**Machine Learning Methods for hacktivist classification.**

**Y. Taib, P. Pellinen, B. Han, J. Lee, M. Lion Sjin Tjoe, J. Ortiz**

The Hague University of Applied Sciences of the Netherlands

***Abstract***

*Microblogging today has become a very popular communication tool among Internet users. Millions of users share opinions on different aspects of life everyday. Because microblogging has appeared relatively recently, there are a few researches works that were devoted to this topic. In our paper, we focus on using Twitter, the most popular microblogging platform, for the task of classify hacktivists. We show our mining process, how we collect the data from Twitter, we build a Naive Bayes classifier that is able to determinate if the tweet is an hacktivist tweet or if is not. The experimental evaluations shows our results applying the machine learning methods. In our research, we worked with English, however, the proposed technique can be used with any other language.*

***Keyword***

*Hacktivism, Bayesian, words filtering, Machine Learning algorithms*

**Introduction**

Social networks have been a powerful platform to achieve globalization in recent years, millions of data are generated every day, it is easy to learn about the things that are happening in the world. It is also easy to share information about food, traditions, the culture of other countries, etc. Therefore, you can learn about anything from social networks. Twitter[[1]](#footnote-2) is one of the most popular forms of social networks[1]. Users can tweet their thoughts, retweet others, favourite other tweets and use hashtags in order to find tweets that are similar to their tweets. This helps certain hashtags to go viral.

In this huge social network, there is a problem called ‘hacktivism’. Hacktivism is the use of technology to promote political agenda or social change. The reason for hacktivism is to draw public attention. The hacktivists believe that the issue is important, they address issues like freedom of information or human rights. Hacktivists also use it to oppose something in which case they use images or messages to websites of organizations they believe something wrong[2]. Hacktivism can have several meanings. In some cases, it is about breaking into the security barriers of a computer as well as it can mean hacktivists gain unlawful access into a network.

‘Cyber Security’ is a project group that aims to create two working classifiers for twitter. The main subject of this project is identifying a hacktivist based on language used in tweets. This is achieved by creating a keywords list, this keywords list is in turn used by two classifiers that use machine learning in order to identify the number of times a word has been used. With two classifiers a greater accuracy can be achieved in distinguishing a hacktivist from a non-hacktivist. Machine learning will be used to show patterns and make predictions on the twitter data.

This paper focuses on Naive Bayes and SMV by using data which is extracted from tweets in order to distinguish a hacktivist from a non-hacktivist. These two classifiers can reach the frequency of some words that there are identified as hacktivism words because we labelled one of the datasets that we are using to be sure that there are some tweets that in fact encourage to the hacktivism, and how do we know that? Because we read more than 9,000 tweets and classified in hacktivist and non-hacktivist tweets, that can give us a ground truth of some context that in this case, a computer can´t give us, because we are able to understand the context of some of the tweets.

The rest of the paper is organized as follows. Section 3 provides terminology that is used in the project. Section 4 illustrates the approach to achieve the goal. Sections 5-6 detail the data gathering, labelling, the classifiers used in this project and the results. Section 7 concludes this project.

**2. Main question and sub questions of the research**

**2.1 Main question**

Can we distinguish hacktivists from normal users by language?

**2.2 Sub questions**  
What kind of behaviour do hacktivists display?

Which of the hacktivists are active, and which are passive?

What are the terms active hacktivists use and what are the terms passive hacktivists use?

What are the trends by hacktivists?

**3. Terminology**

In this section, hacktivism, Tweepy will be explained. Hacktivism is an online form of activism. Used by hackers to target people as well as organisations. Hacktivism targets people and organisations in order to make a change in the world they live in by denying service of an app or hacking into for example a Twitter page of a dangerous group, to send out a message that what they’re doing is unacceptable. This can be done on a variety of platforms ranging from a twitter page (Which is the subject of this research plan) to banks, government websites [3].

**3.1 Tweepy**

Tweepy gives access to the Twitter API, using Tweepy is it possible to attain any object. Using Twitter developer page, it is possible to access tweets information. The main purpose of using Twitter is monitoring tweets an important part of Tweepy is a stream listener object that monitors tweets in real time.  
Tweepy is the open-source library that provides access to the Twitter API for python. It relies on Twitter API and that has excellent documentation. Which means it can very much be used as a reliable library. Other libraries like python-twitter provide many functions, but Tweepy has a large community and the most code that’s added [4].

**3.2 Anonymous**

The most famous group of hacktivists is Anonymous. Anonymous is a group of hackers that create missions and tasks in order to make a change in the world they live in. This includes taking down paedophiles and sites that host child pornography, and this along with other missions that serve to better society.

They have amassed an enormous following on social media. Anonymous is also highly active on social media including Twitter. Besides Anonymous having their own social media accounts, it’s members also have accounts. Using their account to not only express views and alert about upcoming events, but they also have accounts that they use in their daily lives [5].

1. **Approach**

Our main goal is to distinguish hackers from normal users in Twitter from the language they use. To achieve that, we need to complete a few steps; Extracting the Twitter data, processing the data to a more accessible form, build two classifiers for accuracy and lastly train and test the classifiers.

Our approach was to start from mining the data from Twitter. It is the first step because, without the data, we cannot do anything else. Data mining was done with an API from Twitter called Tweepy, which was modified to fill our purposes better. To get the data extraction working we had to apply and receive a developer account from Twitter. After that, we could start to extract and store tweets in a text file.

Text pre-processing was used to clean the data and pass on only relevant information. In text pre-processing, there are a few different steps of manipulating the data. The goal of pre-processing was to make all the tweets to be in the same format for easier use later on. In this step, all the tweets are first in their original form. Then a few functions are applied to the data frame of tweets to make them unified. Punctuation was removed first because it makes further applications of functions easier. While removing punctuation, also emoticons are removed. After that, all the tweets were fully converted to be only lower-case characters. In addition, English stop words were removed. Natural language toolkit list of Stop words include words like “a, an, the, in, on” and so on.

Now we have tweets that are stripped from unnecessary words, there is no punctuation, and everything is in lower case for easier manipulation. Text pre-processing is vital for the next step of counting the frequency of words used in the tweets of followed users.

Terms frequency counting was used for comparing the most used words for a keywords list provided to us by the product owner. In addition, new keywords were added to the existing list from the most frequent words used by hacktivists. On addition of counting the frequent words, different counter functions were used to count only hashtags or usernames used in the tweets.

The next step is analysing the data. There was some manual work involved in the analyzation process in labelling by hand a part of tweets by being related to hacking or not. The tweets labelled were from users that we know are hacktivists. This was a tool used in our classifier to determine the users’ involvement in the hacktivist scene.

Data was then visualized to get a better understanding of the results. Visualization helps for example in a situation when one needs to access word frequency of specific dates or the popularity of specific words on different dates.

With visualization, it is also easier to spot false positives and negatives from Naive Bayes classifier. Naive Bayes classifier was chosen because it is optimal for text categorization and offers a solution to the problems of the project. Supervised learning with previously mentioned manually labelled tweets are used to teach the classifier. Lastly, sentiment analysis is used for classifying the tweets in three categories; positive, neutral and negative tweets. With sentiment analysis it is possible to detect a change in the emotion of the tweets, possibly alerting from a future defacement of a website and more.

**4.1 Related Work**

In this section, relevant work will be discussed. Similar research and the result will be displayed in this section. For this research, a study concerning social sentiment and a study concerning Twitter will be used for reference.

Cyber-attacks are on the rise because of the increasing globalisation. These attacks form a great risk for the following areas; Denial of service, data leaking and application compromising among others.  
A variety of anti-threat measures are in effect in order to combat attacks like DDoS[[2]](#footnote-3). Predictive analysis can be particularly beneficiary for Twitter because of the fact that certain Twitter functions that include retweets, favourites and replies can be characterized and this along with the polarity of the text can improve predicting events like political elections and the release of new products. The predictive power of social networks can be used by investigating published data and statistical modelling and that can assist in identifying statistical similarities between social users on social media. Therefore, it can be concluded that sentiment analysis is a useful way to analyse tweets.

In research [6] Twitter has been researched using Sentiment Analysis, using Natural Language Processing and Machine Learning techniques to interpret sentimental tendencies related to user opinions and make predictions about user opinions and make predictions about real events.

In the paper, a Social Sentiment Sensor in Twitter has been used in order to collect historical tweets. This in order to classify negative, positive and security-oriented tweets. When three different classification algorithms were used to evaluate results, maximum entropy provided the most effective results. Naive Bayes and Support vector machine followed and were responsible for less accurate results. In this research, classification results of 80% can be achieved.

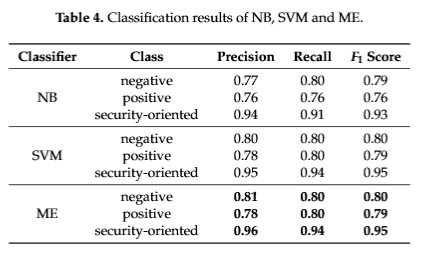


Table 1 Classification results of NB, SVM and ME.

Concluding that sentimental analysis for Twitter can be used as a reliable base to analyse tweets.   
Research [7] looks into movie reviews in order to attain the sentiment of the reviewers. Where it starts by stating that research towards this topic has come from a shift of interest from the topics of discussion. This can be for example sports or politics. The researchers have concluded that this shift has changed into the interest for the sentiment of the users rather than just the discussion topics. This can be about a product review, focusing on whether the review would be positive or negative. On Rotten Tomatoes (a movie critic website) the reviews are labelled with a rating system in addition to the review.

The similarities between this related research and the research that is being presented right now becomes apparent, while it’s possible to extract the keywords (because keywords hold a clear value because its only one word) with sentiment comes a variety of possibilities to express emotions. For example; Without a rating, a sentence like: Well this was great, Contains zero words with a negative value. However, when you also take in account a rating system 1-10 and it has been graded by the reviewer with a low score, it becomes apparent that the reviewer used sarcasm in order to get their point across.

Three machine learning methods were used in order to see what classification method Naive Bayes classification, maximum entropy classification, and Support Vector Machine. Two of which were also used in this research, Naive Bayes and Support Vector Machine will be explained in the next section. The classification accuracy for Naive Bayes combined with Support Vector Machine topped 80%. Where the accuracy lacked was in the sentences that had an abundance of positive words. For an example a sentence where the effort put into performance was graded in the review rather than the performance itself.

In the relevant works that have been examined, there has been a clear pattern is revealed.   
Also, it confirms that it is difficult to differentiate the positive or negative sentiment for either a tweet or a movie review. Researcher confirm that using Support Vector Machine and Naive Bayes is completely legitimate and reliable ways of using machine learning in order to attain the classifier to recognize hacktivist or no hacktivist.

1. **Methodology**

Machine learning field is a subfield from the broad field of artificial intelligence, this aims to make machines able to learn like human. Learning here means understood, observe and represent information about some statistical phenomenon[8]. In our tweets classification, the features could be the words in the line and the classification of the line, hacktivist or non-hacktivist. So the input to hacktivist classification task can be viewed as a two dimensional matrix, whose axes are the tweets and the label. We divided the hacktivist classification into several tasks. First step is data collection and representation, second step, data preparation, we cleaned the characters that are not relevant for us (@,#,Rt, http//…). Finally, the hacktivist classification phase of the process finds the actual mapping between training set and testing set. In the following section we will review our machine learning methods.

**5.1 Naïve Bayes classifier method**

In 1998 the Naïve Bayes classifier was proposed for spam recognition. Bayesian classifier is working on the dependent events and the probability of an event occurring in the future that can be detected from the previous occurring of the same event [9]. This technique can be used to classify hacktivist or no-hacktivist but al so spam e-mails; words probabilities play the main rule here. If some words occur often in spam but not in ham, then this incoming e-mail is probably spam. Naïve bayes classifier technique has become a very popular method in mail filtering software. Bayesian filter should be trained to work effectively. Every word has certain probability of occurring in spam or ham email in its database. If the total of words probabilities exceeds a certain limit, the filter will mark the e-mail to either category. Here, only two categories are necessary: spam or ham. Almost all the statistic-based spam filters use Bayesian probability calculation to combine individual token's statistics to an overall score [9] and make filtering decision based on the score. The statistic we are mostly interested for a token T is its spamminess (spam rating) [10], calculated as follows:

S[T] =

Where **C**Spam(**T**) and **C**Ham(**T**) are the number of spam or ham messages containing token T, respectively. To calculate the possibility for a message M with tokens {T1,......,TN}, one needs to combine the individual token's spamminess to evaluate the overall message spamminess. A simple way to make classifications is to calculate the product of individual token's spamminess and compare it with the product of individual token’s hamminess

(H[M] = )

The message is considered spam if the overall spamminess product **S[M]** is larger than the hamminess product **H[M].** The above description is used in the following algorithm [3]:

**Stage1. Training**

Parse each email into its constituent tokens Generate a probability for each token W

S[W] =(W) / ((W) +(W))

store spamminess values to a database

**Stage2. Filtering**

For each message M while (M not end) do scan message for the next token query the database for spamminess S() calculate accumulated message probabilities S[M] and H[M] Calculate the overall message filtering indication by I[M] = f(S[M] , H[M]) f is a filter dependent function,

such as I [M] =

if I[M] > threshold

msg is marked as spam

else

msg is marked as non-spam

In the project we use the same method as explained above. We change object spam to hacktivist and ham as no-hacktivist to get the accurate prediction.

**5.2 Support Vector Machine classifier method**

Support Vector Machines are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships, the SVM modelling algorithm finds an optimal hyperplane with the maximal margin to separate two classes, which requires solving the following optimization problem.

Maximize

**½**

Subject to

**= 0**

where 0 ≤ αi ≤ b, i = 1, 2,….n

Where αi is the weight of training sample x1. If αi > 0, x1 is called a support vector b is a regulation parameter used to trade-off the training accuracy and the model complexity so that a superior generalization capability can be achieved. K is a kernel function, which is used to measure the similarity between two samples. A popular radial basis function (RBF) kernel functions.

K() = exp(-y||) , y

After the weights are determined [11], a test sample x is classified by

y = Sign (,

Sign (a) = +1, if a > 0

-1, otherwise

To determine the values of < γ, b >, a cross validation process is usually conducted on the training dataset [12]. Cross validation is also used to estimate the generalization capability on new samples that are not in the training dataset. A k-fold cross validation randomly splits the training dataset into k approximately equal-sized subsets, leaves out one subset, builds a classifier on the remaining samples, and then evaluates classification performance on the unused subset. This process is repeated k times for each subset to obtain the cross-validation performance over the whole training dataset. If the training dataset is large, a small subset can be used for cross validation to decrease computing costs. The following algorithm [13] can be used in the classification process.

1. **Results**

In order to test the performance of above mentioned methods, we have several collections of tweets that we get it using Tweepy[[3]](#footnote-4) API, labeled by ourselves. After that we have divided the data into training and testing sets keeping, in each such set, the same proportions of non-hacktivist and hacktivist messages as in the original example set.

**6.1 Collecting data**

Using a development account that can be attained via Twitter’s corporation you will be able to create applications, then using Tweepy API we have access to collection methods, we collected 9000 Tweets from hacktivist users, the fields that we collected were: Tweets, Name of the user, Length of the Tweet, ID of the user, Date of the Tweet, Source(From where they published the Tweet), Likes and RTs[[4]](#footnote-5). One of the problems that we faced was that some users had Tweets in English but also in other languages. In Figure 1 we show an example of the first collection of the data that we had, before filtering.

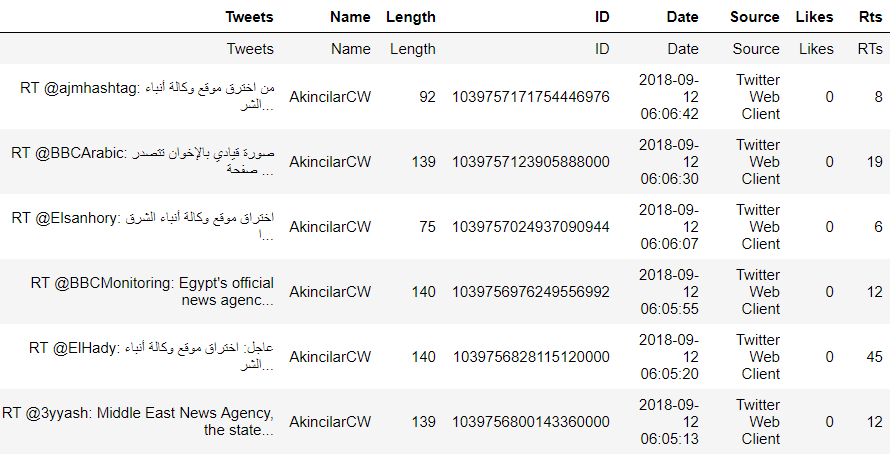


Figure 1. Uncleaned dataset.

**6.2. Data Filtering.**

After that we collected the data, we split the data and then focused on the English Tweets. So we split the data in order to label the tweets that contained words related to hacktivism, we labeled with a “1” if the tweet had some context or words that encouraged hacktivism, and with “0” those who we considered neutral tweets even if they were posted from hacktivist users. After that, we delete all the columns that were irrelevant for us and cleaned the tweets in this way:

Usernames:

Users often include Twitter usernames in their tweets in order to direct their messages. A de facto standard is to include the @ symbol before the username (e.g. @alecmgo). An equivalence class token (USERNAME) replaces all words that start with the @ symbol.

Usage of links:

Users very often include links in their tweets. An equivalence class is used for all URLs. That is, we convert a URL like “http://tinyurl.com/cvvg9a” to the token “URL.”

Retweets:

Twitter special words (such as “RT”), and emoticons.

In the Table 2 below we put an example of a tweet that was filtered and cleaning against the original version of that Tweet.

|  |  |
| --- | --- |
| Original Version of the Tweet | Cleaned Tweet |
| RT @3yyash: Middle East News Agency, the state news agency in Egypt, was hacked by who seems to be a Turkish hacker. The hacker posted the… | middle east news agency the state news agency in egypt was hacked by who seems to be a turkish hacker the hacker posted the |

Table 2. Uncleaned tweet vs cleaned tweet.

**6.3 Data process**

After doing that we took away the English common words from our tweets, some articles such as “the”, “in”, “a” and so on. We did this in order to have a better accuracy between the specific hacktivist words and the simple words in tweets. In Table 3 we show an example of the cleaned tweet with the English articles and the tweet that doesn’t have those articles anymore.

|  |  |
| --- | --- |
| Cleaned Tweet | Stopped English words´s Tweet |
| middle east news agency the state news agency in egypt was hacked by who seems to be a turkish hacker the hacker posted the | ['middle', 'east', 'news', 'agency', 'state', 'news', 'agency', 'egypt', 'hacked', 'seems', 'turkish', 'hacker', 'hacker', 'posted'] |

Table 3. In this table we show the example of a tweet in which had been removed the common english words.

After doing that we put the words into a matrix using CountVectorizer[[5]](#footnote-6) method in order to give the input to the Naive Bayes classifier. After that we split our Data in Training set 6839 tweets and Test Data 3369 tweets. We search for the best value of alpha. We made a table in order to compare which is the most accurate value of alpha in the Test set. We consider it the following fields: alpha, Train Accuracy, Test Accuracy, Test Recall and Test Precision. So we were able to compare the results and see which value gave us the best results.

We did the same with the Supported Vectors Machine classifier, we split the data and start making the comparation but this time with a C value instead the alpha and the same fields Train Accuracy, Test Accuracy, Test Recall and Test precision.

Finally, we conclude that the best classifier for us was Naive Bayes classifier if we compare it with SVM classifier, because we got a Test Accuray of 0.9121 in Naive Bayes classifier while we got 0.8993 Test Accuracy in SVM classifier.

**7. Conclusions**

This paper presented a methodology to distinguish hacktivists by using our own classifiers and sentiment analysis in Twitter. Our methodology collects tweets from 70 hacktivists and classifies them as negative, positive by hand-written sentiment classifier. Specifically, we have shown that the proposed methodology can detect whether the user is hacktivist. We notice that Machine learning algorithms can do high accuracy for distinguishing users when using our methodology.

Social media can help people around the world communicate freely. we can use the social media as useful analysis media to detect the potential threats. In this paper, it has been shown that it is efficient to analyse individual tendencies to detect possible hacktivist users. To do this, sentiment analysis, and word count frequency was conducted on the collected tweets, and we classified the users by the level of threat according to our criteria. And then, the classified possible hacktivist users were verified by performing information security compliance matching process. Machine learning algorithms were applied to detect possible hacktivist users. In this way, a methodology has been proposed to prevent damage to the organization’s information systems. This paper contributes to the analysis of data on social media to show that the criteria for detecting hacktivism threats are based on the sentiment level, word count frequency and the ratio of negative emotions, and it can be verified based on the concept of information security compliance. Above all, to improve the level of information protection of the organizations, it is necessary that not only the information protection person but also the management’s active interest and efforts are put together.

Our work is actually not limited to detect hacktivists. Our future goal is making complete machine learning algorithm for predicting cyberattacks from hacktivists.

**7.1 Future Work**

In this work we have explained a robust methodology for the study whether the addition of Twitter data helps in forecasting. Our work could be improved in many ways, we proceed by explaining a few of the ideas we have not implemented yet. For example, we are currently using libraries to recognize sentiment in the sentiment analysis. In order to train a sentiment classifier, supervised learning usually re-quires hand-labelled training data. With the large range of topics discussed on Twitter, it would be very difficult to manually collect enough data to train a sentiment classifier for tweets. Because uses of context is very difficult to explain through the vocabulary that people use online. like shortening words, emoticons, words with numbers and words with punctuation marks. Also analysing URL from a tweet was very difficult to accomplish. Machine learning techniques perform well for the classifier and classifying sentiment in tweets. We believe that the accuracy could still be improved.

**References**

1. Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford, 1(12).
2. Rahman, A. (2018, 11 april). Hacking and Hacktivism: What it is and How Can it Affect Us. Retrieved from <https://medium.com/@rahman.alif1/hacking-and-hacktivism-what-it-is-and-how-can-it-affect-us-611e3e341967>
3. Rahman, A. (2018, 11 april). Hacking and Hacktivism: What it is and How Can it Affect Us. Retrieved from <https://medium.com/@rahman.alif1/hacking-and-hacktivism-what-it-is-and-how-can-it-affect-us-611e3e341967>
4. Novalić, A. (2013, August 13). Introduction to tweepy, Twitter for Python. Retrieved from <https://www.pythoncentral.io/introduction-to-tweepy-twitter-for-python/>
5. Rahman, A. (2018, 11 april). Hacking and Hacktivism: What it is and How Can it Affect Us. Retrieved from <https://medium.com/@rahman.alif1/hacking-and-hacktivism-what-it-is-and-how-can-it-affect-us-611e3e341967>
6. Hernandez-Suarez, A., Sanchez-Perez, G., Toscano-Medina, K., Martinez-Hernandez, V., Perez-Meana, H., Olivares-Mercado, J., & Sanchez, V. (2018). Social Sentiment Sensor in Twitter for Predicting Cyber-Attacks Using ℓ₁ Regularization. Sensors (Basel, Switzerland), 18(5), 1380. doi:10.3390/s18051380
7. Thumbs up? Sentiment Classification using Machine Learning Techniques. Retrieved from <http://www.cs.cornell.edu/home/llee/papers/sentiment.pdf>
8. W.A. Awad, S.M. ELseuof. “Machine Learning Methods For Spam E-mail Classification” International Journal of Computer Science & Information Technology (IJCSIT), Vol 3, No 1, Feb 2011
9. Almeida,tiago. Almeida, Jurandy.Yamakami, Akebo " Spam filtering: how the dimensionality reduction affects the accuracy of Naive Bayes classifiers" Journal of Internet Services and Applications, Springer London , February 2011
10. M. N. Marsono, M. W. El-Kharashi, and F. Gebali, “Binary LNS-based naïve Bayes inference engine for spam control: Noise analysis and FPGA synthesis”, IET Computers & Digital Techniques, 2008
11. Li, K. and Zhong, Z., “Fast statistical spam filter by approximate classifications”, In Proceedings of the Joint international Conference on Measurement and Modeling of Computer Systems. Saint Malo, France, 2006
12. El-Sayed M. El-Alfy, Radwan E. Abdel-Aal "Using GMDH-based networks for improved spam detection and email feature analysis"Applied Soft Computing, Volume 11, Issue 1, January 2011
13. Yuchun Tang, Sven Krasser, Yuanchen He, Weilai Yang, Dmitri Alperovitch ”Support Vector Machines and Random Forests Modeling for Spam Senders Behavior Analysis” IEEE GLOBECOM, 2008
14. Hao Zhang, Alexander C. Berg, Michael Maire, and Jitendra Malic. "SVM-KNN: Discriminative nearest neighbour classification for visual category recognition", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006
15. Munkhdorj, B., & Sekiya, Y. (2017). Cyber attack prediction using social data analysis. J. High Speed Networks, 23, 109-135.
16. Park, W., You, Y., & Lee, K. (2018). Detecting Potential Insider Threat: Analyzing Insiders' Sentiment Exposed in Social Media. J. Security and Communication Networks, 2018, 7243296:1-7243296:8.
17. Arias, M., Arratia, A., & Xuriguera, R. (2013). Forecasting with twitter data. ACM Transactions on Intelligent Systems and Technology, 5, 8:1-8:24.
18. Bahia, J. (2018). The online battleground : the use of online platforms by extremist groups and hacktivists to form networks and collective identities. T. Doctoral dissertation, University of British Columbia. Retrieved from <https://open.library.ubc.ca/collections/ubctheses/24/items/1.0369285>

1. <https://twitter.com/> [↑](#footnote-ref-2)
2. A distributed denial-of-service (DDoS) [↑](#footnote-ref-3)
3. http://www.tweepy.org/ [↑](#footnote-ref-4)
4. An abbreviation for retweet, which means citation or reposting of a message [↑](#footnote-ref-5)
5. <https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html> [↑](#footnote-ref-6)