A Peer-Based Approach on Analyzing Hacked Twitter Accounts

Benjamin Murauer
University of Innsbruck
benjamin.murauer@uibk.ac.at

Eva Zangerle
University of Innsbruck
eva.zangerle@uibk.ac.at

Günther Specht
University of Innsbruck
guenther.specht@uibk.ac.at

Abstract

Social media has become an important part of the lives of their hundreds of millions of users. Hackers make use of the large target audience by sending malicious content, often by hijacking existing accounts. This phenomenon has caused widespread research on how to detect hacked accounts, where different approaches exist. This work sets out to analyze the possibilities of including the reactions of hacked Twitter accounts' peers into a detection system. Based on a dataset of six million tweets crawled from Twitter over the course of two years, we select a subset of tweets in which users react to alleged hacks of other accounts. We then gather and analyze the responses to those messages to reconstruct the conversations made. A quantitative analysis of these conversations shows that 30% of the users that are allegedly being hacked reply to the accusations, suggesting that these users acknowledge that their account was hacked.

1. Introduction

Twitter is a popular microblogging service that allows users to write short messages, called tweets, which may not exceed a length of 140 characters. The last official numbers from May 2015 state that approximately 500 million tweets were sent every day and consumed by a total of 310 million active users [1].

Twitter's popularity in terms of the number of tweets sent and the number of users active on the platform every day makes it an interesting target for cyber-criminals. In principle, cybercriminals aim to hack Twitter accounts to spread spam which contain URLS leading to e.g., phishing websites over these accounts [2], [3]. On the Twitter platform, tweets appear on the timeline of all followers of a particular user. Therefore, cybercriminals aim at hacking accounts with an already established social network (i.e., accounts with a substantial number of followers) which allows for directly delivering spam messages to a multitude of users. More importantly, in online social networks such as Twitter, a notion of trust is created between users and their followers—regardless of whether users actually know each other in real life or not [4]. This trust among users is exploited by cybercriminals as people are more likely to click on links sent by trusted peers [5], which naturally is a desirable property for hackers to exploit. Therefore, cybercriminals are more likely to hack into accounts with an existing social network than creating own accounts, which are also more prone to be detected earlier [2], [6].

Heymann et al. find that there are three different types of countermeasures that may be used to cope with spam in social networks: (i) detection, (ii) demotion and (iii) prevention [7]. During the last years, Twitter has developed mechanisms for detecting accounts which are used for spreading spam [2]. Thomas et al. found that these mechanisms allow for detecting 77% of all Twitter accounts which are used to spread spam within the first day of having started to send out malicious contents and 92% of all accounts are detected (and subsequently suspended) within three days [8], [9]. Approaches for detecting spam, spamming accounts and hence, compromised accounts include information about the content of the messages themselves as well as other meta-information, such as the amount of follower relationships a user has [10]–[13].

Zangerle et al. performed an analysis of the reactions of Twitter users once they found that their account has been hacked [14]. They find that 27% of those users change to a new account and 51% of those users send out a tweet stating that their account was hacked and apologize for any unsolicited tweets. However, little research is done on the reactions of the peers of allegedly hacked users who might even point those users to their hacked accounts (if the hacked user does not recognize that the account has been compromised and used for sending spam). We hypothesize that peers that use Twitter may be faster in detecting a hack by reading the posted (possible malicious) content. Therefore, we are interested in analyzing Twitter conversations revealing that users might be pointed to a compromised account by peers. Particularly, we aim to reliably reconstruct conversations on Twitter, where users point other users to the fact that their account might have been hacked (e.g., *"@AllRiseSilver is your twitter account hacked?"*) to analyze the behavior of the Twitter users taking part in these conversations. Subsequently, we

URI: http://hdl.handle.net/10125/41378
ISBN: 978-0-9981331-0-2

analyze these tweets to study to which extent such conversations might be suited to extract hints about hacked Twitter accounts and to perform a detailed analysis on how users behave in these conversations.

From a dataset of six million tweets collected over the course of two years, we extract a subset of tweets that suggest that peers point each other to the fact that the other's account has been hacked and also incorporate the reaction of the hacked user to these suggestions. We realize these analyses by using a supervised machine learning method. The responses of these alleged victims are then crawled and classified. Using this method, we show that 30% of the accused victims respond to the accusations, either confirming a hack or explaining the situation. We also find that 48% of all users that actually reply to these allegations, respond within the first hour after having received a tweet suggesting that their account might have been hacked.

The remainder of this paper is structured as follows. In Section 2, we first explain some background of Twitter and the problem of hacked accounts, including related work on this topic. Section 3 presents the dataset underlying the performed analyses. Section 4 presents the methods utilized for the analyses and Section 5 presents the results. Section 6 presents a discussion of the results and Section 7 concludes the paper.

2. Related Work

Wherever social communication is able to gather a large audience, misuse of the services is interesting for hackers. The scenarios of misusing Twitter include impersonation [15], the creation of fake accounts (so-called "sybils") [16], phishing [17], malware distribution [18] or spamming campaigns [19]. Thomas et al. [8] lay out that URLs that are posted on Twitter have a significantly higher likelihood of being followed than URIs in email spam, especially if the user account posting the link is trustworthy (e.g. in a follower relationship). They also emphasize that hackers favor taking control over existing accounts over creating dedicated spamming accounts, as this increases the trust between the hacker and the victim as well as creates new possible attack points, such as direct messages, which are difficult to analyze because they cannot be crawled using the public APIs. This finding is also confirmed by Kanich et al. [20]. However, it is more difficult to obtain information about direct messages since only the sending and receiving users have access to their content.

Generally, the detection of hacked accounts has been tackled from different perspectives. Methods used include many different aspects, using the content of the messages sent, geographical and timely information as well as the social connections on Twitter. Mostly, approaches for detecting spam relies on a multitude of features, including content features as similarity of tweet texts, social network information such as number of followers, and behavioral features such as the retweet ration [10], [12], [13], [21]–[23]. Lee et al. create social honeypots to analyze the behavior of cybercriminals. Based on the information collected, they propose a spam identification method [11], [13]. Also, social features of spamming accounts, the social network of cybercriminals have been studied [24], [25].

Twitter already has profound methods for detecting and disabling suspicious user accounts, but no details on its implementation are publicly available. The user guidelines merely point out several aspects that a user should note to avoid account suspension [9]. Furthermore, Twitter provides means to report spamming user accounts or to report individual spam tweets [26], [27]. Thomas et al. evaluated the effectiveness of Twitter's spam detection methods in 2011 and found that 77% of all spam accounts are detected within the first 24 hours and 92% of all spamming accounts are detected and suspended within three days [8].

Zangerle et al. [14] provide an analysis on how users whose account was compromised react publicly on Twitter. They found that 27% of the users change to a new account, whereas 51% of hacked users apologize for unsolicited messages and spam. However, the analyses at hand does not focus on the user whose account has been compromised, we rather focus on conversations with peers of this particular user. To the best of our knowledge, no other approaches so far focus on the peers of hacked users.

3. Dataset

In the following section, we describe the crawling methods utilized for the collection of the dataset underlying the analyses at hand. Subsequently, we present the main characteristics of the resulting dataset.

In principle, we require a dataset containing tweets about hacked accounts for the analyses to be performed. Therefore, we make use of the public Twitter Streaming API to gather such tweets. The Streaming API provides means for gathering tweets featuring given filter keywords and metadata associated with the individual tweets as JSON-objects [28]. As for the filter keywords used, we restrain the set of crawled tweets to those which contain both the keywords "account" and "hacked" as this method has already been applied by Zangerle et al. for a similar task [14]. Twitter restricts the number of tweets which can be crawled freely over its APIs to approximately 1% of all tweets being sent. Therefore, the number of tweets delivered is capped by a rate limit [29]. However, inspecting the number of

tweets crawled per day shows that the daily number of tweets is constantly well below this 1% mark and hence, this fact suggests that no tweets matching our filter criterion have been capped, ensuring a full coverage of tweets according to the specified filter criterion.

Applying the described crawling method, we were able to collect a total of 4.7 million tweets between November 2012 and October 2014. Table 1 depicts the main characteristics of the dataset. For all tweets gathered over the API, Twitter does not only provide the tweet itself, but also—in case of retweets—the original tweet which was retweeted. As we consider these original tweets valuable for the analyses as well, we extract these and add these to the dataset, which results in a total of 5,984,406 tweets. As can be seen from Table 1, 31.82% (1,495,325) of all tweets are retweets and 54.83% (3,281,005) feature at least one mention of another Twitter user.

Table 1: Dataset characteristics.

Characteristic	Amount
Tweets	4,698,845
Tweets incl. extracted retweets	5,984,406
Distinct authors	2,670,318
Retweets	1,495,325
Tweets containing mentions	3,281,005
Distinct Hashtags	120,690
Distinct URLs	216,318

The extraction of retweeted tweets from the data provided by the API revealed that those tweet texts may have been shortened during the process of retweeting due to Twitter's 140 character limitation for tweets. This behavior is showcased in the following example tweet: *"RT @wayfaringcalum: @Calum5SOS hi cal,I hope youre having fun in America!! If you happen to see this pretty please refollow me,someone hack…"*. As can be seen, the tweet no longer contains the word "hacked" as it was cut off the text of the original tweet due to the need to adhere to the 140 character limit. To still be able to also incorporate the full content of such tweets in our study, we have to fetch the full text of the original tweet in order to be able to reconstruct the cut off tweet content. Therefore, we extract the retweeted message's full content from the JSON-object of the retweeting tweet and add these to the dataset as well.

Based on this dataset, we firstly perform a prefiltering step before being able to perform the actual analyses. The required prefiltering steps and analysis methods utilized are described in the following section.

4. Methods

In the following section, we present the methods utilized for performing the analyses proposed.

In principle, we require a set of messages that suggest that a Twitter account might have been hacked to firstly be able to reconstruct conversations about hacked accounts and to subsequently classify the hacked users' responses. Therefore, the messages have to contain a mention tag of a user and an indication that the affected account is hijacked to be included in the analyses.

Starting from the dataset of crawled tweets, we perform the following analysis steps, which are depicted in Figure 1: (1) Clean the dataset by removing all messages that do not contain any mention. (2) Extract a subset of messages that actually suggest another Twitter account being hacked. (3) Remove the messages that mention users that are no longer active on Twitter. (4) Fetch tweets of the users that were mentioned directly following the tweet in which the users and the alleged hack were mentioned. (5) Classify the responses. In Figure 1, blue wavy blocks represent sets of tweets, the green rectangles labeled "ML" stand for the machine learning processes that are performed. The red boxes denote services provided by Twitter that we utilize over the according API. These five steps are explained in detail in the following section.

As each of the first three steps aims to narrow down the available data to a dataset only containing relevant messages required for the actual analyses, Table 2 shows the amount of messages left after having performed each of the steps. We list which filtering step is performed, followed by the amount of messages that were left afterwards. Also, we list the number of actually fetched timelines based on the information gathered during prefiltering steps 1-3.

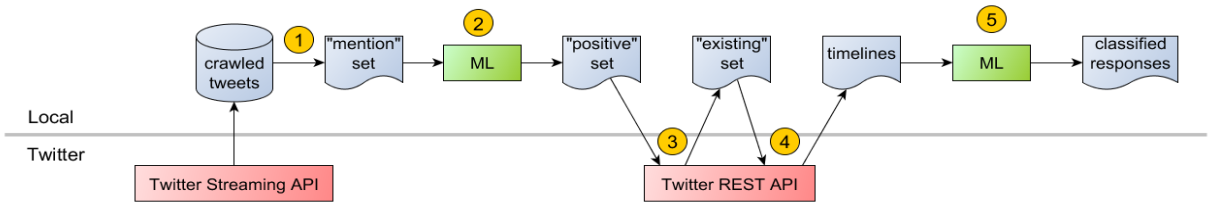

Figure 1: Workflow overview

Table 2: Sizes of intermediate message sets

Step	Message set	Amount
-	All	5,984,406
1	Mention	2,266,935
2	Positive	444,315
3	Existing	412,228
4	Fetched timelines	54,835

4.1. Removing tweets without mentions

As we require information about users who are mentioned within the tweets contained in our dataset to be able to reconstruct conversations about hacked accounts, the first step is to filter the dataset for all tweets which actually mention other users. A substantial amount of tweets within the dataset do not contain any mentions (e.g., when users state that their own account was hacked: "*Looks like my twitter account got hacked. I didn't lose 2.5lbs*").

The JSON file gathered over the Twitter API contains a separate field for any mention information in a tweet. This allows to extract information about the mentioned user account without having to parse the tweet text. We utilize this information to extract all tweets that actually contain a mention of another user to further be processed in the next step. Table 2 shows that out of the total 5,984,406 messages, 2,266,935 tweets (37.88%) are left to process in the next step.

4.2. Extracting tweets suggesting hacks

The word "hacked" can describe a variety of different scenarios, but for the performed analyses only suggestions of another Twitter account being hacked are relevant (e.g., *"@AllRiseSilver is your twitter account hacked?"*). We eliminate different other cases such as the mention of a different kind of account being hacked (e.g., *"@AmpersUK Looks like you Gmail account hasbeen hacked"*) by using supervised machine learning. Along the lines of Zangerle et al. [14] and also following research trends [16], we use a Support Vector Machine (SVM) [30] classifier with a linear kernel. Also, we utilize a Term-Frequency vs. Inverse Document Frequency (TF-IDF) [31] vectorizer to compute the feature vectors representing each message. The parameters of the classifier and vectorizer are determined using a grid search approach [32], which internally uses 5-fold cross validation. Table 3 shows the parameters that performed best for the SVM and the TF-IDF vectorizer when the grid search was optimized for the f1-score. We make use of the free python library *scikit-learn* [33] for all machine learning processes. As for linguistic features of the tweet texts being analyzed, we follow previous research [34], [35] and performed lemmatization [36] (i.e., we map all words to their basic word form).

Table 3: Best parameters for SVM and TF-IDF vectorizer

SVM parameter	**Best value**
C	1.0
TF-IDF parameter	**Best value**
n-gram size	(1, 3)
Max. document frequency	0.9
Min. document frequency	0.0001

Figure 2 depicts the workflow of the machine learning process. First, a subset of messages is manually classified (1). A TF-IDF vectorizer calculates the feature vectors (2). Our workflow includes a chi-square-test (3), which filters the feature vectors to leave the most significant ones. However, the overall performance is best if all features are used. The SVM is then trained based on the feature vectors (4) and subsequently predicts the classes of the remaining unclassified tweets (5). In Figure 2, the blue wavy boxes represent sets of messages. The green rectangles denote machine learning methods we utilize as provided by the scikit-learn toolkit [33]. Intermediate feature representations are displayed by yellow rhombs.

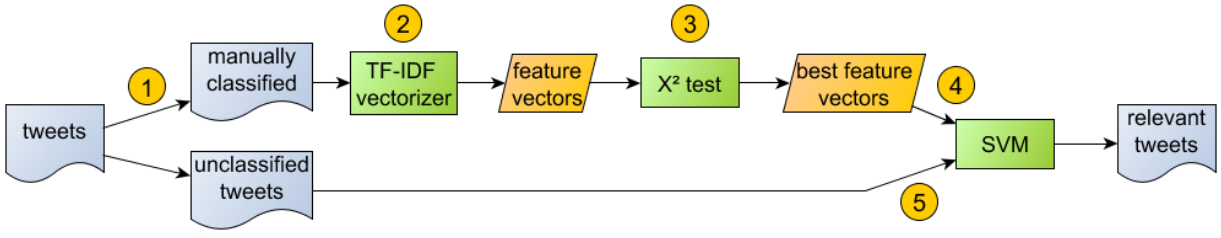

Figure 2: Machine learning process

Table 4: Confusion matrix

	Predicted positive	**Predicted negative**
Is positive	True positive (TP)	False positive (FP)
Is negative	False negative (FN)	True negative (TN)

To evaluate the performance of the classification, different methods can be used. Table 4 shows the confusion matrix of classification and depicts all possible combinations of a sample being classified. Based on this confusion matrix, traditional IR quality measures like precision and recall [36] may be defined. Precision is defined as $pr = \frac{TP}{TP+FP}$ and describes how many of the predicted samples are actually relevant. Recall is defined as $rec = \frac{TP}{TP+FN}$ and describes how many of the available relevant samples were classified as such. It can be seen that the two values on their own are not meaningful for our use-case, as they are easy to optimize. A classifier that predicts all samples as relevant has a perfect recall of 1.0, whereas a classifier that randomly guesses one positive sample as such and ignores all other has a precision of 1.0. Therefore, we require a combination of those two measures to evaluate the performance of the classifier and propose to utilize PR-curves as well as the f1-score as described in the following. The precision vs. recall curve (PR-curve) is a quality measure that visualizes how the precision and recall values change with varying discrimination thresholds of the classifier. The *f1*-score is the harmonic mean of precision and recall and is calculated as $f_1 = 2 \cdot \frac{pr \cdot rec}{pr+rec}$.

A classifier usually performs best if it is trained with the same amount of samples for each class [37], [38]. The training set consists of 3,650 manually classified messages that were randomly chosen from the data set, out of which 455 are labelled as relevant. To cope with the imbalance of classes, there are a number of approaches:

- *Sampling*: by removing overrepresented samples (undersampling) or duplicating existing underrepresented samples (oversampling), a balance in the class sizes can be achieved. In contrast to other sophisticated, domain-aware oversampling approaches like SMOTE [39], we used a blind copying approach where existing samples are randomly duplicated. Research on whether under- or oversampling yields better results are inconclusive [37], therefore we evaluated both of these methods.
- *Automatic class weighting*: we use the built-in weighting function of *scikit-learn*, which incorporates weights inversely proportional to the class frequencies into the classification process [40].

Figure 3 shows the precision vs. recall curve for the performed classification step. The *f1*-values for the four evaluated classification methods dealing with class imbalance (undersampling, oversampling, class weighting, no countermeasures taken) obtained by a 5-fold cross validation are listed in Table 5. As can be seen, none of the analyzed methods outperforms the plain imbalance-unaware classification. Therefore, we utilize the imbalance-unaware classification for step 2 of our analysis workflow.

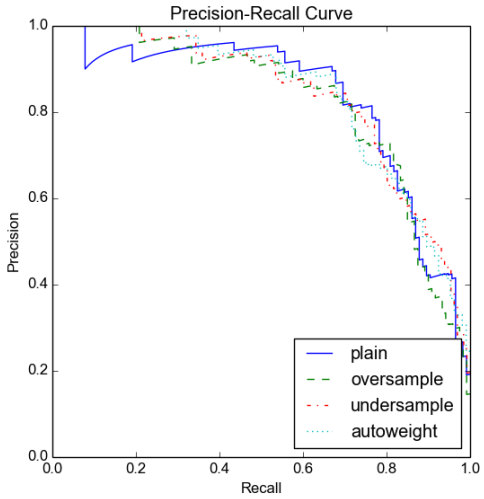

Figure 3: Precision vs. recall of imbalance countermeasures

Table 5: F1-scores of class imbalance countermeasures (step 2)

Method	**F1-score**
Plain	0.73
Undersample	0.72
Oversample	0.70
Class weighting	0.67

In the performed classification step, 444,315 messages were classified as relevant, which amounts to 19.6% of the tweets remaining resulting from step 1.

4.3. Removing inactive users

The next step in the analysis workflow aims to remove inactive users from the dataset to speed up the subsequent fetching step. The bottleneck of the analysis workflow in regards to computing time is the rate limitation of the Twitter API for step 4, which only allows fetching 200 messages at a time, with an additional limit of 180 requests per 15-minute time slot [41]. Therefore, we aim to keep the number of API-

calls to be made for the analysis as low as possible by removing inactive users from the dataset (and hence, from future API calls). Twitter offers an additional API endpoint to fetch users, which allows to check the status of 200 users per request. If a username is not returned by the API, that user account is either inactive (e.g., Twitter suspended the account due to sending spam) or deliberately deleted by the user. Using this method, inactive users are removed and the remaining set of step 2 is narrowed down to 412,228 tweets, implying that 7.2% of the tweets within the dataset were composed by users who were no longer active or banned from the Twitter platform (32,987 of 444,315 tweets) and hence, removed from the dataset.

4.4. Fetching responses

In the next step, we aim to actually fetch responses to tweets which allegedly report a hacked account (as extracted by the previously performed steps). As described in Section 2, Twitter offers a dedicated field for storing the original message if a user replies to a tweet. An obvious way to fetch responses is to use the search API to fetch messages that directly reply to tweets assuming a hack. However, general responses that address multiple users often do not use mentions at all, (e.g., *"sorry for any recently sent spam messages - my twitter account was hacked…")*. To be able to include those responses as well, we fetch all messages that were sent by the mentioned user account directly after the mentioning tweet was published on the Twitter platform.

Twitter's timeline API endpoint allows to fetch a maximum of 200 message per request [42]. To specify the time of the desired messages in a request, two fields *since_id* and *max_id* can be provided, which act as a lower and upper bound for tweet ids. Figure 4 depicts the upper and lower bounds of the timeline API. As lower bound, the mentioning tweet can be used. However, due to the tweets being sent some time in the past, the correct upper bound (i.e., the length of the timespan to be crawled) is not known in advance. The only way to ensure all relevant messages are fetched is to choose the upper bound in the present, which causes Twitter to return the latest 200 messages as a response. After receiving this batch, one can set the *max_id* to the earliest received message and continue in this manner until the timespan of interest is covered.

In addition to a 180 requests per 15-minute timeslot limitation, Twitter only allows users to fetch the latest 3,200 messages from another user's timeline, meaning that fetching possible responses that were sent too long in the past is not feasible.

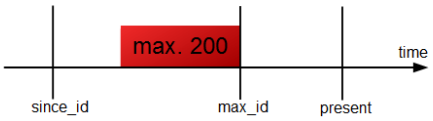

Figure 4: Fetching timelines

To prevent fetching unnecessary messages only to discover that the timespan of interest is not available, a preliminary batch can be requested, with the *max_id* parameter set to the id of the mention. The *since_id* just has to be set early enough in the past to ensure the preliminary batch to return anything, so it is set to zero. If this request returns any messages, the timespan of interest is guaranteed to be available for crawling. Figure 5 shows the schema of this preliminary batch.

When the preliminary batch does not return any results for a user, that user is ignored for all further processing as no information of the desired timespan can be gathered. Otherwise, we fetch the responses until reaching the timespan of interest and beyond.

Out of the 412,228 messages that mention users allegedly being hacked, the timelines and hence, responses of 54,835 users have been fetched from Twitter, where a response represents the overall set of messages that user sent after the mention incident. The presented method allows for fetching a total of 54,835 timelines which represent the input for the subsequent classification step.

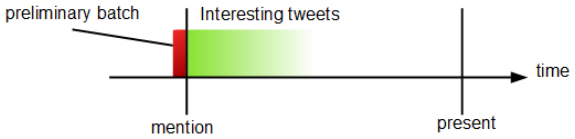

Figure 5: Preliminary batch

4.5. Classifying responses

The last step of the analysis workflow is dedicated to the actual classification and analysis of the conversations gathered. For the manual classification, a response in general is considered relevant when at least one message in the 30 messages following the mention are answering suggestions, either directly (e.g., "@howly Thanks! I will look into it!") or indirectly (e.g., "You can stop writing me, I know I was hacked"). Further analysis of the predicted data shows that 95% of the responses occurred within the first 10 messages that the accused user wrote after the mention. Hence, we argue that analyzing a total of 30 messages following the mention delivers a sufficiently large time window for the analysis.

The classification of the responses is done in the same way as in step 2, using a linear SVM and a TF-IDF vectorizer. However, the class imbalance is more substantial than in the first classification step: of 41,569 randomly chosen and manually classified messages, 617 were selected as relevant. The same methods as in step 2 were applied to cope with the imbalance. Table 6 shows the f1-scores for the tested methods. The undersampling method outperforms the others by a small margin, therefore we chose to use undersampling for the final prediction step.

In an additional experiment, we also performed evaluations regarding linguistic features of the tweets. Therefore, we removed stopwords, hashtags, mentions or URLs (and any subset of these features) from the tweet text before computing the TDF/IDF vectors. However, none of the approaches evaluated led to a substantial increase of the classification performance in regards to F1-scores of the SVM as can be seen from Table 7. Hence, we did not include any of these measures in the final classification step performed.

Table 6: F1-scores of class imbalance countermeasures (step 5)

Method	F1-score
Plain	0.77
Undersampling	0.73
Oversampling	0.78
Class weighting	0.70

Table 7: F1-scores of hashtag, mention and URL removal

Remove hashtags	Remove mentions	Remove URLs	F1-score
			0.760
		✓	0.759
	✓		0.758
✓			0.761
	✓	✓	0.753
✓		✓	0.759
✓	✓		0.757
✓	✓	✓	0.758

5. Results

In the following section, we present the results of the analyses performed by applying the methods described in Section 4 to the dataset presented in Section 3.

As for reliably reconstructing conversations on Twitter up to the point, where a user is pointed to his or her compromised account, we observe that by utilizing the presented research method, we are able to reconstruct conversations of 13.3% of all analyzed mentions. From these responses, we find 30.0% (a total of 16,452 messages) of the messages being relevant in terms of the user replying to the accusing mention. By using the text content of the messages rather than relying on direct answers, we are able to include loose conversations that are not directed to single users but address multiple possible mentioning peers (e.g. *"sorry for any recently sent spam messages – my twitter account was hacked…"*). Using Twitter-specific input sanitizing methods such as removing hashtags, mentions or URLs did not show substantial changes in regards to the classification quality (f1-score).

When it comes to the suitability of extracted conversations for detecting hacked and compromised Twitter accounts, we observe that by analyzing the peers' reactions instead of the allegedly hacked account itself, we are able to detect the distribution of spam on multiple levels, including direct messages, which are impossible to directly analyze due to Twitter's privacy restrictions. This shows that the proposed method may also be used to detect malicious behavior on other contexts than Twitter itself.

Regarding the reaction of users once they are pointed to the fact that their account might have been hacked, we find that 30% of the users that are accused of being hacked generally respond. The reactions cover a large variety of possible scenarios, including confirmations of any suspicions (e.g. *"@katiekellypoet thanks for letting me know Hun"*), but also explanations of the situation (e.g. *"haha no that was my brother"*).

We also analyzed the timespan until users who are pointed to the fact that their account might have been hacked, react to these allegations. In this analysis, we find that 80% of all users within the dataset respond within the first 24 hours. More importantly, 48% of the replies were sent within the first hour after having received the hint that their account might have been hacked. When counting the number of tweets that have been sent between the time of receiving the hint and the actual response tweet, we observe a similar behavior: 95% of all responses to hints are contained within the 10 messages sent, where 53% of all users make use of the first tweet for replying to the alleged hack of their accounts.

6. Discussion

In this section, we further discuss the findings presented in the previous section. Also, we shed light on the limitations of the proposed approach and present plans for future work following up the current study.

Firstly, we find that conversations can be reconstructed reliably. However, one limitation to the presented analyses is the fact that the base dataset was crawled 2014. Due to the fact that Twitter's API only

allows for fetching the last 3,200 messages of any given user, part of the conversations could not be reconstructed due to the amount of time passed (and hence, the number of tweets sent). In total, for 13.3% of all users which were filtered to be relevant during the preprocessing steps, we are able to reconstruct the conversations by fetching the according timelines, which certainly poses a limitation. However, we argue that using the dataset at hand, this analyses can still be regarded as a baseline for such analyses. To be able to increase the amount of users that responses can be fetched from, a future application could use real-time data of users that occur in mentions. This way, the 3,200 messages limit of Twitter is no longer a problem.

The complexity of natural languages often prevents the reconstruction of a conversation without having knowledge about its context. Therefore, a correct classification of a message is often impossible for humans too. The proposed method reaches f1-scores ranging from 0.73 to 0.78. This certainly poses a limitation to our approach. However, we argue that even in this case, our findings provide a baseline for further studies—even if more conversations might be extracted given more recent data. To increase the performance of the machine learning steps in future work, we aim to compare different machine learning methods for the given classification problem (kernel-based SVM, naïve Bayes or Random Forest).

Another limitation to the presented approach lies in the context of messages. Consider a situation in which a user apologizes to a larger audience. Even if the user was notified by a peer, there is no guarantee that this message caused that user to be aware of the hacked account as we are not able to reliably detect what actually made the user realize that his/her account has been hacked. An in-depth classification and qualitative analysis of the responses may give an insight on this situation. We plan to carry out such an evaluation in future work.

To get a deeper understanding of possibly different types of messages within the dataset, we performed an explorative study on the conversations. This analysis showed that there are different types of tweets and responses as discussed in the following.

As for the initial tweets (i.e., tweets that describe an allegedly hacked account), we observe the following types of tweets:

- Messages that plainly suggest a hack (e.g., *"@AllRiseSilver is your twitter account hacked?"*)
- Suggestions to take a specific counter action. Thus, users suggest the victims to change their password (e.g., *"@aam429 I think your account has been hacked change your password good luck"*) or to check applications which were granted permissions (e.g., *"@brijesh58 Did you DM me any link ? Or is your account hacked ? Check apps you have granted permission.")*
- Retweets of the alleged spam; possibly also containing a comment on the content of the retweeet (e.g. *" Bwahahaha! Is your account got hacked bro? RT @owlcity: j0mbl0 h4h4h4 lu k3n4 v12u5 4l4y y4? k37ul424n cy4ph4 wkwkwk - -")*
- Detailed description of the source of the spam attack in detail (e.g., *"@BeckyBeckyh123 I think your account has been hacked, just received a spurious DM from you"*).
- Messages referring to the content of the spam (e.g., *"@jessicalacie I think you've been hacked, got a dieting DM from your account."*).

As can be seen from these types of initial tweets, peers who realize that a user they follow has been allegedly hacked, include a different level of detail about the alleged hack into their tweet.

Analogous to the mentions, we also performed an explorative study on the responses to the initial tweets (as described previously) of the victims of hack attacks. We again find that there are different patterns how allegedly hacked users react:

- Some users clearly are victims of misuse and have not yet regained control over their account or possibly not even noticed that their account was hacked (e.g. *"Quickly burn off stomach fat while dropping 25lbs in a month using http://t.co/BWn4JYPYWB"*).
- Others state a direct answer to a suspicion and also react to specific measures that the mentioning user suggests (e.g. *"@Alisha_Salik thanks dude! Will do* [change password]*"*)
- More general answers often respond to possibly multiple mentioning users (e.g. *"sorry for any rogue DMs, my account got hacked"*)

As for the replies to the tweets hinting the user that his/her account might have been hacked, we observe that those answers are either directed at the initial tweet it followed or to a more general audience, when the user apologizes for the unsolicited tweets and direct messages. These findings are in line with those by Zangerle et al. [14] who generally analyzed how users react once they find that their account was compromised.

The fact that users respond quickly (80% within 24 hours, 48% within the first 60 minutes) suggests that being notified is of great importance to them.

Generally, we consider Twitter a single representative of online social networks and argue that Twitter may only serve as a showcase for such an approach which may be generalized to other online social networks as well.

7. Conclusion

In this paper, we study the behavior and reaction of Twitter users whose account has been compromised and their peer. We present a method to perform such an analysis based on a dataset of tweets about hacked accounts collected over the course of two years. Our methods allow to perform a reliable reconstruction of conversations about a hacked Twitter account. We further find that 30% of the accused victims respond to the accusations, either confirming a hack or explaining the situation. Moreover, we find that 48% of all users that actually reply to these allegations, respond within the first hour after having received the hinting tweet. Similarly, 53% of all users within our dataset make use of the first tweet to respond to the allegations after having been informed that their account might have been hacked.

Future work includes carrying out a deeper qualitative analysis of the conversations revealed by the presented extraction methods, gathering a more extensive and up-to-date dataset and further improvements of the classification methods used. Furthermore, we are interested in comparing the accuracy and the time passed until a hack can be detected to existing quantitative approaches. Another interesting topic would be to analyze the content of the tweets regarding contained topics or sentiment.

8. References

[1] Twitter, "About Twitter," 2015. [Online]. Available: https://about.twitter.com/company. [Accessed: 20-Feb-2016].

[2] C. Grier, K. Thomas, V. Paxson, and M. Zhang, "@ spam: the underground on 140 characters or less," in *Proc. of the 17th ACM conference on Computer and communications security*, 2010, pp. 27–37.

[3] K. Thomas, F. Li, C. Grier, and V. Paxson, "Consequences of Connectivity," *Proc. 2014 ACM SIGSAC Conf. Comput. Commun. Secur. - CCS '14*, pp. 489–500, 2014.

[4] J. Golbeck, *Computing with Social Trust*, 1st ed. Springer Publishing Company, Incorporated, 2010.

[5] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda, "All your Contacts are Belong to us: Automated Identity Theft Attacks on Social Networks," in *Proceedings of the 18th International Conference on WWW*, 2009, pp. 551–560.

[6] K. Thomas, D. Mccoy, C. Grier, A. Kolcz, and V. Paxson, "Trafficking Fraudulent Accounts: The Role of the Underground Market in Twitter Spam and Abuse," *USENIX Secur. Symp.*, pp. 195–210, 2013.

[7] P. Heymann, G. Koutrika, and H. Garcia-Molina, "Fighting Spam on Social Web Sites: A Survey of Approaches and Future Challenges," *Internet Comput. IEEE*, vol. 11, no. 6, pp. 36–45, 2007.

[8] K. Thomas, C. Grier, D. Song, and V. Paxson, "Suspended Accounts in Retrospect: an Analysis of Twitter Spam," in *Proc. of the 2011 ACM SIGCOMM Conference on Internet Measurement*, 2011, pp. 243–258.

[9] Twitter, "Twitter Help Center: The Twitter Rules," 2016. [Online]. Available: https://support.twitter.com/articles/18311-the-twitter-rules#. [Accessed: 28-May-2016].

[10] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, "Detecting Spammers on Twitter," in *Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference (CEAS)*, 2010, vol. 6, p. 12.

[11] K. Lee, B. D. Eoff, and J. Caverlee, "Seven Months with the Devils: A Long-Term Study of Content Polluters on Twitter," in *International AAAI Conference on Weblogs and Social Media*, 2011.

[12] A. Almaatouq, A. Alabdulkareem, M. Nouh, E. Shmueli, M. Alsaleh, V. K. Singh, A. Alarifi, A. Alfaris, and A. (Sandy) Pentland, "Twitter: Who Gets Caught? Observed Trends in Social Micro-blogging Spam," in *Proceedings of the 2014 ACM Conference on Web Science*, 2014, pp. 33–41.

[13] K. Lee, J. Caverlee, and S. Webb, "Uncovering Social Spammers: Social Honeypots + Machine Learning," in *Proceedings of the 33rd International ACM SIGIR Conference*, 2010, pp. 435–442.

[14] E. Zangerle and G. Specht, "'Sorry, I was hacked' - A Classification of Compromised Twitter Accounts," in *Proceedings of the 29th ACM Symposium on Applied Computing*, 2014, pp. 587–593.

[15] Andrew M. Jung, "Twittering Away The Right of Publicity: Personality Rights and Celebrity Impersonation on Social Networking Websites," *Chic. Kent. Law Rev.*, vol. 86, no. 1, pp. 381–417, 2013.

[16] Z. Yang, C. Wilson, X. Wang, T. Gao, B. Y. Zhao, and Y. Dai, "Uncovering Social Network Sybils in the Wild," *ACM Trans. Knowl. Discov. Data*, vol. 8, no. 1, pp. 2:1–2:29, 2014.

[17] T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer, "Social phishing," *Commun. ACM*, vol. 50, no. 10, pp. 94–100, 2007.

[18] A. Sanzgiri, A. Hughes, and S. Upadhyaya, "Analysis of malware propagation in Twitter," in *Proceedings of the IEEE Symposium on Reliable Distributed Systems*, 2013, pp. 195–204.

[19] Q. Cao, X. Yang, J. Yu, and C. Palow, "Uncovering Large Groups of Active Malicious Accounts in Online Social Networks," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014, pp. 477–488.

[20] C. Kanich, C. Kreibich, K. Levchenko, B. Enright, G. M. Voelker, V. Paxson, and S. Savage, "Spamalytics: An Empirical Analysis of Spam Marketing Conversion," in *Proceedings of the 15th ACM conference on Computer and communications security*, 2008, pp. 3–14.

[21] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna,

"COMPA: Detecting Compromised Accounts on Social Networks," in *ISOC Network and Distributed System Security Symposium (NDSS)*, 2013.
[22] M. McCord and M. Chuah, "Spam Detection on Twitter Using Traditional Classifiers," in *Proceedings of the 8th International Conference on Autonomic and Trusted Computing*, 2011, vol. 6906, pp. 175–186.
[23] J. Martinez-Romo and L. Araujo, "Detecting malicious tweets in trending topics using a statistical analysis of language," *Expert Syst. Appl.*, vol. 40, no. 8, pp. 2992–3000, 2013.
[24] S. Ghosh, B. Viswanath, F. Kooti, N. K. Sharma, G. Korlam, F. Benevenuto, N. Ganguly, and P. K. Gummadi, "Understanding and Combating Link Farming in the Twitter Social Network," in *Proceedings of the 21st International Conference on WWW*, 2012, pp. 61–70.
[25] C. Yang, R. C. Harkreader, J. Zhang, S. Shin, and G. Gu, "Analyzing Spammers' Social Networks for Fun and Profit: a Case Study of Cyber Criminal Ecosystem on Twitter," in *Proceedings of the 21st International Conference on WWW*, 2012, pp. 71–80.
[26] Twitter, "Twitter Help Center: Reporting spam on Twitter." [Online]. Available: https://support.twitter.com/articles/64986#. [Accessed: 23-May-2016].
[27] J. Song, S. Lee, and J. Kim, "Spam filtering in twitter using sender-receiver relationship," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2011, vol. 6961 LNCS, pp. 301–317.
[28] "Twitter API: POST statuses/filter." [Online]. Available: https://dev.twitter.com/streaming/reference/post/statuses/filter. [Accessed: 06-Jun-2015].
[29] "Twitter: Monitoring and Analyzing Tweets." [Online]. Available: https://blog.twitter.com/2014/connecting-to-the-pulse-of-the-planet-with-twitter-apis. [Accessed: 23-May-2016].
[30] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, 1992, pp. 144–152.
[31] G. Salton, A. Wong, and C. S. Yang, "A Vector Space Model for Automatic Indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, Nov. 1975.
[32] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A Practical Guide to Support Vector Classification," 2003.
[33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
[34] E. Leopold and J. Kindermann, "Text categorization with support vector machines. How to represent texts in input space?," *Mach. Learn.*, vol. 46, no. 1–3, pp. 423–444, 2002.
[35] T. Mullen and N. Collier, "Sentiment analysis using support vector machines with diverse information sources," *Conf. Empir. Methods Nat. Lang. Process.*, pp. 412–418, 2004.
[36] C. D. Manning and P. Raghavan, *An Introduction to Information Retrieval*, vol. 1. Cambridge university press Cambridge, 2009.
[37] V. Hoste, "Optimization Issues in Machine Learning of Coreference Resolution (PhD Thesis)," Universiteit Antwerpen. Faculteit Letteren en Wijsbegeerte., 2005.
[38] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intell. Data Anal.*, vol. 6, no. 5, pp. 429–449, 2002.
[39] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, pp. 321–357, 2002.
[40] scikit-learn, "scikit-learn: Machine Learning in Python: sklearn.svm.LinearSVC," 2016. [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC. [Accessed: 07-Jun-2016].
[41] Twitter, "Twitter Developers: API Rate Limits," 2016. [Online]. Available: https://dev.twitter.com/rest/public/rate-limiting. [Accessed: 07-Jun-2016].
[42] Twitter, "Twitter API: GET statuses/user_timeline," 2016. [Online]. Available: https://dev.twitter.com/rest/reference/get/statuses/user_timeline. [Accessed: 07-Jun-2016].