Detecting Cyber Security Related Twitter Accounts and Different Sub-Groups: A Multi-Classifier Approach

Mohamad Imad Mahaini* and Shujun Li†
Institute of Cyber Security for Society (iCSS) & School of Computing, University of Kent, Canterbury, UK
Email: *mim@kent.ac.uk, †S.J.Li@kent.ac.uk

***Abstract*—Many cyber security experts, organizations, and cyber criminals are active users on online social networks (OSNs). Therefore, detecting cyber security related accounts on OSNs and monitoring their activities can be very useful for different purposes such as cyber threat intelligence, detecting and preventing cyber attacks and online harms on OSNs, and evaluating the effectiveness of cyber security awareness activities on OSNs. In this paper, we report our work on developing several machine learning based classifiers for detecting cyber security related accounts on Twitter, including a base-line classifier for detecting cyber security related accounts in general, and three sub-classifiers for detecting three subsets of cyber security related accounts (individuals, hackers, and academia). To train and test the classifiers, we followed a more systemic approach (based on a cyber security taxonomy, real-time sampling of tweets, and crowdsourcing) to construct a dataset of cyber security related accounts with multiple tags assigned to each account. For each classifier, we considered a richer set of features than those used in past studies. Among five machine learning models tested, the Random Forest model achieved the best performance: 93% for the base-line classifier, 88-91% for the three sub-classifiers. We also studied feature reduction of the base-line classifier and showed that using just six features we can already achieve the same performance.**

***Index Terms*—Cyber Security, Machine Learning, Classification, OSN, Online Social Network, Twitter, Crowdsourcing, Cyber Threat Intelligence, OSINT, Open Source Intelligence**

I. Introduction

Online social networks (OSNs) have become part of many people's everyday life. According to We Are Social Inc.'s Digital 2020 report (https://wearesocial.com/digital-2020/), over 3.8 billion users are now using OSNs, out of over 4.5 billion people who use the Internet. As Internet experts, both cyber security experts and cyber criminals are among active users of OSNs. Cyber security professionals use OSNs for different purposes such as knowledge exchange, cyber security awareness, and offering help to people and organizations on cyber security matters. On the other hand, cyber criminals often utilize OSNs to reach out to victims, boast about their past "achievements", and even talk about their future attacking plans. The activities of cyber security professionals and criminals have been found a good source of information for many purposes, such as cyber threat intelligence and understanding behaviors of cyber criminals and related groups [1–3]. Monitoring cyber security related accounts on OSNs requires automatic detection of such accounts.

There has been some recent research about the use of machine learning to detect whether a Twitter account is cyber security related or not [4] or to detect if an account belongs to a specific hacktivist group (e.g., Anonymous [2]). However, such work is still limited in their generalizability and validation of performance. In addition, there is a lack of more general-purpose sub-classifiers that can classify different sub-groups of cyber security related accounts, e.g., cyber security individuals (vs. groups and organizations), hackers in general (both people and groups), researchers and research organizations, etc. Such sub-classifiers will allow more fine-grained monitoring of the different sub-groups to support more targeted monitoring and behavioral analysis.

In this paper, we report our work that addresses a number of the issues about classifying cyber security related accounts on Twitter. Our work is based on a three-staged methodology: a more systematic data collection process, crowdsourcing-based labeling experiment, and development of machine learning based classifiers. Our main contributions are as follows:

- We followed a systematic (based on a cyber security taxonomy, more representative account sampling, and crowdsourcing) approach to construct a dataset of labeled Twitter accounts with multiple tags, not just binary labels like in past studies (e.g., [2, 4]).
- We identified a richer set of features for developing such classifiers than what was used in previous studies.
- We developed four classifiers, one for general accounts related to cyber security, and three others for three typical sub-groups: individuals, those related to hacking, and those belonging to academia. For all classifiers, the best machine learning model (Random Forest) achieved a good performance: 93% for the base-line classifier, 88-91% for the three sub-classifiers.
- Among all features we used, we identified a very small number (six) of highly significant ones, which when combined can support a lightweight Random Forest model with the same performance (93%) for the base-line classifier.

We released the anonymized feature sets along, with the source code of our classifiers on the paper's companion web page[1]. It deserves mentioning that our three-staged methodology is very general so can be used to develop benchmarking datasets and classifiers of other communities on OSNs.

The rest of the paper is organized as follows. We introduce related work in Section II. Then, we explain our methodology in Section III, and discuss the three different stages of our work, data collection, the crowdsourcing-based labeling experiment, and the machine learning based classifiers, in Sections IV–VI, respectively. The results of the machine learning based classifiers are discussed in detail in Section VII. The limitations of our work and planned future work are discussed in Section VIII. The last section concludes the paper.

II. Related Work

For social media analytics research, there is a general need to automatically detect a specific community or type of accounts on

[1]https://cyber.kent.ac.uk/research/cyber_Twitter_classifiers/

ASONAM '21, November 8–11, 2021, Virtual Event, Netherlands

ACM ISBN 978-1-4503-9128-3/21/11
https://doi.org/10.1145/3487351.3492716

a specific OSN platform. Classifying users on OSNs has been conducted through a variety of methods [5]. Some people worked on detecting the political orientation [6] or party affiliation [7] from the posts made by users on social media, while others researched detecting gender, age [8], personality [9], income [10], and many other attributes related to social media users.

There has been a range of related work on automatic classification of OSN accounts for cyber security purposes. For instance, since spammer accounts are used by cyber criminals and hackers to perform a wide range of attacks, many of the studies conducted in this field have been around spam/spammers detection [11, 12]. Similarly, due to the role of social bots and fake accounts in spreading mis- and dis-information on OSNs, detection of such accounts has become a hot topic in recent years [13, 14].

As mentioned in the Introduction, monitoring activities of cyber security related accounts on OSN can help us gather useful intelligence about cyber criminals and cyber security professionals. There has however been relatively little work on automatic detection of those accounts. In [15], Lee et al. developed a social media threat intelligence system called Sec-Buzzer, which includes a semi-automated component for adding new cyber security experts on OSNs by combining a number of mechanisms: mentions of active known accounts, a "topic relevance" score defined by the number of relevant cyber security topics, and manual confirmation by the Sec-Buzzer manager. The first fully automated classifier of this kind we are aware of is [4], in which Aslan et al. identified multiple candidate feature sets and tested several machine learning models to develop a classifier for detecting cyber security related accounts. Their best-performing model (Random Forest) achieved accuracy above %97 for different feature sets. The dataset they used is relatively small (424 accounts) and was constructed in an ad hoc manner (e.g., cyber security related accounts were selected from an ad hoc public list, and all account labels were assigned by a single cyber security expert).

Yet another interesting work is [2] where Jones et al. developed a classifier for detecting Twitter accounts affiliated with the well-known hacktivist group "Anonymous" in order to reconstruct a network of such accounts for studying their activities over time. Their classifier based on Random Forest achieved an F1-score of 94%. Their classifier relies on ad hoc features manually identified for Anonymous accounts and the dataset used was collected based on a small number of (five) seed accounts, so it cannot be directly generalized to other types of cyber security related OSN accounts.

The above discussions point to a clear research gap of machine learning based classifiers for detecting cyber security related accounts on OSNs, especially those for detecting different sub-communities such as individual experts, hacking community, and cyber security academia. In addition, there is a lack of public datasets for supporting the development and benchmarking of such classifiers. The datasets developed in [2, 4] are not systematic enough so cannot be easily generalized. In addition, these two datasets have not been made available to other researchers. We aim at filling those gaps, including a more systematically constructed dataset.

III. Methodology

Following similar work of other researchers [4], we designed a general methodology for detecting a specific type of accounts based on textual data, which can be easily applied to any OSN platform. The methodology consists of three main stages as illustrated in Figure 1. The first stage is about data collection, where raw (unlabeled) data about a reasonable number of accounts are collected from the target OSN platform. The second stage is about dataset construction, where a crowdsourcing-based approach is used to produce ground truth labels for the OSN accounts – selected in Stage 1 – by cyber security experts. The final stage is about building machine learning based classifiers based on the labeled dataset from Stage 2, following the general steps for developing supervised machine learning based classifiers: feature extraction, training and testing, performance evaluation, and finally feature importance analysis to reduce the dimensionality of the feature set used. After feature extraction, we also have two additional steps: creating sub-databases for multiple classifiers, and preparing different candidate feature sets to identify the best-performing feature set for each classifier. In this paper, we focus on Twitter as an example OSN platform because we observed a more active community of cyber security related accounts on this platform. The raw data collected in this case include account profile data and tweet timelines (both are user-generated textual data). The following three sections will give details of the three stages, respectively.

IV. Data Collection

A. Harvesting Tweets

Different methods can be used to collect data from Twitter. Some researchers used seed Twitter accounts identified manually (e.g., [2]), while others used public lists created by Twitter users related to the target area of interest (e.g., [4]). In [7], Pennacchiotti and Popescu constructed a dataset by using public Twitter directories (e.g., Twellow and WeFollow) to get data they needed (Twitter accounts that are well-labeled). The majority of past studies used the Twitter Search API to search for tweets and users using a list of relevant keywords (e.g., [10]). The last approach may lead to the inclusion of less representative samples, therefore reducing the overall generalizability of the results.

Considering the limitations of the Twitter Search API, we decided to use the Twitter Sampling API to collect a set of tweets that include cyber security related discussions, from which we can further identify cyber security related Twitter accounts. We created a data harvesting tool that connects to the sampling API endpoint and consumes tweets. According to Twitter (https://developer.twitter.com/en/products/tweets/sample), the sampling API returns a small but random percentage of all public Tweets. We collected 478 million tweets over 16 months starting from January 2019.

B. Sampling Cyber Security Related Tweets

The purpose of this step is to select a subset of tweets from the harvested data collection in the previous step. Those tweets should be cyber security related tweets and thus more likely to be tweeted – or retweeted – by cyber security experts. To achieve this, we need to filter the tweets using a list of cyber security related n-grams.

The required list was extracted from the general cyber security taxonomy reported in [16] (https://cyber.kent.ac.uk/research/cyber_taxonomy). The taxonomy contains more than 1,900 terms and their different wordings, e.g., "Cyber Security" / "Cybersecurity" and "0day" / "zero day". Thus, we compiled a list of 2,236 n-grams in total. Then, the whole list was used to search the tweets collection for its n-grams. As a result, we obtained search results statistics for each n-gram. Then, we manually reviewed each n-gram aided by the search statistics which enabled us to reduce the list of n-grams to just 795 by excluding n-grams that are duplicates, outlets, loosely cyber security related, implicit by other n-grams or have low search results. The final reduced list is quite important not just for this step, but also will be used later in the features extraction step, see Section VI-A.

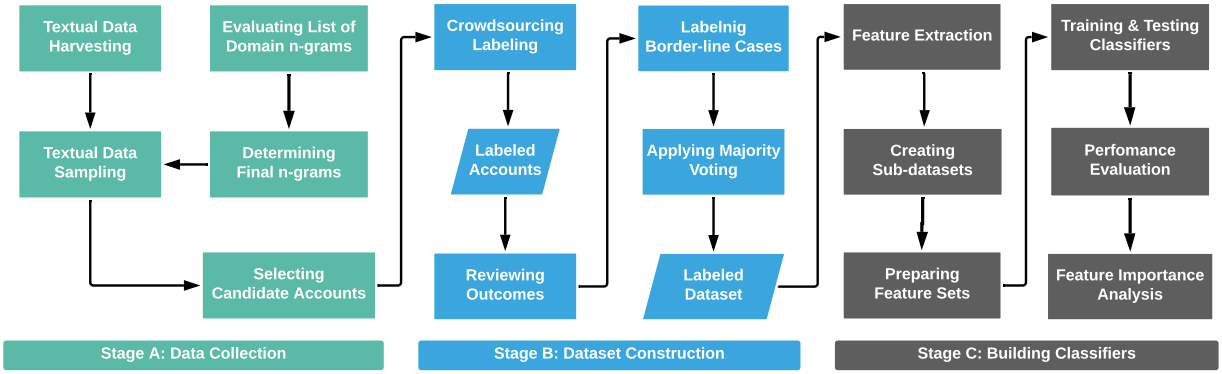


Fig. 1: The proposed methodology for developing classifiers for detecting cyber security related OSN accounts and different sub-groups

Using the final n-grams list, we filtered the 478 million tweets reducing the total to just 1.1 million tweets whose texts contain at least one cyber security n-gram. The returned tweets are 0.23% of the whole dataset. After that, we removed the retweets (53%) and kept only the original tweets (47%) as we were interested in the content that is produced by the Twitter accounts themselves. Thus, we ended up with 538,102 tweets (0.11% of the whole harvested collection).

C. Selecting Candidate Accounts

We wanted to create a good-sized labeled dataset as a ground truth dataset to develop our classifiers. This dataset would also allow us to gain more insights about what makes a Twitter account cyber security related (as perceived by our participants).

We identified the Twitter accounts that posted the "original" tweets from the Tweets Sampling step which resulted in 57,018 unique accounts. After removing the suspended and deleted accounts (9%), we ended up with 51,868 accounts (91%) from which to select a subset. Since the selected accounts would be manually labeled by cyber security experts and due to difficulties in the recruitment process, we set our target to obtain a labeled dataset with 1,300 to 1,400 accounts that were more likely to be labeled as cyber security related accounts in the labeling experiment. That means we needed to select 2.5% out of 51,868 Twitter accounts. To do this, we introduced two simple measures to each Twitter account, TiD (Number of **T**weets **i**n the **D**ata sample, i.e., the original tweets) and KiD (Number of **k**eywords **i**n the **D**escription field of the Twitter account). The keywords list that was used here is the same one we used previously in the Tweets Sampling, see Section IV-B.

Then, we set three criteria: A) Twitter accounts that had at least two tweets in the original tweets data collection (i.e., TiD $\geq$ 2), B) Twitter accounts that had at least one cyber security keyword (i.e., KiD $\geq$ 1) mentioned in their profile description. C) Twitter accounts that satisfied both A and B. To know which criterion to use, we created a group of 100 accounts satisfying the corresponding criterion. Each group was then manually labeled by checking whether each Twitter account was cyber security related or not. The results of the 3 groups were 44, 87, and 89 respectively. Thus, we used Criterion C and randomly selected 1,300 accounts to use in the labeling experiment.

V. Dataset Construction

For this stage of our methodology, we conducted a crowdsourcing-based labeling experiment. By recruiting a number of cyber security experts as human labelers, the experiment allowed us: 1) to construct a labeled dataset of cyber security related Twitter accounts along with other tags like Individual, Hacker, Academia, etc.; 2) to arrange a short questionnaire about how human labelers define "cyber security" and behaviors of cyber security related Twitter accounts. The questionnaire was used to gather potentially useful information for us to better define features used for detecting cyber security related Twitter accounts.

A. Labeling System Implementation

To make the labeling experience efficient and easy to accomplish by participants, we explored the existing tools for crowdsourcing, but unfortunately, none of them met our requirements. We wanted a way to dynamically assign and monitor tasks allocated to participants, considering several factors such as making the procedure robust in case a Twitter account allocated was deleted or suspended by Twitter before it was labeled. Also, we needed the labeling interface to be friendly with on-screen controls so that the participant could assign the required labels with the least time needed. Moreover, the labeling controls were customized as per our needs, making it difficult to use an existing crowdsourcing platform.

Thus, we implemented our own web-based labeling system. To make the system more user-friendly and productive, we co-designed it with some local colleagues who are cyber security experts and potential participants. Also, we had discussion groups with more than 20 local cyber security academics. Finally, to ensure the quality of the labeling results, we used a majority voting mechanism. Each labeled account was allocated to three different participants, so that for an account to be labeled properly (cyber security related or not), three votes were needed to allow a majority voting.

B. Participant Recruitment

As we selected 1,300 accounts to be labeled and since we would apply majority voting, we had a total of $1,300 \times 3 = 3,900$ labeling tasks. Thus, we needed to recruit around 100 participants. Considering the labeling task is quite time consuming and requires human participants to have good cyber security knowledge, we decided to introduce a £15 Amazon UK voucher as a compensation for each participant's contribution.

The participants should have experience in cyber security to be qualified for the experiment. The experience can be theoretical or practical, so we accepted cyber security students, researchers, educators, consultants, and other types of experts. Also, we sent emails about our experiment to several cyber security professional networks, e.g., SPRITE+ (https://spritehub.org), cyber security conferences' mailing lists like FOSAD (https://sites.google.com/uniurb.it/fosad), cyber security research groups and centers in the UK. In addition, we invited some cyber security professionals who worked in industry or

NGOs. We ran the experiment for around three months as recruitment took some time due to the COVID-19 pandemic. We managed to recruit 89 participants in the end. The demographics and other statistics about the participants are in Appendix B.

C. Labeled Dataset

We were unable to recruit 100 participants, so when we applied the majority voting we got some border-line cases, e.g., having just two votes on a given account, one participant labeled it as "Related" while the other one labeled it as "Non-Related". Another case was when we had 3 different votes for the same account because we have three possible options on the labeling interface for each account ("Related", "Not Related" and "Not Sure"). To address those border-line cases, the first co-author of the paper looked at them himself to make a final decision. After that, we applied majority voting on the labeled dataset. As a result, we obtained 987 cyber security related accounts and 231 non-related ones. To balance the final dataset, we added 756 additional non-related accounts, which were randomly selected from the original harvested raw data (see Section IV-C) after manual inspection, leading to a balanced dataset with 1,974 samples (987 in each class).

VI. Building Machine Learning Classifiers

A. Feature Extraction

Since we wanted to build several classifiers, we identified a rich set of (63 types of) features (listed in Table I). Each type of features measures one potentially useful aspect of a Twitter account to train and test our machine learning models. Most of the feature types are simple (i.e., contains just one single feature) while others include a group of features sharing a particular attribute. We arranged the features into 5 larger groups, namely, Profile (**P**), Behavioral (**B**), Content Statistics (**C**), Linguistic (**L**), and Keyword-based (**K**) features. We explain all the features in each group as follows.

*1) **Profile Features**:* They were extracted and calculated using the profile fields of each Twitter account. We divide the features into four categories, **Screen_name** (i.e., Twitter username), **Description**, **Network**, and **Miscellaneous** (Misc). For the Screen_name and Description categories, we calculated the field's length, the number of different types of characters (e.g., Alphabetic, Lowercase, Uppercase, Numerical, and Special). For the description field, we calculated the number of control characters (i.e. Non-Printing Characters, NPC) as we noticed a lot of Hackers' Twitter accounts usually use them in their description. Additionally, we calculated the number of words and cyber security keywords found in the description (i.e., KiD).

*2) **Behavioral Features**:* They cover three aspects, statistics about the account's tweets, the interaction that happens between Twitter users, and the account's general activity patterns on Twitter. We divide the features in this group into three categories: **Tweets Statistics**, **Network**, and **Activity**. Each category covers a different aspect of the account's behavior and interaction with other accounts.

The purpose of the Network features in this group is to represent the interaction between Twitter accounts in a few basic measures. For example, an account can post a tweet, retweet or like a tweet, comment on a tweet, reply to another account, or mention another account in a tweet or comment. All of these actions can be seen as interactions between two Twitter accounts.

*3) **Content Statistics Features**:* They were extracted from the content of an account's timeline. Currently, the Twitter API allows the retrieval of up to 3,250 tweets of an account's timeline. First, we have Keywords Statistics, where we calculated some measures about the cyber security keywords found in an account's timeline. Those keywords were obtained from the general cyber security taxonomy [16]. Second, we have the Readability metrics, which include SMOG (Gobbledygook SMOG score) score, Flesch-Kincaid reading grade level, and finally the Lexical Diversity (which is a simple measure to show the ratio of the unique words in an account's timeline [17]).

*4) **Linguistic Features**:* Linguistic Inquiry and Word Count (LIWC) [18] is a well-established and widely used method for text analysis. For example, LIWC measures were used to analyze posts from underground forums and hacking websites [19]. We used LIWC 2015 Edition v16 to analyze an account's timeline, leading to 93 features representing different linguistic characteristics.

*5) **Keyword-based Features**:* They include a number of sub-groups, each being a set of features defined by a given keyword and a specific metric of the keyword in a given account's timeline. Some metrics require text corpora reflecting the domain of interest. To this end, we prepared two text corpora using the labeled dataset, where we merged all the Twitter timelines of the cyber security related accounts to form one corpus and we applied the same process to non-cyber security related accounts to create another corpus. Each Twitter timeline was pre-processed and cleaned to remove stop words, URLs, email addresses, punctuation marks, screen_names, and other Twitter-related symbols (e.g., RT, #, @). Then, uni-grams and bi-grams were extracted along with their frequencies per timeline and corpus. After that, we calculated several metrics detailed in Appendix A. and used each of them to rank all candidate keywords, from which we selected the top k from each domain to form a list of $2k$ keywords per each metric. Any frequencies used in such metrics are normalized per timeline to allow comparison across accounts. We chose $k = 100$ as a reasonable number of top-ranked keywords, considering a number of factors such as the size of our dataset and the need to reduce the total number of features for our classifiers.

B. Machine Learning Models

For our experiments, we used the labeled dataset to train and test different machine learning models for the four classifiers, so we can determine which model is the best with what feature sets. We chose the following standard models widely used for similar classifiers: Decision Tree, Random Forests (with 100 estimators / trees), SVM, and Logistic Regression. For SVM, we used two different kernels: the Radial Basis Function (RBF) kernel and the Linear one.

C. Classification Tasks

Using our labeled dataset, we created one base-line classifier (Task 1) and three other sub-classifiers (Tasks 2-4) that should be applied after the base-line classifier. For each sub-classifier, a sub-dataset was created using the assigned tags from the labeling experiment.

1) Task 1: Detecting Cyber Security Related Accounts: This is the base-line classifier where we want to detect whether a Twitter account is cyber security related or not.

2) Task 2: Detecting Cyber Security Related Individual Accounts: Another classification task that we considered is the detection of cyber security accounts belonging to individuals and those representing non-individuals such as groups, companies, NGOs, etc. This classifier should be used after the base-line classifier. Thus, all the accounts in the Individual sub-dataset must be cyber security related accounts. In our dataset, we have 542 samples labeled as Individual accounts and 448 as non-individual ones.

3) Task 3: Detecting Hacker-related Accounts: We wanted to create a classifier that can detect if a Twitter account is affiliated to a hacker (an individual or a group) or acting like a hacker. The labeling experiment interface has several tags that correspond to the different

TABLE I: List of all features used

Profile Features (P)			Behavioral Features (B)			Content Statistics Features (C)		
Screen Name	F01	LEN (screen name)	Tweets Statistics	F26	CNT (Tweets)	Cyber Security Keywords Statistics	F48	CNT (Keywords)
	F02	CNT (Alphabetic char)		F27	CNT (Original tweets)		F49	CNT (Keywords) [Original tweets]
	F03	CNT (Lowercase char)		F28	CNT (Retweets)		F50	CNT (Unique keywords)
	F04	CNT (Uppercase char)		F29	CNT (Replies)		F51	CNT (Unique keywords) [Original tweets]
	F05	CNT (Numerical char)		F30	CNT (Tweets with mentions)		F52	CNT (Tweets with keywords)
	F06	CNT (Special char)		F31	Ratio (Original tweets to all)		F53	Ratio (Tweets with keywords to all)
Description	F07	LEN (description)		F32	Ratio (Retweets to all)	Readability & Diversity	F54	Flesch-Kincaid Score
	F08	CNT (Alphabetic char)		F33	AVG (Number of mentions)		F55	SMOG Index
	F09	CNT (Lowercase char)		F34	AVG (Number of hashtags)		F56	Lexical Diversity
	F10	CNT (Uppercase char)		F35	AVG (Number of URLs)			
	F11	CNT (Numerical char)	Network	F36	CNT (Tweets received likes)	**Linguistic Features (L)**		
	F12	CNT (Special char)		F37	CNT (Tweets were retweeted)	LIWC	F57	Measures L01, …, L93
	F13	CNT (Control char)		F38	CNT (Mentioned users)			
	F14	CNT (Words)		F39	CNT (Replied-to users)	**Keyword-based Features (K)**		
	F15	CNT (Keywords)		F40	CNT (Likes given)	Keywords Frequencies	F58	Weirdness Score
Network	F16	CNT (Friends)		F41	CNT (Likes received)		F59	Prototypical Words
	F17	CNT (Followers)		F42	CNT (Retweets received)		F60	TF-IDF Score
	F18	Followers/Friends	Activity	F43	AVG (Daily Tweets)		F61	User Count (UC)
Misc	F19	Profile Image used?		F44	AVG (Weekly Tweets)		F62	Hybrid Metric UC-IDF
	F20	Profile Theme used?		F45	AVG (Monthly Tweets)		F63	Hybrid Metric UC-TFIDF
	F21	Location provided?		F46	AVG (time between tweets)			
	F22	CNT (Lists)		F47	STD (time between tweets)			
	F23	Account protected?						
	F24	URL provided?						
	F25	Account Age						

LEN	Length
CNT	Count
AVG	Average
STD	Standard Deviation

types of hackers, e.g., White-Hat, Grey-Hat, and Black-Hat hackers. Also, there is a general tag "Hacker" to make it easier for participants. We got 166 accounts labeled as Hackers and we added randomly (yet manually checked) another 166 general cyber security accounts from the main dataset to make the Hacker sub-dataset balanced.

4) Task 4: Detecting Cyber Security Accounts Related to Academia: The purpose of this classifier is to detect if a cyber security related account is for someone (or a group) in academia as a sector. The correspondent tags from the labeling experiment are Student, Lecturer, Researcher, and a general tag "Academia". We got 129 Academia accounts, and we added another 129 general cyber security related to make the Academia sub-dataset balanced.

VII. Experimental Results

We used Scikit-Learn (https://scikit-learn.org/), the widely used machine learning library in Python, for our experiments. All the used machine learning models were trained and tested using 5-fold stratified cross-validation. The results were reported using four performance metrics: Accuracy, F1 score, Precision and Recall. The experimental results for all the classifiers are shown in Table II. Each row in the table is for a feature set for a specific classifier, Column #F shows the total number of features, and Column #S is the number of samples used for each feature set. Note that for some feature sets in the base-line classifier (Task 1), the number of samples is smaller than 1,974 because not all samples include the required features, and so on for Tasks 2-4. The keyword-based feature sets have a prefix **K** before their names.

A. Base-Line Classifier

For the base-line classifier, we used different feature sets based on the feature groups we described in Section VI-A. We experimented many feature sets resulted from either one feature group or a mixture of two or more feature groups. The purpose is to see the impact of selecting different groups of feature sets on the results.

The best performance achieved in terms of F1 score using behavioral features is 77%, while the figure is 86% for Profile features and 88% for Linguistic features. For the Content Statistics features, the best performance achieved is 93%, which is surprisingly good considering it is a small feature group with just 9 features. As for the keyword-based features, the best performance was achieved for the UC sub-group with 93%, followed by the UC-TFIDF sub-group with 90%. When all keyword-based features are together (K_ALL), the performance is also 93%, which is likely due to the UC sub-group.

For the mixed feature sets, we tried different combinations of all feature groups we have. For example, PBC refers to P, B, and C feature groups combined together, and PBCLK_ALL means all features together. The results showed that such combined feature sets generally performed well, with an F1 score between 92-93%. Considering C features alone can already achieve an F1 score of 93%, we consider such combined feature sets unnecessary.

In terms of the five machine learning models, looking at the general patterns shown in Table II, we can see that the Random Forest model is the only model achieving the best performance for all feature sets. The other four models also achieved good performance for many feature sets, but did not perform very well for some feature sets. As an overall conclusion, we recommend using Random Forest and the 9 Content Statistics features for the base-line classifier.

B. "Individual" Sub-Classifier

As for the "Individual" sub-classifier, we examined also different feature sets as seen in Table II. Among all feature groups, the Linguistic features (L) performed the best with an F1 score of 89%. The best performance however was achieved by a combined feature set PBCL, with an F1 score of 90%. The best-performing model is Random Forest across all feature sets.

C. "Hacker" Sub-Classifier

For the "Hacker" sub-classifier, we found that non-keyword feature sets did not perform well, with the highest F1 score being just 69% for the PBCL feature set. For the keyword-based features, we found features based on the UC-IDF metrics gave a much better F1 score of 88%. The best-performing model is Random Forest, as well.

TABLE II: Overall experimental results for the four classifiers.
(For some classifiers we only show the best-performing sub-groups of keyword-based features to save space)

Task	Feature set	#F	#S	Decision Tree				Random Forest				Logistic Regression				SVM (Linear)				SVM (RBF)			
				Accu	F1	Prec	Rec	Accu	F1	Prec	Rec	Accu	F1	Prec	Rec	Accu	F1	Prec	Rec	Accu	F1	Prec	Rec
Related	P	25	1974	0.78	0.78	0.79	0.77	0.85	0.86	0.81	0.91	0.82	0.82	0.82	0.82	0.84	0.84	0.82	0.87	0.84	0.84	0.83	0.86
	B	22	1974	0.71	0.71	0.71	0.70	0.77	0.77	0.75	0.79	0.74	0.73	0.76	0.70	0.74	0.73	0.76	0.71	0.74	0.74	0.74	0.74
	C	9	1882	0.89	0.89	0.90	0.89	0.92	0.93	0.90	0.95	0.92	0.92	0.94	0.90	0.92	0.92	0.92	0.93	0.92	0.93	0.91	0.95
	PBC	56	1974	0.88	0.88	0.88	0.88	0.92	0.92	0.90	0.95	0.91	0.90	0.92	0.89	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.90
	L	93	1882	0.80	0.81	0.80	0.82	0.87	0.88	0.87	0.88	0.85	0.86	0.85	0.88	0.87	0.88	0.86	0.90	0.87	0.88	0.86	0.90
	PBCL	149	1974	0.88	0.88	0.88	0.87	0.91	0.92	0.89	0.94	0.91	0.91	0.92	0.90	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91
	K_WEIRD	200	1885	0.81	0.79	0.90	0.70	0.83	0.82	0.91	0.74	0.52	0.68	0.52	1.00	0.51	0.68	0.51	1.00	0.57	0.70	0.55	0.97
	K_PROTO	200	1885	0.68	0.55	1.00	0.38	0.68	0.56	1.00	0.39	0.51	0.68	0.51	1.00	0.51	0.68	0.51	1.00	0.51	0.68	0.51	1.00
	K_TFIDF	200	1885	0.66	0.51	1.00	0.34	0.66	0.51	1.00	0.35	0.51	0.68	0.51	1.00	0.51	0.68	0.51	1.00	0.52	0.68	0.52	1.00
	K_UC	199	1885	0.87	0.87	0.88	0.87	0.93	0.93	0.91	0.96	0.88	0.88	0.90	0.87	0.62	0.73	0.58	1.00	0.91	0.91	0.90	0.93
	K_UC-IDF	200	1885	0.85	0.83	1.00	0.71	0.87	0.85	1.00	0.74	0.51	0.68	0.51	1.00	0.51	0.68	0.51	1.00	0.79	0.74	0.98	0.60
	K_UC-TFIDF	196	1885	0.87	0.87	0.87	0.87	0.90	0.90	0.89	0.92	0.76	0.81	0.68	0.99	0.51	0.68	0.51	1.00	0.89	0.89	0.90	0.88
	K_ALL	903	1885	0.87	0.87	0.88	0.86	0.92	0.93	0.91	0.95	0.88	0.88	0.90	0.87	0.63	0.73	0.58	1.00	0.90	0.90	0.90	0.91
	PBCLK_ALL	1052	1885	0.88	0.89	0.89	0.88	0.93	0.93	0.90	0.97	0.91	0.92	0.91	0.92	0.92	0.92	0.91	0.93	0.91	0.92	0.91	0.93
Individual	P	25	957	0.65	0.67	0.68	0.67	0.76	0.78	0.78	0.79	0.76	0.79	0.77	0.81	0.77	0.79	0.78	0.80	0.75	0.77	0.77	0.78
	B	22	957	0.76	0.78	0.78	0.78	0.82	0.83	0.85	0.81	0.80	0.81	0.82	0.81	0.80	0.81	0.83	0.78	0.81	0.81	0.86	0.78
	C	9	937	0.70	0.73	0.72	0.74	0.78	0.79	0.81	0.77	0.79	0.81	0.78	0.84	0.79	0.81	0.79	0.83	0.80	0.82	0.82	0.81
	PBC	56	957	0.77	0.79	0.78	0.80	0.85	0.85	0.89	0.82	0.85	0.87	0.86	0.87	0.85	0.86	0.87	0.86	0.85	0.86	0.88	0.84
	L	93	937	0.83	0.84	0.84	0.84	0.88	0.89	0.92	0.86	0.86	0.86	0.92	0.82	0.85	0.85	0.91	0.81	0.85	0.85	0.93	0.78
	PBCL	149	957	0.82	0.84	0.83	0.84	0.89	0.90	0.92	0.87	0.89	0.89	0.91	0.88	0.89	0.89	0.91	0.88	0.87	0.88	0.92	0.84
	K_UC	129	939	0.74	0.76	0.75	0.77	0.82	0.83	0.88	0.78	0.54	0.70	0.54	0.98	0.54	0.70	0.54	1.00	0.80	0.81	0.85	0.77
	K_UC-IDF	200	939	0.80	0.77	1.00	0.63	0.84	0.83	1.00	0.71	0.54	0.70	0.54	1.00	0.54	0.70	0.54	1.00	0.63	0.74	0.60	0.99
	K_UC-TFIDF	152	939	0.71	0.73	0.72	0.74	0.81	0.82	0.86	0.79	0.54	0.70	0.54	0.98	0.54	0.70	0.54	1.00	0.81	0.82	0.84	0.79
Hacker	P	25	317	0.53	0.53	0.55	0.53	0.62	0.62	0.62	0.63	0.68	0.68	0.69	0.68	0.67	0.66	0.68	0.65	0.66	0.66	0.67	0.65
	B	22	317	0.53	0.54	0.53	0.57	0.60	0.61	0.60	0.62	0.64	0.67	0.62	0.72	0.62	0.66	0.60	0.73	0.62	0.65	0.61	0.71
	C	9	313	0.54	0.54	0.54	0.55	0.55	0.54	0.55	0.54	0.65	0.67	0.63	0.72	0.61	0.67	0.58	0.79	0.64	0.67	0.62	0.72
	PBC	56	317	0.56	0.59	0.56	0.63	0.61	0.63	0.61	0.65	0.66	0.66	0.65	0.67	0.66	0.67	0.66	0.68	0.66	0.68	0.65	0.71
	L	93	313	0.61	0.60	0.61	0.59	0.68	0.68	0.68	0.69	0.66	0.66	0.66	0.66	0.65	0.66	0.64	0.68	0.67	0.66	0.69	0.65
	PBCL	149	317	0.57	0.57	0.58	0.57	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.70	0.69	0.69	0.68	0.70	0.65	0.64	0.65	0.64
	K_UC-IDF	200	314	0.80	0.81	0.82	0.86	0.88	0.88	0.89	0.89	0.50	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.56	0.23	0.89	0.14
Academia	P	25	249	0.50	0.46	0.50	0.43	0.48	0.44	0.47	0.42	0.44	0.40	0.44	0.37	0.44	0.38	0.43	0.37	0.43	0.33	0.42	0.32
	B	22	249	0.59	0.60	0.59	0.64	0.63	0.64	0.62	0.67	0.60	0.60	0.61	0.60	0.64	0.65	0.64	0.68	0.63	0.64	0.63	0.65
	C	9	243	0.58	0.57	0.57	0.58	0.60	0.60	0.60	0.61	0.65	0.66	0.62	0.70	0.64	0.68	0.60	0.77	0.66	0.66	0.65	0.68
	PBC	56	249	0.57	0.57	0.56	0.59	0.64	0.65	0.63	0.68	0.63	0.63	0.64	0.63	0.61	0.62	0.61	0.63	0.62	0.62	0.62	0.62
	L	93	243	0.56	0.52	0.55	0.50	0.67	0.66	0.66	0.67	0.63	0.62	0.62	0.62	0.66	0.66	0.64	0.70	0.65	0.66	0.63	0.69
	PBCL	149	249	0.61	0.63	0.61	0.64	0.63	0.64	0.63	0.66	0.63	0.62	0.63	0.62	0.64	0.64	0.64	0.64	0.65	0.64	0.65	0.64
	K_UC-IDF	200	245	0.79	0.83	0.71	1.00	0.89	0.91	0.83	1.00	0.51	0.00	0.00	0.00	0.51	0.00	0.00	0.00	0.58	0.51	0.73	0.63

D. "Academia" Sub-Classifier

For the "Academia" sub-classifier, we tried all the feature sets and the best-performing was the keyword-based features defined by the UC-IDF metric, with 91% for F1, similar to the case of the "Hacker" sub-classifier. The best-performing model is still Random Forest.

E. Features Importance

As we have many features for each classifier, we were interested in knowing which features contribute more to the classification task. The information will help produce even more optimized classifiers with hopefully a smaller number of important features only, while keeping the same or similar overall performance. Here, we give an example of the PBCLK feature set for the base-line classifier.

To calculate the feature importance, we used the χ^2 feature selection method [20]. The rankings of the top 20 ranked features in the PBCLK feature set are shown in Figure 2a. We can notice that the top four features are all Content Statistics features based on simple aggregated statistics of cyber security related keywords, which is not surprising. For example, Feature F53 – ranked at the first position – is the ratio of tweets with cyber security related keywords to the whole collected account timeline. Similarly, Feature 50 – ranked second – counts the number of unique cyber security related keywords found in an account's timeline. Another interesting feature – ranked fifth – is F15, which is the number of cyber security related keywords in an account's profile description.

For the importance of L features used for the "Individual" sub-classifier, Figure 2b lists the top 20 features ranked by χ^2. It is interesting to see that the top-ranked feature is L12, which is the "i" variable from LIWC reflecting the usage of words that correspond to the first person singular (e.g., "I", "me", "my" and "mine").

F. Features Reduction

Even though for the base-line classifier, the best-performing feature set (C) has only 9 features, this is not the case for other sub-classifiers. The different feature importance as discussed in the previous subsection motivated us to look at reducing the number of features for all classifiers to make them more lightweight.

Using the χ^2 feature selection algorithm again, we tried to identify the smallest feature set from the most complete feature set PBCLK for the base-line classifier. Based on the feature importance scores, we select the top m features with the highest χ^2 scores, and then train the Random Forest classifier again to see its performance. We evaluated how accuracy and F1 change as adding more features (until top 51 features), and the results can be seen in Figure 3.

Feature	$\chi 2$
F53	312.75
F50	218.40
F51	182.72
F52	157.80
F15	114.53
L40	66.16
F08	62.23
L77	60.24
L62	58.22
F09	57.76
L12	54.00
L53	53.28
L39	51.25
L23	47.29
F35	46.35
L52	46.29
L35	44.96
F42	41.64
L04	41.25
L15	38.19

(a) The top 20 PBCLK features of the base-line classifier

Feature	$\chi 2$
L12	36.61
L47	32.24
L23	28.13
L17	23.11
L04	22.16
L63	19.25
L62	14.31
L44	13.99
L45	13.12
L80	12.44
L24	11.96
L20	11.88
L40	11.55
L10	11.26
L21	10.15
L41	10.02
L03	7.46
L09	7.37
L77	7.29
L13	7.12

(b) The top 20 L features of the "Individual" sub-classifier

Fig. 2: The top 20 features of two classifiers, ranked according to their χ^2 significance values in the decreasing order. Here, Li means the i-th feature in the F57 feature group (LIWC).

The model achieved the highest F1 score (93%) after reaching only 6 features, even 3 less than the previously considered best feature set C. The performance is largely saturated after the sixth-best features, indicating that adding more features is not actually helpful. Having just 6 features will significantly improve the efficiency of the base-line classifier, without compromising its effectiveness at all. A similar process of identifying the minimum set of important features can also be applied to the three sub-classifiers.

G. Comparison to Related Work

The most relevant work for our base-line classifier from the literature was the work reported in [4], where Aslan et al. achieved an F1-score over 97% using the Random Forest model and for several different feature sets. The performance difference is more likely due to the different datasets we used. Note that the dataset used by Aslan et al. is much smaller and also more likely biased since it was constructed following a more ad hoc approach. After contacting Aslan et al., we obtained their dataset and ran a comparison of our base-line classifier and their classifier on both their dataset and our own one, leading to the finding that our base-line classifier performed better as a whole.[2]. In addition to the better performance of our base-line classifier, our work also goes beyond [4] by providing three sub-classifiers and a larger and more systematically constructed dataset.

Another related work is [2], where Jones et al. developed a machine learning classifier to detect Twitter accounts affiliated with the Anonymous group. Their best-performing model is also Random Forest, with an F1 score of 94%. While their performance is higher than our hacker classifier (88%), ours is much more general so it was expected to be less accurate. Jones et al.'s work is based on ad hoc features defined for the Anonymous group only, so cannot be directly generalized to detect general hackers or hacking groups.

[2]We discovered some mistakes in the reported results in [4], leading to lowered performance figures for their classifiers, especially for the behavioral features. We discussed our findings with authors of [4] and confirmed our findings. Our performance comparison with Aslan et al.'s work was based on their corrected performance figures.

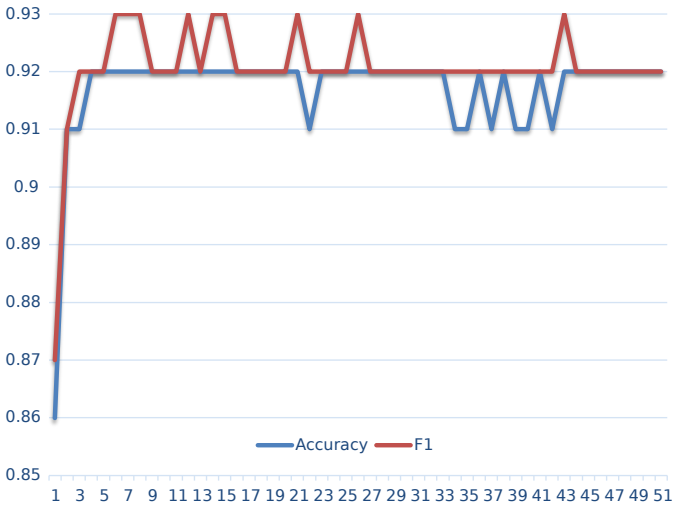


Fig. 3: Feature reduction analysis for the base-line classifier, showing how performance changes as we add more features

VIII. Limitations and Future Work

There are some limitations in the reported work that we plan to address in our future work. Using a list of cyber security related n-grams – derived from the cyber security taxonomy in [16] – was necessary to filter the huge data collection we harvested from Twitter. However, while being representative, this list is far from complete, especially for such fast-evolving domains as cyber security where new concepts and terms keep appearing. We plan to help refine the cyber security taxonomy to make it more dynamically updated, and then use it to update the classifiers developed.

The recruited participants were not as diverse as we wished, considering different demographic factors such as gender, age, employment status, level of cyber security experience, and educational level. For example, only about 25% of the participants were females. We plan to conduct future experiments to recruit more participants to enlarge our dataset with a more diverse participation pool.

Our work focused on English terms and tweets only. However, many cyber security related accounts use other languages on OSNs as well. Therefore, to study such accounts and their activities conducted in other languages, we have to consider supporting multiple languages in the classifiers. The methodology presented in this paper can be extended and generalized to support other languages.

In this work, we used five standard machine learning models without trying to fine-tune their configurations or parameters. We did not try other models, especially those based on neural networks or deep learning. An immediate follow-up work would be to investigate if other models such as deep learning based ones can improve the classification performance further.

Last but not the least, as we mentioned in the Introduction, the classifiers we developed can be used to analyze cyber security related accounts on OSNs for many different purposes. One example application for OSINT (open source intelligence) is to use the Hacker sub-classifier to help identify more hacker-related accounts so we can study more hacker-related phenomena beyond what has been reported [2, 3]. A second example is to automatically detect and monitor different types of cyber security related accounts to collect cyber threat intelligence. A third example is about using the classifiers to study cyber security influencers on Twitter. Having multiple classifiers will allow us to detect influencers and influencees belonging to

different sub-communities of the cyber security community, therefore allowing us to gain more insights about how such influence is formed and spread across an OSN platform and members of different (sub-)communities.

IX. Conclusion

This paper reports our work on creating a dataset for the development of four machine learning based classifiers for detecting cyber security related accounts and different sub-groups of such accounts (individual accounts, hacker-related accounts, and accounts belonging to academia). A general three-staged methodology was proposed to ensure the dataset is more representative and accurate, using a cyber security taxonomy and a crowdsourcing-based labeling experiment. We trained and tested the four classifiers with five machine learning models using the dataset and 63 types of features in 5 larger groups (with a total number of over 1,000 features).

Our results showed that the Random Forest model is the best machine learning model for all four classifiers, with the best F1 score ranging from 88-93%. We also investigated the importance of different features and found that only a very small number of features (e.g., 6 for the base-line classifier) are already sufficient to produce a lightweight classifier with the same best performance.

Acknowledgments

Mohamad Imad Mahaini was funded by the European Union's Horizon 2020 Research and Innovation program under the Marie Skłodowska-Curie NeCS project, Grant Agreement No 675320.

Shujun Li's work was partly supported by the research projects PriVELT and ACCEPT, funded by the EPSRC (Engineering and Physical Sciences Research Council) in the UK, under grant numbers EP/R033749/1 and EP/P011896/2, respectively.

References

[1] R. P. Lippmann, W. M. Campbell, D. J. Weller-Fahy, A. C. Mensch, G. M. Zeno, and J. P. Campbell, "Toward finding malicious cyber discussions in social media," in *Proceedings of AAAI 2017 Workshops*. AAAI, 2017, pp. 203–209.

[2] K. Jones, J. R. Nurse, and S. Li, "Behind the mask: A computational study of Anonymous' presence on Twitter," in *Proceedings of ICWSM 2020*. AAAI, 2020, pp. 327–338. [Online]. Available: https://aaai.org/ojs/index.php/ICWSM/article/view/7303

[3] C. B. Aslan, S. Li, F. V. Celebi, and H. Tian, "The world of defacers: Looking through the lens of their activities on Twitter," *IEEE Access*, vol. 8, pp. 204 132–204 143, 2020. [Online]. Available: https://doi.org/10.1109/ACCESS.2020.3037015

[4] C. B. Aslan, R. B. Sağlam, and S. Li, "Automatic detection of cyber security related accounts on online social networks: Twitter as an example," in *Proceedings of SM&Society 2018*. ACM, 2018, pp. 236–240. [Online]. Available: https://doi.org/10.1145/3217804.3217919

[5] M. Pennacchiotti and A.-M. Popescu, "A machine learning approach to Twitter user classification," *Proceedings of ICWSM 2011*, pp. 281–288, 2011. [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/14139

[6] E. Colleoni, A. Rozza, and A. Arvidsson, "Echo chamber or public sphere? predicting political orientation and measuring political homophily in Twitter using big data," *Journal of Communication*, vol. 64, no. 2, pp. 317–332, 2014. [Online]. Available: https://doi.org/10.1111/jcom.12084

[7] M. Pennacchiotti and A.-M. Popescu, "Democrats, republicans and Starbucks afficionados: User classification in Twitter," in *Proceedings of KDD 2011*. ACM, 2011, pp. 430–438. [Online]. Available: https://doi.org/10.1145/2020408.2020477

[8] C. Peersman, W. Daelemans, and L. Van Vaerenbergh, "Predicting age and gender in online social networks," in *Proceedings of SMUC 2011*. ACM, 2011, pp. 37–44. [Online]. Available: https://doi.org/10.1145/2065023.2065035

[9] L. Liu, D. Preotiuc-Pietro, Z. R. Samani, M. E. Moghaddam, and L. Ungar, "Analyzing personality through social media profile picture choice," in *Proceedings of ICWSM 2016*. AAAI, 2016, pp. 211–220. [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/14738

[10] D. Preoţiuc-Pietro, S. Volkova, V. Lampos, Y. Bachrach, and N. Aletras, "Studying user income through language, behaviour and affect in social media," *PLOS ONE*, vol. 10, no. 9, 2015. [Online]. Available: https://doi.org/10.1371/journal.pone.0138717

[11] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, "Detecting spammers on Twitter," in *Proceedings of CEAS 2010*, 2010. [Online]. Available: https://homepages.dcc.ufmg.br/~fabricio/download/ceas10.pdf

[12] R. Krithiga and E. Ilavarasan, "A comprehensive survey of spam profile detection methods in online social networks," *Journal of Physics: Conference Series*, vol. 1362, 2019. [Online]. Available: https://doi.org/10.1088/1742-6596/1362/1/012111

[13] M. Abulaish and M. Fazil, "Socialbots: Impacts, threat-dimensions, and defense challenges," *IEEE Technology and Society Magazine*, vol. 39, no. 3, pp. 52–61, 2020. [Online]. Available: https://doi.org/10.1109/MTS.2020.3012327

[14] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro, "Aiding the detection of fake accounts in large scale social online services," in *Proceedings of NSDI 2012*. USENIX, 2012. [Online]. Available: https://www.usenix.org/conference/nsdi12/technical-sessions/presentation/cao

[15] K. C. Lee, C. H. Hsieh, L. J. Wei, C. H. Mao, J. H. Dai, and Y. T. Kuang, "Sec-Buzzer: cyber security emerging topic mining with open threat intelligence retrieval and timeline event annotation," *Soft Computing*, vol. 21, no. 11, pp. 2883–2896, 2017. [Online]. Available: https://doi.org/10.1007/s00500-016-2265-0

[16] M. I. Mahaini, S. Li, and R. B. Sağlam, "Building taxonomies based on human-machine teaming: Cyber security as an example," in *Proceedings of ARES 2019*. ACM, 2019. [Online]. Available: https://doi.org/10.1145/3339252.3339282

[17] R. Aswani, A. K. Kar, and P. V. Ilavarasan, "Detection of spammers in Twitter marketing: A hybrid approach using social media analytics and bio inspired computing," *Information Systems Frontiers*, vol. 20, no. 3, pp. 515–530, 2018. [Online]. Available: https://doi.org/10.1007/s10796-017-9805-8

[18] Y. R. Tausczik and J. W. Pennebaker, "The psychological meaning of words: LIWC and computerized text analysis methods," *Journal of Language and Social Psychology*, vol. 29, 2010. [Online]. Available: https://doi.org/10.1177/0261927X09351676

[19] J. McAlaney, S. Hambidge, E. Kimpton, and H. Thackray, "Knowledge is power: An analysis of discussions on hacking forums," in *Proceedings of Euro S&P 2020 Workshops*. IEEE, 2020, pp. 477–483. [Online]. Available: https://doi.org/10.1109/EuroSPW51379.2020.00070

[20] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Proceedings of ICML '97*,

vol. 97. Morgan Kaufmann, 1997, pp. 412–420.
[21] K. Ahmad, L. Gillam, L. Tostevin, and A. Group, "University of Surrey participation in TREC 8: Weirdness indexing for logical document extrapolation and retrieval (WILDER)," 2000. [Online]. Available: https://trec.nist.gov/pubs/trec8/papers/surrey2.ps

APPENDIX A
KEYWORD-BASED FEATURES

The keyword-based features are described in detail below based on a number of metrics, one in each subsection.

A. Weirdness Score

Ahmad et al. [21] argued that the distribution of items (i.e., terms) will differ between a special corpus s and a general one g. Thus, they introduced the weirdness score, which is defined as follows:

$$\text{Weirdness}(w_i) = \frac{\text{TF}_s(w_i)}{\text{TF}_g(w_i)}, \quad (1)$$

where $\text{TF}(w_i)$ is the normalized frequency for the word w_i in a given domain[3] d and can be calculated as below:

$$\text{TF}_d(w_i) = \frac{F_d(w_i)}{t_d}, \quad (2)$$

where $F_d(w_i)$ is the number of times the word w_i appeared in the domain d, and t_d is the total number of words in that domain.

B. Prototypical Words

Prototypical words can be used as features for classification tasks as these words can describe the corresponding classes [7]. This means that the Twitter accounts in the cyber security related class will have a set of lexical expressions that are "typical" among cyber security related people and the same for the other class.

Suppose that we have n classes, and S_i are the seed Twitter accounts that belong to class c_i. Each word w will be assigned a proto score for each class using the following formula:

$$\text{proto}(w, c_i) = \frac{|w, S_i|}{\sum_{j=1}^{n} |w, S_j|}, \quad (3)$$

where $|w, S_i|$ is the total number the word w was issued across all users S_i in class c_i. The denominator in the formula above can not be zero for a given word if it was found at least once in any account's timeline. Also, to get a higher value for the proto score of a word, it should appear only in one domain, not both, according to the formula.

Before calculating the proto scores, we eliminated the words that have a frequency below 6 or less than 3 characters long. For our work, we selected the top k words from each class. The user u score for a particular prototypical word w_p can be given using this formula:

$$\text{f\_proto\_w}_p(u) = \frac{|u, w_p|}{\sum_{w \epsilon W_u} |u, w|}, \quad (4)$$

where $|u, w_p|$ is the frequency of w_p in the user u timeline, and W_u is the set of all words found in that timeline.

C. Term Frequency–Inverse Document Frequency (TF-IDF)

TF-IDF is commonly used in information retrieval tasks. However, it is also useful for keywords extraction to be used as features for machine learning models and was used for this purpose in the literature [1, 4]. TF-IDF is defined over a set of documents corresponding to corpora. It can be defined by the following formula:

$$\text{TFIDF}_d(w) = \text{TF}_d(w) \times \text{IDF}(w), \quad (5)$$

where $\text{TF}_d(w)$ is the normalized frequency of the word w in the document d, and $\text{IDF}(w)$ is the Inverse Document Frequency defined as follows:

$$\text{IDF}(w) = \log(\frac{N}{1 + N_w}), \quad (6)$$

where N is the number of documents. In our case, we have 2 classes, and the 2 corresponding corpora are the documents. Thus $N = 2$ and N_w is the number of documents that contain the word w.

D. User Count (UC)

Some of the keywords selected by TF-IDF or Prototypical metrics got a relatively high score, although they were only found in about 25% of all account timelines (i.e., documents[4]) in both corpora. In order to provide a new supplementary metric, we also calculated each keyword's user count (UC) score, which is the number of documents (i.e., timelines) in the domain corpus that this keyword appeared in at least once [20]. The UC score for a word w is defined as follows:

$$\text{UC}(w, d) = \frac{U(w, d)}{U(d)}, \quad (7)$$

where $U(w, d)$ is the number of users from the domain d (i.e., class) that have at least one occurrence of the word w in their timelines, and $U(d)$ is the total number of users in d.

E. Hybrid Metric UC-IDF

To further increase possible features, we also defined some hybrid metrics, one of them combining UC and IDF scores, which is the product of UC by IDF as illustrated below:

$$\text{UCIDF}(w, d) = \text{UC}(w, d) \times \text{IDF}(w), \quad (8)$$

where $\text{UC}(w, d)$ is the User Count score for the word w in the domain d, and $\text{IDF}(w)$ is the Inverse Document Frequency of the word w across both domains.

F. Hybrid Metric UC-TFIDF

Another hybrid metric we used is the combination of UC and TF-IDF, defined as follows:

$$\text{UCTFIDF}(w, d) = ((1 - \alpha) \times \text{TFIDF}_d(w)) + \alpha \times \text{UC}(w, d), \quad (9)$$

where α is a constant between zero and one ($0 < \alpha < 1$). Setting $\alpha = 0$ means that we ignore the UC part, while setting ($\alpha = 1$) will ignore the TF-IDF part. In our experiments, we found that 0.2 is the best value of α.

The results and performance of the used keywords metrics can be found in Section VII. To see the differences between these metrics in terms of the keywords generated, we show the top 20 keywords produced by applying each of the keyword metrics (as a ranking method) in Table III.

APPENDIX B
RECRUITED PARTICIPANTS STATISTICS

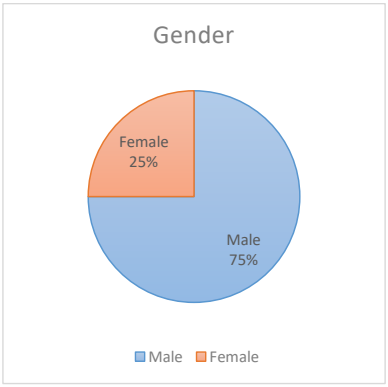
Here are some statistics about the recruited participants and their demographics, e.g., gender, age, employment status, cyber security experience, and level of education. See Figure 4 for more details.

[3]A "domain" represents a "class" of documents

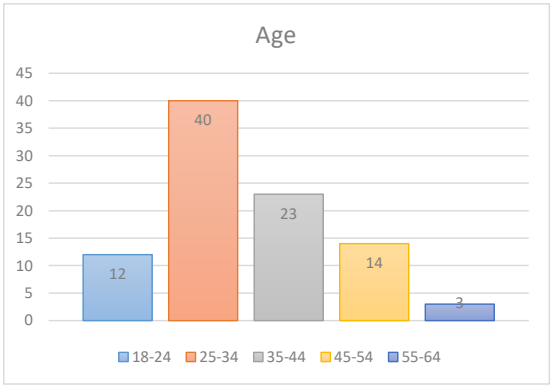
[4]In TF-IDF, A "Document" means the class's corpus, while here it means the Twitter account's timeline. All timelines in a given class form its corpus.

Weirdness	Prototypical	TFIDF	UC	UC-IDF	UC-TFIDF
threatpost	frama	frama	security	attack method	frama
securityaffair	vulnerabilitymanagement cybersecurity	vulnerabilitymanagement cybersecurity	time	apache strut	vulnerabilitymanagement cybersecurity
eprint	bugbounty appsec	bugbounty appsec	help	xxe	bugbounty appsec
bigdata security	appsec vulnerabilitymanagement	layer7datasolutions	data	xss vulnerability	layer7datasolutions
innovation science	pirate news	appsec vulnerabilitymanagement	day	ransomware malware	appsec vulnerabilitymanagement
trustyourinbox	ddo security	pirate news	check	android ransomware	ddo security
systems daily	shock daily	ddo security	attack	pay ransomware	pirate news
vulnerabilitymanagement	layer7datasolutions	cyberhoot	start	directory traversal	cyberhoot
osint security	cyberhoot	shock daily	free	bec scam	shock daily
securityaffair hacking	iot computer	iot computer	create	imperva	iot computer
secaas	layer7datasolutions msp	layer7datasolutions msp	team	desktop protocol	cyware
drericcole	computer ciso	cyware	learn	target windows	layer7datasolutions msp
dataprotection cybersecurity	cyware	computer ciso	user	security bulletin	computer ciso
hacking pentest	readcybernews	readcybernews	read	mining malware	readcybernews
avira	cybersecurity readcybernews	mikejulietbravo	people	execution flaw	mikejulietbravo
securityawareness	solutions daily	cybersecurity readcybernews	change	locky ransomware	cybersecurity readcybernews
releases security	msp secaas	solutions daily	system	cybersecurity regulation	solutions daily
shibboleth	exploit pack	opendir	support	bypass vulnerability	opendir
hacking osint	mikejulietbravo select	msp secaas	network	email threat	msp secaas

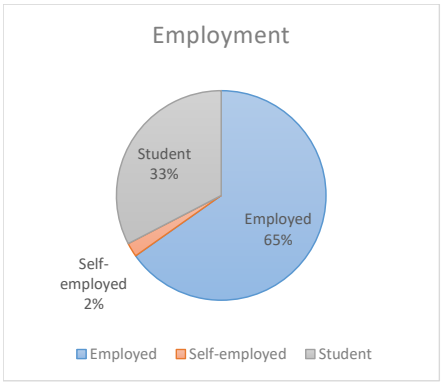
TABLE III: The top 20 keywords by each keyword metric from the cyber security corpus



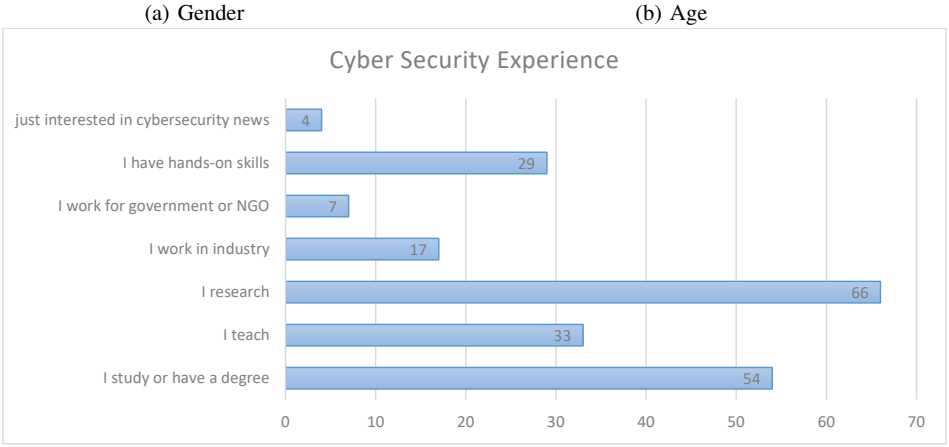
(a) Gender



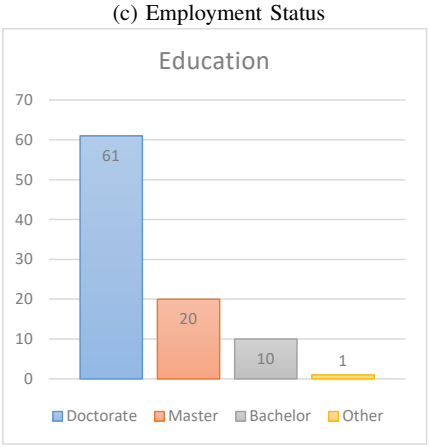
(b) Age



(c) Employment Status



(d) Experience



(e) Education

Fig. 4: Participants statistics