Detecting Hacked Twitter Accounts based on Behavioural Change

Meike Nauta[1], Mena Habib[2] and Maurice van Keulen[1]
[1]*University of Twente, Enschede, The Netherlands*
[2]*Maastricht University, Maastricht, The Netherlands*

Keywords: Hacked Account Detection, Social Media.

Abstract: Social media accounts are valuable for hackers for spreading phishing links, malware and spam. Furthermore, some people deliberately hack an acquaintance to damage his or her image. This paper describes a classification for detecting hacked Twitter accounts. The model is mainly based on features associated with behavioural change such as changes in language, source, URLs, retweets, frequency and time. We experiment with a Twitter data set containing tweets of more than 100 Dutch users including 37 who were hacked. The model detects 99% of the malicious tweets which proves that behavioural changes can reveal a hack and that anomaly-based features perform better than regular features. Our approach can be used by social media systems such as Twitter to automatically detect a hack of an account only a short time after the fact allowing the legitimate owner of the account to be warned or protected, preventing reputational damage and annoyance.

1 INTRODUCTION

As hundreds of millions of people use online social networks, these platforms are ideal for cybercriminals to easily reach a large audience. For years, criminals distributed spam messages using fake accounts, but nowadays social media sites have systems and tools to detect and delete spam accounts (Twitter, 2016a). As a response, criminals are hacking accounts to spread spam in the name of the legitimate owner. To this very day, Twitter accounts are an active target for hackers. In June 2016, a hacker offered millions of Twitter account credentials for sale (Whittaker, 2016).

In this study, a Twitter account is called hacked (also known as compromised) if the account is controlled by a third party and subsequently used for sending tweets without the knowledge and consent of the legitimate owner. Some Twitter accounts are hacked on purpose by an acquaintance of the owner with the goal of damaging the reputation of the legitimate owner. However, a more common cause of being hacked is that users unknowingly entrusted their username and password to a malicious third-party application or website. Furthermore, accounts with weak passwords are vulnerable for viruses or malware.

Criminals can use compromised accounts in social networks for spreading phishing links, malware, and spam. They can also collect information on specific people or commit fraud such as by asking friends of the legitimate owner to send money. Most often, hackers sell the hacked account to other cybercriminals (Demidova, 2014). A compromised Twitter account is now even more valuable on the black market than a stolen credit card (Ablon et al., 2014).

The negative consequences for the legitimate owner of the compromised account can be significant. Users have built a relationship based on authenticity and trust with their followers and this relationship is damaged after a hack. Sometimes, a hacked tweet can cause panic and is even known to be able to result in a stock market drop (Moore and Roberts, 2013). Furthermore, for both the owner and its followers, spam is simply annoying.

Problem statement: A possible solution to the problem is to construct a model that can automatically detect that an account is hacked which makes an adequate reaction possible such as quickly suspending the detected account. Early outbreak detection that contains the spread of compromise in 24 hours can spare 70% of victims (Thomas et al., 2014).

The goal of this research is to develop a model for real-time detection of compromised accounts on Twitter by detecting a change in behaviour.

Approach: We approach the detection problem by looking for a behavioural change, i.e., tweets that differ from a previously constructed behavioural profile. Such a profile contains, for example, at which time of day one usually tweets, which hashtags one

Nauta, M., Habib, M. and Keulen, M.
Detecting Hacked Twitter Accounts based on Behavioural Change.
DOI: 10.5220/0006213600190031
In *Proceedings of the 13th International Conference on Web Information Systems and Technologies (WEBIST 2017)*, pages 19-31
ISBN: 978-989-758-246-2

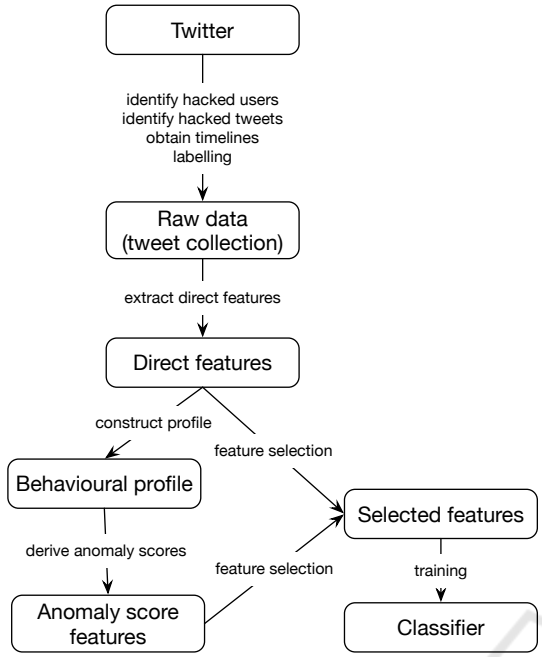

Figure 1: Overview of approach.

typically uses, etc. The behavioural profile is used to, besides *direct features* obtained from the tweet and user profile, also derive *anomaly scores*. An anomaly score is a measure for how much a feature differs from the behavioural profile. The anomaly scores are also used as features. Using these direct features and anomaly score features of all tweets of the timelines of several hacked and non-hacked users, we train a classifier that classifies tweets into classes 'hacked' and 'benign' (see Figure 1).

Deployment: The model flags a tweet as hacked when there is a significant change in behaviour. This can be used to automatically warn users about a potential hack and/or ask for confirmation of the authenticity of the tweet. A system could be put in place that would even temporarily deactivate an account and inform a user's followers when a certain amount of tweets are sent that are classified as hacked.

Obviously, a classifier typically is not perfect, i.e., it is possible that the model classifies a tweet incorrectly, hence a detected behavioural change is not malicious at all. The user could for example just have bought a new phone, is on a holiday in a foreign country, or some other non-malicious event occurred that caused the behavioural change. Therefore, it is important that the user is given the chance to authenticate the tweet, i.e., marking the tweet as benign when it is written by the user him- or herself. Such an approach is used by, for example, Google, who sends an e-mail to the user containing time and place of the login when suspicious activity is detected (Google,).

A succesful deployment should also consider the issue of bootstrapping, how to obtain a behavioural profile for new users with only few tweets. We imagine that the detection model is only activated when a certain number of tweets have occured that allow a strong enough behavioural profile. After that the behavioural profile can be updated incrementally: every tweet classified or authenticated as benign can be used to extend the profile.

Contributions: The main contributions of this paper are

- An approach for real-time detection of hacking of a social media account based on a detection of behavioural change.
- Technique for turning 'ordinary' *direct* features into *anomaly* features that correlate with behavioural change.

Outlook: Section 2 discusses related work. The subsequent sections go into more detail on the various steps: section 3 presents how the data was collected; section 4 provides details on the candidate direct features while section 5 shows how a behavioural profile can be constructed which in turn allows the derivation of anomaly score features. Feature selection is discussed in section 6. Section 7 presents the set-up and results of our experiments and section 7.5 examines the potential of the anomaly score technique by experimentally comparing it with a classifier that only uses direct features. We conclude with conclusions and future work in section 8.

2 RELATED WORK

Online social networks suffer from malicious activities. Most research focused on detecting *fake spam accounts* as opposed to tweets posted from hacked accounts from real users.

One of the first influential studies into spam on Twitter was done in 2010 by Yardi et al. (Yardi et al., 2010). They examined behavioral patterns to identify accounts that were purely created to send spam, but found only small differences on network and temporal properties compared to normal accounts. Of course, Twitter itself also reacts to the spam problem on Twitter. It is continuously optimising its solutions to combat spammers. Thomas et al. analysed accounts suspended by Twitter and found out that 77% of spam accounts identified by Twitter are deleted within one day and 92% within three days (Thomas et al., 2011). However, a handful of actors control thousands of Twitter accounts and continuously create new accounts to take place of the suspended.

Over the years, more research was done on spam

on Twitter. Gawale and Patil implemented a system to detect malicious URLs on Twitter (Gawale and Patil, 2015). Chen et al. evaluated the ability of spam detection of various machine learning algorithms (Chen et al., 2015). They found that other classifiers tend to outperform Naive Bayes and SVM on spam detection.

Hacked accounts, however, are not created for sending spam and are therefore much harder to combat. Significantly less research is done on such compromised accounts. A. Aggarwal and P. Kumaraguru landscaped the underground markets that provide Twitter followers and account credentials (Aggarwal and Kumaraguru, 2015). Zangerle and Specht classified tweets about being hacked, based on the reaction and behavior of the initial user (Zangerle and Specht, 2014). In addition, Thomas et al. studied the impact and spread of compromise by exposing the large-scale threat of criminal account hijacking and the resulting damage (Thomas et al., 2014).

The only research that has the same intention as this study is done by Egele et al. (Egele et al., 2013). They developed a tool that identifies compromised accounts on Facebook and Twitter. The tool takes six features into account. Each feature has a certain weight which is determined from a labelled training dataset. For Twitter, the weights for the features were found to be as follows: Source (3.3), Personal Interaction (1.4), Domain (0.96), Hour of Day (0.88), Language (0.58), and Topic (0.39). The tool was able to detect compromised accounts with a false positive rate of 3.6%.

Besides the results of these six features, there are other feature selection studies done using Twitter data. Mei et al. performed a hybrid feature selection method to predict user influence on Twitter (Mei et al., 2015). Benevenuto et al. (Benevenuto et al., 2010), McCord and Chuah (McCord and Chuah, 2011), and Amleshwaram et al. (Amleshwaram et al., 2013) studied Twitter spammers and the corresponding relevant features to detect these spammers.

3 DATA COLLECTION

In this section we describe how hacked users and their hacked tweets were identified. *Hacked tweets* are defined as tweets which are sent by a third-party without the knowledge of the legitimate account owner.

3.1 Identifying Hacked Users

Because Zangerle and Specht found that 50.91% of hacked users state on Twitter that they were hacked and/or apologize for any inconvenience (Zangerle and Specht, 2014), we look for hacked users by collecting accounts that state something about being hacked. However, only searching for the Dutch word 'gehackt', gives a lot of false positives like retweeted news articles and videos. Therefore, we created a list of short Dutch sentences (see Table 1) that are more precise about being hacked on Twitter. This list was tested using the advanced search method on Twitter.com and turned out to produce a manageable number of false positives while finding many true ones. The keywords on the list contain variations of a Dutch finite verb and the word 'gehackt'.

Table 1: List of short Dutch sentences used to find hacked Twitter accounts.

ben gehackt	ik was gehackt
ben gehacked	ik was gehacked
acc was gehackt	account was gehackt
acc is gehackt	account is gehackt
we zijn gehackt	gehackt geweest
Twitter gehackt	Twitter is gehackt

We searched for tweets which posted at least one of those keywords in 2014 or 2015 using the Twitter data set that was initially collected and described by E. Sang and A. van den Bosch (Sang and Van Den Bosch, 2013). We refer to this data set as "the Dutch Twitter data set" and this set contains billions of Dutch tweets starting from 2011, including the metadata; new tweets are constantly being added. The data of each tweet is in JSON-format, meaning that there is an attribute-value pair for each field (Twitter, 2016b).

Because of the size of the collection, we used the Hadoop Framework[1] and Apache Spark[2] to execute SQL queries with Spark's SQL module looking for hacked tweets. Ultimately, 18,746 tweets that stated something about being hacked were found.

Besides searching for tweets about being hacked, we also looked for hacked users by taking their screen name into account. It so happens that users who still have access to their account while hacked, change their screen name to something like 'this account is hacked' or 'please delete me' to warn their followers. Therefore, we used the search method on Twitter.com to search for users who had the Dutch word 'gehackt' in their screen name. This method resulted in a find of twenty users who got hacked between 2013 and 2016.

For all the collected users, we looked at the specific tweet that stated something about being hacked. Even when we had excluded a lot of irrelevant tweets

[1] http://hadoop.apache.org/
[2] https://spark.apache.org/

by using the keywords from the list, it turned out that the collection still contained tweets that were irrelevant. There are roughly four categories of irrelevant tweets that contain one of the keywords from the list of Table 1.

- A tweet about being hacked on another medium, like Facebook, Instagram or Steam. An example: *"Hhhmmm my instagram account is hacked, does someone know how I can contact the Instagram company? #dta #daretoask"*
- A retweet of someone else being hacked, recognizable by the 'RT' at the beginning of the text. An example: *"RT username: I'm sorry people, I was hacked and now all my followers have a link"*
- A teasing tweet about someone who could state that he is hacked to avoid having to admit that he sent an earlier tweet himself. An example: *"That police instructor is going to say that his account is hacked in a few hours. Bet?"*
- A joke or April fool that can only be recognized by the date, the context or because the joke is unveiled some tweets later. An example sent on April 1st: *"This account has been hacked by the NvvP. Let stop this defamation of Procesje!"*

The relevant versus these irrelevant tweets have only subtle linguistic differences or can only be recognized by taking the context into account. Automatically collecting the right tweets would therefore be quite complex so we did the final selection manually. We added the user to the collection of hacked users when the user himself stated that his Twitter account was/is hacked, or when a friend of the user pointed out that the user is hacked.

3.2 Identifying Hacked Tweets

With the hacked users collected, the next step is to find the hacked tweets, i.e., those sent by the hacker as opposed to the legitimate owner of the account. However, the Dutch Twitter data set used for finding hacked users barely contains hacked tweets because these hacked tweets are predominantly non-Dutch and are therefore not available in the data set. After all, this data set is collected by selecting Dutch messages and applying a language checker afterwards, as described in more detail by E. Sang and A. van den Bosch (Sang and Van Den Bosch, 2013).

Therefore, we had to turn to Twitter.com for obtaining the hacked tweets. This faces two obstacles: First, a small portion of the hacked users is already suspended by Twitter and as a result, no tweets are left on their timeline. Second, users can delete tweets which can't be recovered. This means that not all the hacked users we had in our collection, still have hacked tweets on their timeline. However, it turns out that there are still users who don't delete hacked tweets and these tweets can therefore be collected.

To partially overcome the problem of deleted hacked tweets, we also used the Twitter Streaming API to listen if the user posted at least one of the keywords of Table 1. By immediately collecting a user's timeline when one of the keywords is posted, we reduce the risk of deleted hacked tweets, because the user has no time to delete the tweets after sending the tweet about being hacked. This approach resulted in a more complete data collection.

When manually searching for hacked tweets, we specifically look at the tweets that were sent right before and after the tweet where the user (or a friend of the user) complains about being hacked. For these tweets, we pay attention to any change in language, linguistic usage and source compared to earlier tweets. For example, a Russian tweet from a Dutch user that never tweeted in Russian is highly suspicious and is probably sent by an automatic bot.

We also take the subject of the tweet into account. For example, if a user sends humiliating tweets about himself, it is likely that these tweets are not sent by the user but by someone with bad intentions. Furthermore, we check if any mentioned URL, picture or video is malicious by checking if the URL is still alive. Most malicious URLs are not live anymore and contain an alert that this link is deleted because of malicious content.

Sometimes a user even exactly describes the hacked tweets while stating that these tweets are not sent by the user, making the hacked tweets easy to find.

We could roughly recognize two kinds of hacked tweets:

- An automatically created tweet for spreading phishing links, malware and spam. An example: *"Cut 2+ inches from your tummy while dropping body mass using http://URL"*
- A tweet that is sent on purpose by someone with knowledge of the legitimate owner, often with the intent of causing reputational damage. An example: *"My name is joshua I am 11 years old I am ugly and my best friend is my teddy bear and I am gay"*

3.3 Collecting User Timelines

For each user from the hacked user collection presented in section 3.1, we look if there were hacked tweets on Twitter.com. Egele et al. empirically determined that a stream consisting of less than 10 mes-

sages does not contain enough variety to build a representative behavioural profile for the account (Egele et al., 2013). Therefore, only if there were at least 10 tweets on the timeline, we collected their tweets using the Twitter REST API "GET statuses/user_timeline". This method returns up to 3,200 of a user's most recent tweets. Each tweet is a JSON-object, with the same attributes as in the Dutch data set.

Apart from the hacked users, we also collected timelines of 67 randomly selected Dutch users. We collected their usernames using the Dutch data set (Sang and Van Den Bosch, 2013) and used the aforementioned Twitter API to collect their timelines. These users are not hacked and are added to train and evaluate our model.

This data collection method resulted in: 37 hacked users who sent 33,716 tweets in total of which 983 tweets are sent by a hacker, and 67 'normal' users who sent 53,045 tweets in total. This is the data collection used in the rest of the paper.

3.4 Labelling

Our approach is based on supervised learning, hence the data set needs to be labelled. There are two entities to label: the accounts and the tweets. The accounts were already known to be from hacked or normal users (see previous Section), so they could be easily labelled as "hacked" or "benign". The tweets were labelled as "hacked" or "benign" using the method described in section 3.2.

4 CANDIDATE FEATURES

4.1 Feature Selection Process

As illustrated in Figure 1, we experimented with two kinds of features: direct features and anomaly score features. In the sequel, we first describe our list of direct features and how we obtained them. We then describe how a behavioural profile can be constructed from the direct features of the tweets of a user's timeline. Based on the behavioural profile, anomaly score features can be derived for 'new' tweets. As this results in a quite large set of candidate features, we use a hybrid filter-wrapper method (Mei et al., 2015) to select the feature set with the best performance with the aim of making knowledge discovery easier and more efficient (Hall and Holmes, 2003).

4.2 Direct Features

We obtain candidate direct features from literature as well as from studying the domain itself.

4.2.1 Candidate Features From Literature

The only other research with the intention of detecting compromised Twitter accounts is done by Egele et al. (Egele et al., 2013). However, more research is done on fake Twitter accounts that are purely created for sending spam. Their results could be applicable to our research, because most of the hacked Twitter accounts are used for sending spam.

Egele et al. developed a tool that identifies hacked accounts on Facebook and Twitter (Egele et al., 2013). The tool takes six features into account. Each feature has a certain weight which is determined from a labelled training dataset. For Twitter, the weights for the features were found to be as follows: 'Source' (3.3), 'Personal Interaction' (1.4), 'Domain of URL' (0.96), 'Hour of Day' (0.88), 'Language' (0.58), and 'Topic' (0.39). From this result, it follows that the feature 'Topic' is the least relevant feature and almost 10 times less relevant than source. Therefore, we won't take Topic into account. Furthermore, because our aim is different and has nothing to do with the social network, we ignore network features including their feature 'Personal Interaction'.

This leaves us with the following candidate features: 'Source', 'Domain of URL', 'Hour of Day' and 'Language'.

'Language' and 'Source' also proved relevant in the research of Amleshwaram et al. to detect Twitter spammers (Amleshwaram et al., 2013). Furthermore, they take the variance in number of tweets per unit time into account. This matches with our observation that hacked accounts, particularly hacked accounts that send spam, send a lot of messages in a short time period. Therefore, we use 'Frequency' as a candidate feature to capture the behaviour of the user by counting the number of tweets per day.

'Hour of day' was also used as a feature by McCord & Chuah to detect spam on Twitter (McCord and Chuah, 2011). Their conjecture was that spammers tend to be most active during the early morning hours while regular users will tweet much less during typical sleeping hours. Furthermore, whereas Egele et al. looked at the domain of the URL, McCord & Chuah took the number of mentioned URLs into account. Therefore, we will combine these findings by using 'URLs' as a feature in which we incorporate both the domain of the URL and the number of tweets that contain a URL.

The importance of URLs is also proven by Benevenuto et al. (Benevenuto et al., 2010). They studied spammers on Twitter and took more than 60 features into account. According to them, the fraction of tweets that contain a URL is by far the best performing feature. The second best feature is the age of the account, but because we are dealing with hacked accounts instead of spam accounts and we want to detect hacked accounts real-time, it does not make sense to take this feature into account. Another good performing feature for detecting spammers according the research of Benevenuto et al. (Benevenuto et al., 2010) is 'Hashtags'. This corresponds with a result of McCord & Chuah, who proved that spammers on Twitter use much more hashtags than legitimate users (McCord and Chuah, 2011). Therefore, we will also add 'Hashtag' to the set of candidate features.

4.2.2 Candidate Features based on Domain Knowledge

Having domain knowledge is a great advantage for determining candidate features, because a better set of "ad hoc" features can be composed (Iguyon and Elisseeff, 2003). By analysing tweets that are sent by hackers and using domain knowledge, we defined a number of new features that we didn't encounter in literature but have an intuitive potential for relevance nonetheless.

First, we noticed that a hacked tweet is rarely a retweet, a repost of another Twitter user's tweet on the user's own profile to show to its own followers. In almost all cases, a retweet is sent by the legitimate owner of the account. A retweet can therefore indicate that the tweet is not anomalous, because the user trusted the tweet.

Furthermore, Twitter has a field 'possibly_sensitive', which is an indicator that the URL mentioned in the tweet may contain content or media identified as sensitive content. We can imagine that spam tweets could be classified as 'sensitive' by Twitter.

Twitter's metadata also has a field 'coordinates' which shows the longitude and latitude of the tweets location. A tweet that is sent on a location that the user has not been before, could also be malicious.

And in addition to URLs, some spam tweets also contain media elements uploaded with the tweet. Therefore we also take Media as a candidate feature.

Some other user statistics like the number of followers could also be very interesting, as a drop in the number of followers could indicate that the user sends tweets which are not appreciated by its followers. However, because the data is collected by downloading the user's timeline, we only have the user details of that current moment. That is, all tweets have the exact same data because Twitter does not give historical user data, like the development in number of followers. Therefore, we had to ignore these features and had to defer studying this class of features to future research.

Final List of Candidate Features In the end, we defined the 10 candidate direct features of Table 2. A more detailed description of some of the features is given below.

Language: The metadata of each tweet contains a language identifier corresponding to the machine-detected language by Twitter. If no language could be detected the label "und", an abbreviation of 'undefined' is assigned. The machine-detected language is, however, not always reliable. Wrongly identified tweets can be recognized by their low frequency.

As we will see in section 5.1, the language aspect of the behavioural profile is constructed by counting how often which languages are used. For example, the language profile could be {Dutch: 218, English: 87, Indonesian: 1, Swedish: 1, undefined: 12}. It is likely that the appearance of Indonesian and Swedish is wrong due to wrong classification. It is not practicable to check the language of these tweets manually, so we determined an appropriate threshold by doing testing with a sample of the data. By taking a percentage instead of a real absolute value, the threshold is relative to the total number of tweets sent. It turned out that if less than 2 percent of the total number of tweets were assigned to a certain language, it was expected that these tweets would be classified wrong. Then, for each wrongly classified tweet we changed the language identifier to "und".

Time: Each tweet has in its metadata a field with the date and time of creation of the tweet. To recognize at which times a user tweets most, we ignore the date and only take the time into account. However, to count how often a user tweets at a specific time, we apply discretization. This step consists of transforming a continuous attribute into a categorical attribute, taking only a few discrete values (Beniwal and Arora, 2012). In this case, we assign each timestamp to one of the twelve time intervals with a length of 2 hours: (00–02h, 02–04h,...,22–00h).

URLs: A user can include URLs in the text of a tweet. Each tweet of the user is categorized as either "true", meaning that the tweet contains a URL, or "false", meaning that the tweet does not contain a URL.

Apart from this binary categorization, we also extract the domains from the URLs to be used as a feature. An exception was made for the domain 'tinyurl'

Table 2: Candidate direct features.

Feature	Definition
Source	Utility used to post the tweet
URLs	Indicator whether a URL is included in the tweet. If yes, it also indicates the domain of the URL
Time	Two-hour time interval within which the tweet was created, starting from 00.00.00h–02.00.00h, . . . , 22.00.00h–00.00.00h
Language	Twitter's BCP 47 language identifier corresponding to the machine detected language of the tweet text, or "und" if no language could be detected
Hashtag	The keyword assigned to information that describes a tweet, designated by a 'hash' symbol (#).
Retweet	Indicates whether the tweet starts with 'RT', indicating that the tweet is a repost
Sensitive	Twitter's indicator if an URL mentioned in the tweet may contain sensitive content
Location	The longitude and latitude of the tweet's location, or 'false' when there is no location attached
Frequency	The number of tweets that were sent by the user on the day of the tweet
Media	Indicates whether media element(s) are uploaded with the tweet

which refers to a URL shortening service that provides short aliases for redirection of long URLs[3]. Twitter gives no possibility to check where the url points to, hence we cannot determine what the actual domain is. Therefore, we decided to simply exclude 'tinyurl' from the set of domains when it occurs.

Retweet: A retweet is a repost of another Twitter user's tweet on the user's own profile to show to its own followers, and is recognizable by 'RT' at the beginning of a text. The 'Retweet' features is binary: "true" for tweets starting with 'RT'; "false" otherwise.

5 ANOMALY SCORE FEATURES

5.1 Behavioural Profile

After the tweets are collected and labelled, we create a behavioural profile for each user based on a set of features F extracted from benign tweets from a period for which the account is sure to be not hacked. Ultimately, F is the set of features after feature selection, see section 6, but in general a behavioural profile can be constructed using any feature set F.

The profile captures the past behaviour of a user with which future tweets can be assessed in search for a behavioural change. A tweet that appears to be very different from a user's typical behaviour is expected to indicate a hack. This comparison with a profile is necessary, because, for example, an English tweet with a URL could be highly suspicious for one user but very normal for another. Therefore, direct features such as 'Language' are expected to correlate much weaker with being hacked than features based on diffences with the profile. We investigate this expectation experimentally in section 7.5.

[3]http://tinyurl.com/

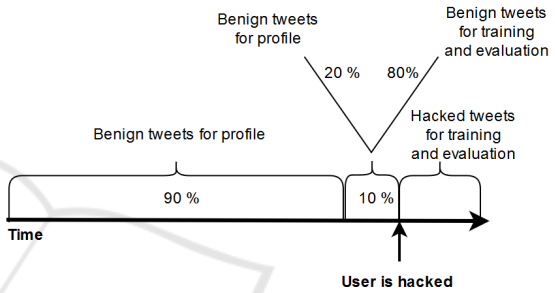

Figure 2: Division of tweets into profile construction, training, and test tweets.

Profile vs training tweets: For setting up the user's behavioural profile, we only used the tweets that were sent before the time that the first hacked tweet was sent, to ensure that no hacked tweets would taint the profile.

Furthermore, for training and testing, hacked as well as benign tweets are needed for each user. But it would be improper if the same benign tweets were used for both training as well as for construction of the behavioural profile. Therefore, we used 90% of the user's benign tweets (i.e. tweets before the hack) to set up the user's profile. The most recent 10% is excluded from the profile and can thus be used for training and testing of the model.

Ignoring the last 10% of tweets, however, means that recent behaviour is not captured in the profile. For example, if a user buys a new phone in the excluded period, then the profile will not match its behaviour anymore. Therefore, we split this 10% of most recent tweets into two parts. 20% randomly selected tweets are used for the construction of the profile as well. The remaining 80% is used for training and testing. A schematic overview of the partition of these three kinds of tweets can be found in Figure 2.

These thresholds are chosen by pursuing a balance between the number of tweets for setting up the pro-

Table 3: Example of a user's behavioural profile.

Languages	{nl: 669, und: 95, en: 78}
Times	{02-04: 1, 06-08: 72, 08-10: 82, 10-12: 133, 12-14: 74, 14-16: 86, 16-18: 85, 18-20: 146, 20-22: 121, 22-00: 42}
Sources	{http://twitter.com/download/iphone: 691, http://mobile.twitter.com: 65}
URLs	{true: 33, false: 809}
URL domains	{twitter, facebook, youtube}
Media	{true: 33, false: 809}
Retweets	{true: 94, false: 748}
Frequency	{1: 328, 2: 173, 3: 182, 4: 83, 5: 56, 8: 12, 9: 4, 12: 2}
Coordinates	{false: 804, '4.676, 52.503': 36, '4.684, 52.523': 2}
Sensitive	{true: 2, false: 840}
Hashtags	{dtv: 12, maandag: 3, yolo: 5, Fissa: 2, false: 815}

file and the number of tweets for training and testing. It is desirable to have enough tweets for making a reliable profile, while still having some benign tweets left for training. We used a sample of the data to test this approach by setting up two profiles: one with all the benign tweets before the hack and one with the benign tweets used for training left out. It turned out that the differences between these two profiles were negligible and therefore we can conclude that the thresholds are appropriate.

Setting up a behavioural profile: A behavioural profile contains historical information about the user's activities on Twitter to capture its normal behaviour. It is based on the identified candidate direct features. For each feature f, it is counted how often a value is present in the user's tweets. We define a feature tuple as a pair of the feature value v and the number of appearances c. For example, $(v,c) = (\text{English}, 78)$ is a tuple for $f = \text{language}$, representing the information that 78 tweets of the user have a language detected as 'English'. The profile contains for each feature f, the set of tuples associated with all possible values v. An example of such a behavioural profile is shown in Table 3.

5.2 Anomaly Scores

We derive an anomaly score feature for a new tweet by calculating an anomaly score between the feature value and the corresponding entry in the behavioural profile. The anomaly score is a real value in the interval $[0,1]$, where 0 denotes 'normal and not malicious' and 1 denotes 'anomalous and malicious'. For each feature f, its anomaly score feature is called 'as$f$'.

Egele et al. is the only study we found with a comparable approach (Egele et al., 2013). Since they report quite good results, we used their method for calculating the anomaly scores of the following features: 'Source', 'Language', 'URLs' and 'Retweet'. In this section, we extend the method with anomaly score functions for our other features.

The calculation method of Egele et al. is as follows:

1. If the value of that feature is not present in one of the tuples of the feature profile, then an anomaly score of 1 is returned.
2. If the value is present in a tuple of the feature, then they compare c to the mean $\overline{M}$. The mean is calculated as follows:

$$\overline{M} = \frac{\sum_{i=1}^{|f|} c_i}{|f|} \quad (1)$$

where $|f|$ is the number of tuples in f (i.e., the number of feature values) and c_i is the frequency of the i^{th} tuple in f.

   (a) If $c \geq \overline{M}$, then the value of the tweet matches the profile because the user has evidently sent a significant number of tweets with this value v. In this case, an anomaly score of 0 is returned.

   (b) If $c < \overline{M}$, then the value of the tweet is partially anomalous. A score between 0 and 1 is returned, indicating to what extent the tweet is anomalous. This score s is calculated by Egele et al. by taking 1 minus the relative frequency of the value v:

$$s = 1 - \frac{c_i}{N} \quad (2)$$

While using this calculation method, we added two small improvements:

- If a domain of a URL is present in the user's profile, the anomaly score for the URL feature is 0. Since at least one benign tweet of the user contained a URL with this domain, we consider this domain benign and therefore the tweeted URL is not anomalous.
- If the language of the new tweet is undefined, we ignore the language identifier and the anomaly score for the language feature is 0. Because if Twitter can't define which language the tweet is, we of course also cannot tell if the tweet differs from languages the user ordinarily uses.

For our other candidate features, the existing calculation method of Egele et al. (Egele et al., 2013) is not suited for calculating the anomaly scores of these features. Therefore, we defined new functions for these features.

Time: We discretized the time feature values by assigning each timestamp of a tweet to a time interval. However, by using as much as twelve intervals, Equation 2 of Egele et al. is not suitable because c_v will usually be quite low. When calculating one minus the relative frequency (Equation 2), this would then result in a high anomaly score. Therefore, we improved the function by comparing the difference

(Equation 3) between c_v and $\overline{M}$ to the mean (Equation 4). This equation is used in the case $c < \overline{M}$.

$$d = \overline{M} - c_v \tag{3}$$

$$s = \frac{d}{\overline{M} + d} \tag{4}$$

By using Equation 4 a high score is obtained if c_v differs a lot from $\overline{M}$ and a low score if there is just a small difference.

Frequency: Frequency is a feature which is not used in the research of Egele et al. (Egele et al., 2013) and the calculation of this feature is also quite different from the other features. Frequency is not just a field in Twitter's metadata, but is determined by taking other tweets of the user into account.

Equation 2 is not suited for the 'Frequency' feature, because there can be many feature values. Therefore we look at the position of the frequency of the tweet compared to all previous frequencies. Recall that the 'Frequency' profile is a set of tuples (v, c_v) for each v being the number of tweets of that day, and c_v as the number of times this frequency appeared.

We check if the frequency of the new tweet is in the lowest or in the highest half of all feature values. The half is calculated as follows:

$$h = \frac{\sum_{i=1}^{|f|} c_i}{2} \tag{5}$$

We then take the highest v that still belongs in the lowest half, which we call the 'critical point'.

- If the frequency of the new tweet is lower than the critical point (i.e., in the lowest half of all frequencies), an anomaly score of 0 is returned. The frequency is not labelled as anomalous because at least half of all frequencies is higher than the tweet's frequency.
- If the frequency of the new tweet, c_v, is higher than the critical point (i.e., in the highest half of all frequencies), we look at the position in this half to determine the anomaly score. We compare the part of the frequencies that is higher than the tweet's frequency to h, the half of all c_v.

$$s = \frac{h - \sum_{i=v+1}^{|f|} c_i}{h} \tag{6}$$

Figure 3 illustrates the case when the frequency is higher than the critical point. The anomaly score s is then calculated by $\frac{h-x}{h}$.

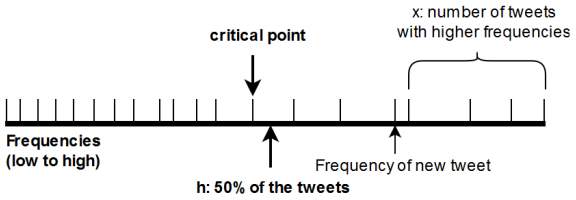

Figure 3: Example of calculating the frequency anomaly score.

6 FEATURE SELECTION

For feature relevance analysis, there are two main kinds of algorithms: *filter* methods and *wrapper* methods. A filter method directly evaluates the quality of features according to their data values and is therefore a quick way to eliminate the less relevant features. A wrapper method employs learning algorithms as the evaluation criteria to select optimal feature subsets for a high accuracy. So, unlike filter approaches, a wrapper algorithm detects the possible interactions between variables.

However, according to (Mei et al., 2015), wrapper methods often bring in a higher degree of computational complexity. Therefore, we use a hybrid filter-wrapper method. We first use a filter method to dismiss the candidate features that have absolutely no relevance. With the other features left, we use wrapper methods to evaluate their interactions. Depending on the chosen method, a particular subset of features will be brought up. The classification results of these subsets is compared to each other to eventually select the best feature subset.

We first focus on the anomaly score features. We use WEKA's Gain Ratio Attribute Evaluator to find out which features have absolutely no relevance; see Table 4. It follows that the features 'asSensitive' and 'asLocation' can be dismissed. Their irrelevance can perhaps be explained by the fact that there are just a few tweets that have sensitive content and that only a marginal subset of Twitter users have location enabled. Therefore, we dismiss these two features.

To evaluate feature combinations, we use multiple evaluation algorithms on all possible feature subsets. With 8 out of 10 candidate features left, there are $2^8 = 256$ different subsets. We used WEKA's Classifier Subset Evaluator, the Correlation-based Feature Selection Subset Evaluator (Hall, 1999), and the Wrapper Subset Evaluator.

After applying these feature selection algorithms, there are four different subsets that could be the best subset to identify hacked tweets. The features 'asTime', 'asLanguage', 'asSource', 'asURLs' and 'asFrequency' are in the resulting set of every algorithm. The question, however, is which extra feature(s) are needed for the best classification results.

To answer this question, evaluation of features and subsets of features, is done in two ways:

- How many tweets are misclassified in total?

- How many hacked tweets are misclassified?

Table 4: Ranking of candidate anomaly score features in terms of Gain Ratio.

Position	Information gain	Feature
1	0.2370	asLanguage
2	0.2197	asURLs
3	0.2098	asSource
4	0.1891	asFrequency
5	0.1847	asTime
6	0.1747	asHashtag
7	0.1277	asRetweet
8	0.0614	asMedia
9	0	asSensitive
10	0	asLocation

Table 5: Number of misclassified tweets for the best feature subsets according to several wrapper methods.

Algorithm	Feature subset	Misclassified (total)	Misclassified (hacked)
J48	asTime, asSource, asLanguage, asURLs, asFrequency, asRetweet, asHashtag	27	12
Classifier Subset Evaluator	asTime, asSource, asLanguage, asURLs, asFrequency, asRetweet	24	10
Cfs Subset Evaluator	asTime, asSource, asLanguage, asURLs, asFrequency, asMedia	22	11
Wrapper Subset Evaluator	asTime, asSource, asLanguage, asURLs, asFrequency	21	11

Table 5 presents the results. We conclude that the set {'asTime', 'asLanguage', 'asSource', 'asURLs', 'asFrequency'} performs best in lowering the total number of wrongly classified tweets. However, when 'asRetweet' is added to this feature set, one more hacked tweet is detected but three benign tweets are classified wrongly. The interesting thing is that adding the feature 'asHashtag' or 'asMedia' to the feature set results in a higher number of misclassifications, proving that feature selection is indeed an indispensable process for getting the best results.

We now examine the features by iteratively adding features in the order of the ranking of Table 4. The reduction in the number of misclassified tweets is presented in Table 6.

In conclusion, the feature selection methods do not completely agree, but since according to Table 6 'asHashtag' and 'asRetweet' appear to worsen classification and 'asMedia' doesn't make a difference, we decided to use the feature subset {asLanguage, asURLs, asSource, asFrequency, asTime} in the sequel. The coincides with what the Classifier Subset Evaluator method in Table 5 suggests.

Table 6: Number of misclassified tweets when iteratively adding features.

Features	Misclassified (total)	Misclassified (hacked)
asLanguage	351	131
asLanguage, asURLs	100	78
asLanguage, asURLs, asSource	37	20
asLanguage, asURLs, asSource, asFrequency	27	18
asLanguage, asURLs, asSource, asFrequency, asTime	21	11
asLanguage, asURLs, asSource, asFrequency, asTime, asRetweet	24	10
asLanguage, asURLs, asSource, asFrequency, asTime, asHashtag, asRetweet	27	15
asLanguage, asURLs, asSource, asFrequency, asTime, asHashtag, asRetweet, asMedia	27	15

7 EXPERIMENTS

7.1 Experimental Set-up

We compared two machine learning algorithms, which proved themselves in other studies. The first one is Sequential Minimal Optimization (SMO), which gave good results in the research of Egele et al. (Egele et al., 2013). SMO is a Support Vector Machine learning algorithm for solving quadratic programming problems and has good scaling properties (Platt, 1999).

The second considered algorithm is J48, which showed the best results in a classification experiment for spam email filtering (Youn and McLeod, 2007). J48 is an open source Java implementation of the C4.5 decision tree algorithm in the WEKA data mining tool.

As explained in section 3, our data collection contains 3,698 tweets of which 2,715 are labelled 'benign' and 983 'hacked'. The collected tweets were sent by 104 distinct Dutch users, of which 37 users were hacked for some time.

We use 10-fold cross-validation. In this validation technique the data is partitioned into 10 equally

Table 7: Overview of experimental results.

(a) Accuracy of J48 en SMO

Algorithm	Accuracy
J48	99.351%
SMO	96.106%
SMO (CSC)	96.214

(b) Confusion matrix J48

	predicted	
actual	benign	hacked
benign	2701	14
hacked	10	973

(c) Confusion matrix SMO

	predicted	
actual	benign	hacked
benign	2683	32
hacked	112	871

(d) Confusion matrix SMO (CSC)

	predicted	
actual	benign	hacked
benign	2654	61
hacked	79	904

(e) Confusion matrix J48 using direct features

	predicted	
actual	benign	hacked
benign	2684	31
hacked	38	945

sized segments and 10 iterations are performed such that within each iteration a different fold of the data is held-out for validation while the remaining 9 folds are used for learning.

7.2 Results for J48

The J48 algorithm resulted in a decision tree with a total size of 59 and 30 leaves. The complete tree can be found via https://goo.gl/IkHU2Z.

The tree first looks if 'asURLs' > 0.3985. If so, it subsequently looks at 'asLanguage'; otherwise it subsequently looks at 'asSource'. The most benign tweets, 2308 to be exactly, are detected by following this path:

asURLs $< 0.399 \rightarrow$ asSource $< 0.991 \rightarrow$ asRetweet $> 0 \rightarrow$ asLanguage < 0.979

The most hacked tweets, 563, are detected by following this path in the decision tree:

asURLs $> 0.399 \rightarrow$ asLanguage $> 0.777 \rightarrow$ asFrequency > 0.703

The decision tree produced by J48 predicts as much as 99.351% of the tweets correctly (see Table 7). The confusion matrix is shown in Table 7(b). It follows that it classified only 0.516% of the benign tweets incorrectly. Furthermore, it misclassified only 1.017% of the malicious tweets.

We manually inspected and analyzed the 10 misclassified tweets.

- 4 tweets are ambiguous, i.e., we as humans also don't know for sure that these are hacked tweets. So it could be that we manually classified these four tweets incorrectly, meaning that the algorithm recognized them correctly
- 2 tweets are Dutch spam tweets, and are the only Dutch spam tweets we found. Because the language is normal and the source was also benign, the only anomaly score which was greater than 0 was the URL score, which wasn't enough to classify the tweets as hacked.
- 2 tweets are English spam tweets with a URL. However, the user sent many benign English tweets from the same source, so the URL was the only malicious feature.
- 2 tweets are spam tweets in Russian, without a URL and with a normal source. Only a high language score is not enough to classify the tweet as hacked. Other Russian tweets in this message stream did contain a URL and these were correctly classified as hacked.

7.3 Results for SMO

The Sequential Minimal Optimization algorithm (SMO) predicted only 96.106% of the tweets correctly, which is much lower than the result of the J48 algorithm (see Table 7). The confusion matrix is shown in Table 7(c). It follows that it classified 1.179% of the benign tweets incorrectly. However, it misclassified 11.394% of the malicious tweets, which is much higher than the result of J48. Especially this increase in the number of false positives is disquieting.

This difference between SMO and J48 corresponds fairly well with the performance difference that was found by S. Youn and D. McLeod when using 3000 data instances. They compared J48 and SVM for email spam filtering and found that SVM had a success rate of 92.40% where J48 had a success rate of 97.27% (Youn and McLeod, 2007). This means that the ratio J48:SVM in their study was 1:0.950. This is comparable with our result: ratio J48:SMO is in our case 1:0.967.

To lower the number of false positives when using SMO, we used WEKA's Cost Sensitive Classifier (CSC). In this classifier, we defined the costs of the hacked tweets to be 2 times the costs of the benign tweets.

The confusion matrix after using the Cost Sensitive Classifier is shown in Table 7(d). As can be seen, this approach gives better results in terms of false positives, but is still worse than J48 (see Table 7).

7.4 Comparison with Egele et al.

As stated before, Egele et al. did a comparable study to detect hacked Twitter accounts (Egele et al., 2013). They had a data collection of 343,229 Twitter ac-

counts, of which 12,382 accounts were compromised. They used the SMO algorithm and misclassified 3.6% of the benign tweets. This is higher than our false negative ratios. Our model in combination with J48 had the lowest ratio, with only 0.516%. Furthermore, the false negative ratio of our model when used with SMO was 1.179%.

More interesting is the false positive ratio. Our model had a false positive ratio of 1.017% when used with J48 and 11.394% when used with SMO. Egele et al. doesn't explicitly give these numbers, so we calculated their false positive ratio by interpreting their results. They missed 58 of the 2,586 hacked accounts, which corresponds to 2.243%.

Based on this data, we can conclude that our model in combination with J48 has a lower false positive ratio and a lower false negative ratio. These lower ratios are probably produced by selecting more suitable features. In our model, the features 'asRetweet' and 'asFrequency' turned out to be very useful but are not used by Egele et al. However, the different results could also be explained by the different data sets used. This study focused on Dutch users, whereas Egele et al. took random users. Furthermore, the data set collected by Egele et al. contains more users.

7.5 Comparison with a Model based on Direct Features

To examine the strength of using the behavourial profile and anomaly score features instead of direct features, we also trained a classifier with J48 using only direct features.

The result of this approach is that 98.56% of the tweets is classified correctly. This is lower than the results using J48 when the tweet is compared to its behavioural profile, but still quite good. The confusion matrix is shown in Table 7(e). It follows that 3.87% of the hacked tweets (false positives) and 1.14% of the benign tweets (false negatives) is misclassified. This is much higher than the results of the J48 algorithm when used with the anomaly scores, but the false negative ratio is still lower then the ratio of Egele et al. which was 3.6% (Egele et al., 2013).

8 CONCLUSIONS AND FUTURE WORK

In this paper, an approach is presented to detect hacked Twitter accounts by comparing a new tweet to a user's behaviour profile which could help Twitter and other social media providers with detecting hacked accounts more precisely and quickly. Moreover, with the use of this approach the legitimate owner of the account and its followers can be warned that the account may be hacked, preventing annoyances and reputational damage.

The approach is based on supervised learning for which a number of hacked and normal accounts are needed with their timelines. We found hacked users by looking for tweets in which users stated that they are hacked using a large Dutch data set. We then manually searched for tweets that are sent by hackers on Twitter.com. We identify hacked tweets by looking at the topic of the tweet, changes in language and linguistic use and context. We also checked if mentioned URLs were malicious.

We collected 37 Dutch users who still got hacked tweets on their timelines. We also added 67 'normal' Dutch users to the dataset. All their tweets were labelled as 'benign' or 'hacked'.

As opposed to traditional approaches to supervised learning, we direct our features on detecting behavioural change. The 10 candidate features we defined are not used directly. We use them to construct a behavioural profile that captures a user's normal behaviour. Anomaly score features are derived by calculating anomaly scores for the features in comparison with the behavioural change.

We trained a classifier on anomaly score features of 3698 tweets using two different classification approached: J48, a decision tree-based algorithm, and SMO, an SVM-based algorithm.

Our approach performs best when used with J48: only 1.017% of the hacked tweets were missed and in total 99.351% of the tweets were classified correctly. The good results show that looking at behavioural change is a suitable approach for detecting hacked accounts. Furthermore, these results are better than the only other comparable research we found (Egele et al., 2013), which misclassified more tweets.

The technique of using anomaly score features instead of direct features has been shown to reduce the number of false positives by a factor of 3.5. Nevertheless, our feature set still achieved an accuracy of 98.56% when using direct features.

The classifier can be deployed by a social media provider in many different ways: warning users, asking for confirmation, temporarily deactivating an account, informing followers, etc.

Future Work: We expect that our model is applicable to tweets in languages other than Dutch, but future work is needed to prove our hypothesis.

Our model can be extended with other features that proved successful in other studies, such as message similarity and following rate. Yang et al. proved

that these features achieve a high detection ratio (Yang et al., 2011). Also the IP-address a tweet is sent from, can be valuable information. This information is not available to the public, but can be implemented by Twitter itself. Furthermore, the approach can be improved by checking if the URL is listed on blacklists. N.S. Gawale and N.N. Patile already implemented a system to successfully detect malicious URLs on Twitter (Gawale and Patil, 2015). Twitter also lends itself for network features, such as number of followers, user distance and mutual links. However, Twitter does not offer a way to retrieve historical data about changes in these network features, so such an extension could only be developed and evaluated by monitoring a large set of users 'hoping' that they will get hacked. Finally, features based ontext analysis may have potential, because malicious tweets, spam tweets in particular, use very striking and suspicious sentences.

REFERENCES

Ablon, L., Libicki, M. C., and Golay, A. A. (2014). *Markets for Cybercrime Tools and Stolen Data: Hackers' Bazaar*. RAND Corporation, Santa Monica, CA.

Aggarwal, A. and Kumaraguru, P. (2015). What they do in shadows: Twitter underground follower market.

Amleshwaram, A., Reddy, N., Yadav, S., Gu, G., and Yang, C. (2013). Cats: Characterizing automation of twitter spammers.

Benevenuto, F., Magno, G., Rodrigues, T., and Almeida, V. (2010). Detecting spammers on twitter.

Beniwal, S. and Arora, J. (2012). Classification and feature selection techniques in data mining. *IJERT*, 1(6).

Chen, C., Zhang, J., Chen, X., Xiang, Y., and Zhou, W. (2015). 6 million spam tweets: A large ground truth for timely twitter spam detection.

Demidova, N. (2014). Social network frauds. https://securelist.com/analysis/publications/63855/ social-network-frauds/.

Egele, M., Stringhini, G., Kruegel, C., and Vigna, G. (2013). Compa: Detecting compromised accounts on social networks.

Gawale, N. and Patil, N. (2015). Implementation of a system to detect malicious urls for twitter users.

Google. Security notification settings, alerts for new sign-ins. https://support.google.com/accounts/answer/2733203, Last visited: June 18th 2016.

Hall, M. and Holmes, G. (2003). Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering*, 15(6):1437–1447.

Hall, M. A. (1999). *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato.

Iguyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182.

McCord, M. and Chuah, M. (2011). Spam detection on twitter using traditional classifiers.

Mei, Y., Zhang, Z., Zhao, W., Yang, J., and Nugroho, R. (2015). A hybrid feature selection method for predicting user influence on twitter.

Moore, H. and Roberts, D. (2013). Ap twitter hack causes panic on wall street and sends dow plunging. http://www.theguardian.com/business/2013/apr/23/ap-tweet-hack-wall-street-freefall, 23th April 2013.

Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization. In *Advances in kernel methods*, pages 185–208. MIT Press Cambridge. ISBN 0-262-19416-3.

Sang, E. and Van Den Bosch, A. (2013). Dealing with big data: The case of twitter. *Computational Linguistics in the Netherlands Journal*, 3:121–134.

Thomas, K., Grier, C., Song, D., and Paxson, V. (2011). Suspended accounts in retrospect: An analysis of twitter spam. *Proc. of IMC 2011*, pages 243–258.

Thomas, K., Li, F., Grier, C., and Paxson, V. (2014). Consequences of connectivity: Characterizing account hijacking on twitter.

Twitter (2016a). Reporting spam on twitter. https://support.twitter.com/articles/64986, Last visited: May 9th 2016.

Twitter (2016b). Tweets field guide, developers documentation overview. https://dev.twitter.com/overview/api/tweets, Last visited: May 13th 2016.

Whittaker, Z. (2016). A hacker claims to be selling millions of twitter accounts. ZDNet, http://www.zdnet.com/article/twitter-32-million-credentials-accounts-selling-online/, June 9th 2016.

Yang, C., Harkreader, R., and Gu, G. (2011). Die free or live hard? empirical evaluation and new design for fighting evolving twitter spammers.

Yardi, S., Romero, D., Schoenebeck, G., and Boyd, D. (2010). Detecting spam in a twitter network. *First Monday*, 15(1).

Youn, S. and McLeod, D. (2007). A comparative study for email classification. In *Advances and Innovations in Systems, Computing Sciences and Software Engineering*, pages 387–391. Springer.

Zangerle, E. and Specht, G. (2014). "sorry, i was hacked" a classification of compromised twitter accounts. *Proc. of ACM SAC 2014*, pages 587–593.