnerability-threat prediction and topic analysis. In: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 1771–1776 (2020)
10. Lau, R.Y., Xia, Y., Ye, Y.: A probabilistic generative model for mining cybercriminal networks from online social media. IEEE Comput. Intell. Mag. **9**(1), 31–43 (2014)
11. Lingam, G., Rout, R.R., Somayajulu, D.V., Das, S.K.: Social botnet community detection: a novel approach based on behavioral similarity in Twitter network using deep learning. In: Proceedings of the 15th ACM Asia Conference on Computer and Communications Security, pp. 708–718 (2020)
12. Mahaini, M.I., Li, S.: Detecting cyber security related twitter accounts and different sub-groups: a multi-classifier approach. In: Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 599–606 (2021)
13. Mcauley, J., Leskovec, J.: Discovering social circles in ego networks. ACM Trans. Knowl. Discov. Data (TKDD) **8**(1), 1–28 (2014)
14. McDaid, A.F., Greene, D., Hurley, N.: Normalized mutual information to evaluate overlapping community finding algorithms. arXiv preprint arXiv:1110.2515 (2011)
15. Pennacchiotti, M., Popescu, A.M.: Democrats, republicans and starbucks afficionados: user classification in Twitter. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 430–438 (2011)
16. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. Proc. Natl. Acad. Sci. **101**(9), 2658–2663 (2004)
17. Shchur, O., Günnemann, S.: Overlapping community detection with graph neural networks. In: Deep Learning on Graphs Workshop, KDD (2019)
18. Shen, H., Cheng, X., Cai, K., Hu, M.B.: Detect overlapping and hierarchical community structure in networks. Physica A **388**(8), 1706–1712 (2009)
19. Wang, B., Jia, J., Zhang, L., Gong, N.Z.: Structure-based sybil detection in social networks via local rule-based propagation. IEEE Trans. Netw. Sci. Eng. **6**(3), 523–537 (2018)
20. Wu, Y., Lian, D., Xu, Y., Wu, L., Chen, E.: Graph convolutional networks with Markov random field reasoning for social spammer detection. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 1054–1061 (2020)

21. Yamak, Z., Saunier, J., Vercouter, L.: Sockscatch: automatic detection and grouping of sockpuppets in social media. Knowl.-Based Syst. **149**, 124–142 (2018)
22. Zhang, Y., Zhang, H., Yuan, X., Tzeng, N.F.: TweetScore: scoring tweets via social attribute relationships for twitter spammer detection. In: Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security, pp. 379–390 (2019)