2017

Predicting Software Vulnerability Using Security Discussion in Social Media

Andrei Queiroz
d15127570@mytudublin.ie

Brian Keegan
Technological University Dublin, brian.x.keegan@tudublin.ie

Fredrick Mtenzi
Technological University Dublin, Fredrick.Mtenzi@tudublin.ie

Recommended Citation

Queiroz, A., Keegan, B. & Mtenzi, F. (2017). Predicting software vulnerability using security discussion in social media", *European Conference on Information Warfare and Security, ECCWS*, 2017, Dublin, Ireland. doi:10.21427/zgtj-nx67

Funder: CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico

Predicting Software Vulnerability Using Security Discussion in Social Media

Andrei Queiroz, Brian Keegan and Fredrick Mtenzi
Dublin Institute of Technology, Ireland
andrei.queiroz@mydit.ie
brian.x.keegan@dit.ie
fredrick.Mtenzi@dit.ie

Abstract: Social media has been used as a tool for the dissemination and exchange of information among people in many different areas of knowledge. Computer security is one which utilises social media in this way. Researchers and specialists in security are using social media tools for informing their discoveries on subjects as computer security, software vulnerabilities, exploits, data breach and hacker intrusion. Within the context of social media, Twitter might be the first channel used by security researchers for disclosing novelty (such as exploits or backdoors) in computer security. This paper proposes a Support Vector Machine (SVM) classification model using Twitter posts (tweets) as a source for filtering relevant information related to software vulnerabilities. In this paper, tweets considered relevant will be those alerting about new vulnerabilities in software (being exploited or not), as well as posts alerting software users about security patches and updates. The non-relevant information will be considered as those which have no warning characteristic, i.e.: tweets about opinion, general conversation and topics which have no sense of alert. The proposed model achieved an accuracy of 94% by using simple features such as the frequency of words (unigram and bigram). Reasonable rates of recall and precision into the desirable class values were recorded as, 68% and 46% respectively for the same simple features. This experiment opens a path for future studies about the relationship between how alerts and discoveries in computer security are expressed by the security community on social media posts.

Keywords: social media, Twitter, machine learning, support vector machine, software vulnerability, cybersecurity

1. Introduction

Twitter is commonly used as a tool for information exchange by people from different fields of knowledge. As its information can be publicly accessed and is a long reaching, some researchers focus on studying the trustworthiness of Twitter posts and how this data can be used as a non-official source of information as in Sankaranarayanan et al (2009) and Cataldi, Di Caro, and Schifanella (2010). An example of information being used for alerting is found in Sakaki, Okazaki and Matsuo (2010), which has shown that, in Japan, Twitter posts about earthquakes could warn people faster than the official seismic agencies in that country.

Likewise, in the security field, researchers and experts in computer security are using Twitter for transmitting its discoveries. Professionals observed that their posts, in general, are a mix of issues that can range from computer security discussions to personal opinion or informal conversation in several subjects.

Twitter is a very important tool for these professionals as they can use it for self-promotion and demonstration of their skill as security specialist. As a result, their tweets can warn people and software vendors about news regarding vulnerabilities, hacker attacks, data breaches and software patching/updating.

In terms of software vulnerability, the experiment in Trabelsi et al (2015) found evidences about researchers disclosing information about new vulnerabilities on Twitter before a NIST Vulnerability Database (NVD), which is a well-known source sponsored by US government that aims to publicly disclose software vulnerability information. However, security experts are not using their accounts just to talk about security issues, they generally use them to post some personal information which is not considered as an alert or warning. For instance, tweets relating to a book promotion, conferences or software products, their posters political opinion, and sometimes tweets regarding their personal preferences and daily events. With this context in mind, this paper is proposing a machine learning model to classify useful information coming from security expert posts on Twitter social media. In this article, a machine learning model was used for classification of useful information and alerts about software vulnerabilities that can be valuable for computer users and vendors.

The paper is organised into the following sections: Section 2 will discuss the related work with a focus on works that use social media as source of information for computer security and what their main findings are. Section 3 will discuss how the dataset was gathered and the relevance of the chosen security experts. In addition, it will discuss how the dataset was labelled. Section 4 will discuss how the tweets were pre-processed to produce the key features and an explanation of the algorithm chosen for modelling the predictor. In section 5, the results will be presented along with supporting discussion. Section 6 provides a discussion about the challenges of this classification task. In section 7, the conclusion is presented in addition to future work.

2. Related work

In this section, we provide a discussion about papers related to security and the use of social media as a source of information for cybersecurity.

In Nunes et al (2016), the authors have focused on classifying information from Deep Web forums. The work tries to separate information about software security from information related to criminal aspects like trading of drugs and credit card numbers.

In Trabelsi et al (2015), the authors claim that software vulnerability information is scattered around several sources. For example, vendors, NVD and others non-structured sources. In order to deal with such scattering, they build a framework to monitor Twitter posts about software vulnerability. They claim that 0-day vulnerability in Linux kernel was released in Twitter before an official source.

In Mittal et al (2016), the authors describe Twitter as OSINT (Open Source Intelligence) that is capable of providing real time information about security threats and vulnerabilities. They claim that their framework, based on semantic web RDF and SWRL rules, could inform about current events in cyber security which, provides the possibility of reacting against these threats.

In Benjamin et al (2015), the authors carried out research in different media in order to find evidence of emerging cyber threats. Their chosen source was IRC, which is a protocol of conversation on the Internet, commonly used in hacker forums and carding shops.

In Sabottke, Suciu and Dumitras (2015), the authors conducted a quantitative and qualitative analysis of information in Twitter in order to present the existence of exploits being used for attackers. Additionally, they provide a system to detect real exploits by using Twitter post and metadata information.

In Lippmann et al (2016), the authors provide a model to detect cyber discussion in social media. They used Twitter, Reddit and Stack Exchange as source of information. They presented a classifier that was able to identify whether the content discussed in those social media were related to cyber threats or not.

In Dunphy et al (2015), the authors carried out research about security experiences from users in social media. They retrieved all posts with the hashtag "Password" and noticed that people are putting their security at risk when disclose information about their usage habits regarding passwords and authentication system in websites.

3. Dataset

In this section, we discuss how the dataset was formed and how the samples were selected. Additionally, we are going to discuss how the data was labelled and which class will be the focus of this work.

3.1 Gathering data

In this experiment, the Twitter API was used for retrieving Twitter posts from security research and specialist profiles. We chose twelve Twitter profiles which shared the following characteristics:

- They have to be known as security specialist or they should proclaim themselves as a security expert.
- The majority of tweets in their timeline should be about security-related subjects and technology.

Six of these Twitter profiles are from well-known research security experts with an average number of followers of 18,800. The other six are from less-know security specialists, with an average number of followers of 1,100. We split the profiles in this way in order to consider the opinion from different groups. The collected tweets have a one year range from early March 2016 to early March 2017. The total number of tweets gathered was 11,833. All tweets were considered into the dataset. It means that posts of conversation using @[username] and Retweets (RT) was used to fit the proposed model. Retweet posts are posts marked with the abbreviation RT at the beginning, which generally refers to posts of another user on Twitter. At the end of the process, we reached an imbalanced dataset with more instances from non-desirable class than (general alerts) desirable class (useful alerts). Sometimes, in datasets like this, the weights of the desirable class are adjusted for achieving better prediction.

3.2 Labelling data

The posts were split into two classes. The desirable class, which contains relevant posts regarding alerts about software vulnerabilities will be referred to as useful alerts and the other part, the non-relevant information, will be called as general alerts. Twitter posts considered as useful alerts will be those alerting about a new vulnerability found in software or new exploits being used to hack those flaws, as well posts alerting software users about security patches and updates. The non-relevant information will be those which have no warning characteristic, i.e.: tweets about opinion, general discussion and topics with security content but which has no sense of alert. In table 1 and table 2, we can see examples of tweets considered as useful alerts and general alerts respectively. During the pre-processing of features, the URL of tweets was replaced by the mark [URL]. We manually categorize the aim of useful alerts tweets in table 1 as “Security Fix/Patch alert”, “Vulnerability alert”, “Exploit alert”. We did the same for general alerts tweets in table 2 as “Daily events”, “Conversation”, “Announcement”, “Opinion”.

Table 1: Useful tweets and categories.

Tweet	Category
Time to update if you have an iPhone[URL]	Security Fix/Patch alert
If you have an iPhone, make sure you update today. Big list of fixes: [URL]	Security Fix/Patch alert
CVE-2016-9838 - Joomla! Account Takeover Remote Code Execution [URL]	Vulnerability alert
Update your WhatsApp [URL]	Security Fix/Patch alert
If you're using the Zotero Desktop app, it's vulnerable to DNS rebinding. Following macOS firewall rule should miti… [URL]	Vulnerability alert
iOS 9.3.2 &amp; OS X 10.11.5 are out with security fixes. Update now! [URL]	Security update alert
Cisco have announced more browsers, plugins and versions affected by WebEx vulnerabilities. [URL]	Vulnerability alert
More Symantec and Norton remote code execution vulnerabilities fixed today, full advisory is here	Security Fix/Patch alert
Multiple remote memory corruption vulns in all Symantec/Norton antivirus products, including stack buffer overflows [URL]	Vulnerability alert
Screen root exploit 4.5.0 [URL]	Exploit alert
Systemd v228 local root exploit (CVE-2016-10156) [URL]	Exploit alert
Multiple vulnerabilities found in Quanta LTE routers (backdoor, backdoor accounts, RCE, weak WPS ...) [URL]	Vulnerability alert
Samsung SmartCam iWatch Root Exploit [URL]	Exploit alert
74k modems possibly affected in CZ Czech Republic [URL]	Hacking alert

Table 2: General tweets and categories

Tweet	Category
My best talk was probably remote jeep hack (2015). My worst was probably battery hacking (2011).	Daily events
This will be the first year I haven't submitted to @{useraccount} since 2006.	Daily events
@{useraccount} @{useraccount} how do you even have a ticket already?	Conversation
Locked out of my banking site because I bought a new computer and can't remember what I thought my favorite TV show was as a kid	Daily events
One funny thing is my current boss and my last boss both made the list. I gotta go into management!	Daily events
No one who has hacked a car has given or taken car hacking training. Wanna learn? Read the 500+ pages written by @{useraccount} I 4 free	Announcement
We'll be publishing our white paper in the next few days, keep your eyes peeled for it.	Announcement
RT @{useraccount}: If you're new to vulnerability research or thinking about starting, come see @{useraccount}and I's talk at @{useraccount}24! [URL]	Announcement
I'm afraid to report vulns now because I don't want to get made fun of like the bad lock folks.	Opinion
RT @{useraccount}: An immutable law of security research: if you find a vulnerability, someone will describe your handling of it as 'irresponsi…	Opinion

4. Building the supervised model

In this section, we discuss how the words in tweets were pre-processed to produce the features set. A briefly explanation about the algorithm chosen will be given, as well a discussion about the parameters used into the algorithm.

4.1 Pre-processing and chosen features

The feature for the model will be based on all words that we have in our test set. In order to have meaningful words only, all links, number, dates, year and stop words like preposition and articles are removed. With a now clean sample, we create a bag of words representation with the frequency of each term used as feature for the model as seen in Zhang, Jin and Zhou (2010). The experiment is going to test the values of classification by comparing three different sets of word features, or n-gram set. The first set will use the frequency of unique word of the test set, or n-gram of size 1 (unigram), for instance, the sentence "update software" has 2 terms, with frequency of 1 each word. The second will use the frequency of pairs of words, n-gram of size 2 (bigram), by counting words that appears together in the sentence, for instance, the phrase "update software" has 2 terms, but the frequency is 1 to both words. Finally, the third set will use the previous sets together, i.e. the unigram set plus bigram set.

4.2 Support Vector Machine algorithm

Support Vector Machine (SVM) is a supervised learning method introduced by Cortes and Vapnik (1995) in which this paper will be used for classification task. The experiment of Joachims (1998) compared SVM with other classification models which has demonstrated better results utilising samples with high dimensional features. In addition, the papers in Nunes et al (2016) and Sabottke, Suciu, and Dumitras (2015) reach significant results with SVM for classification of cyber discussion in social media. Based on these findings we are going to use the same algorithm for this experiment. It was used the Scikit-learn to implement the algorithm. Different values of parameter C was used in order to adjust weights of the model for better classification rates. Finally, the kernel tested in this experiment was the Radial Based Function (RFB) expressed by $K(x, x') = \exp(-\gamma\| x - x' \|^2)$ where $\gamma = 1/\sigma^2$.

5. Results

For evaluation of the model, the hold-out method evaluation was used, which consist in split the dataset values into training and testing sets. For this experiment we are going to use 80% training and 20% for testing. The results in this section is giving by F1-score, prediction and recall of the features set unigram, bigram and unigram + bigram adjusted by values of C ranging from 1 to 1500. We see in Figure 1, that the result in terms of accuracy does not change significantly into the features set by adjusting C. It ranges between 92% to 94%. In Figure 2, we can see different F1-score of the model by n-gram features by adjusting C. The best F1-score, 55%, is reached by using unigram + bigram features and parameter C=500 or C=1000 and using bigram features with C=1000. In addition, we see poor performance of bigram compared to the other two features set. The Figure 3 shows the precision, recall and F1-score measure of the model by adjusting the values of C. The highest F1-score was achieved by using unigram features with C=1000 and by using unigram + bigram features with C=500 and C=1000. We can observe by the bar graph that precision and recall are inversely proportional, when one increase, the other decline.

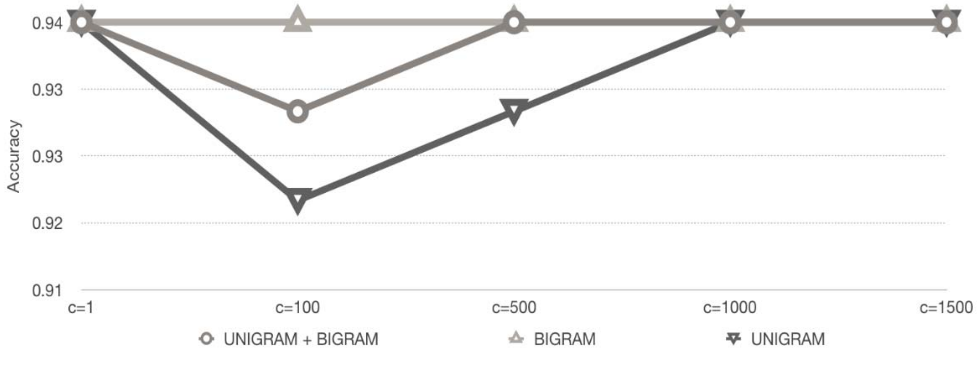

Figure 1: Accuracy of the model showed by n-gram features with C ranging from 1 to 1500

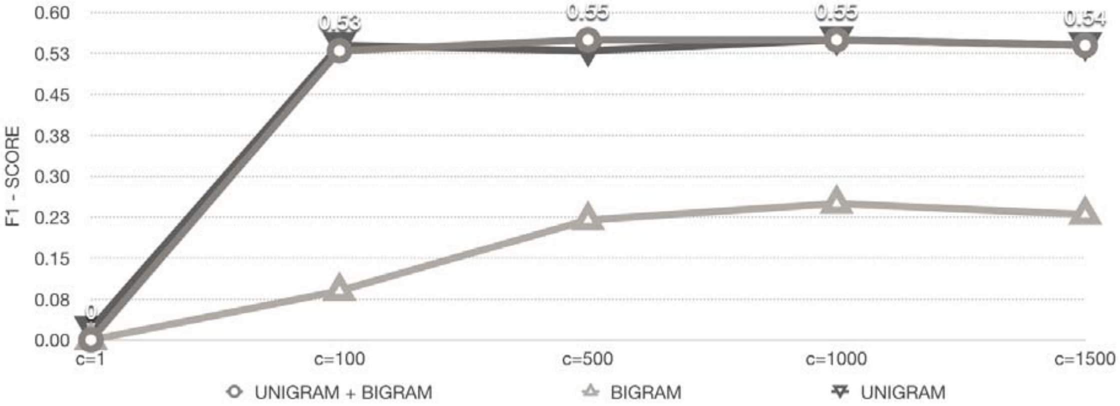

Figure 2: F1-score of the model showed by n-gram features with C ranging from 1 to 1500

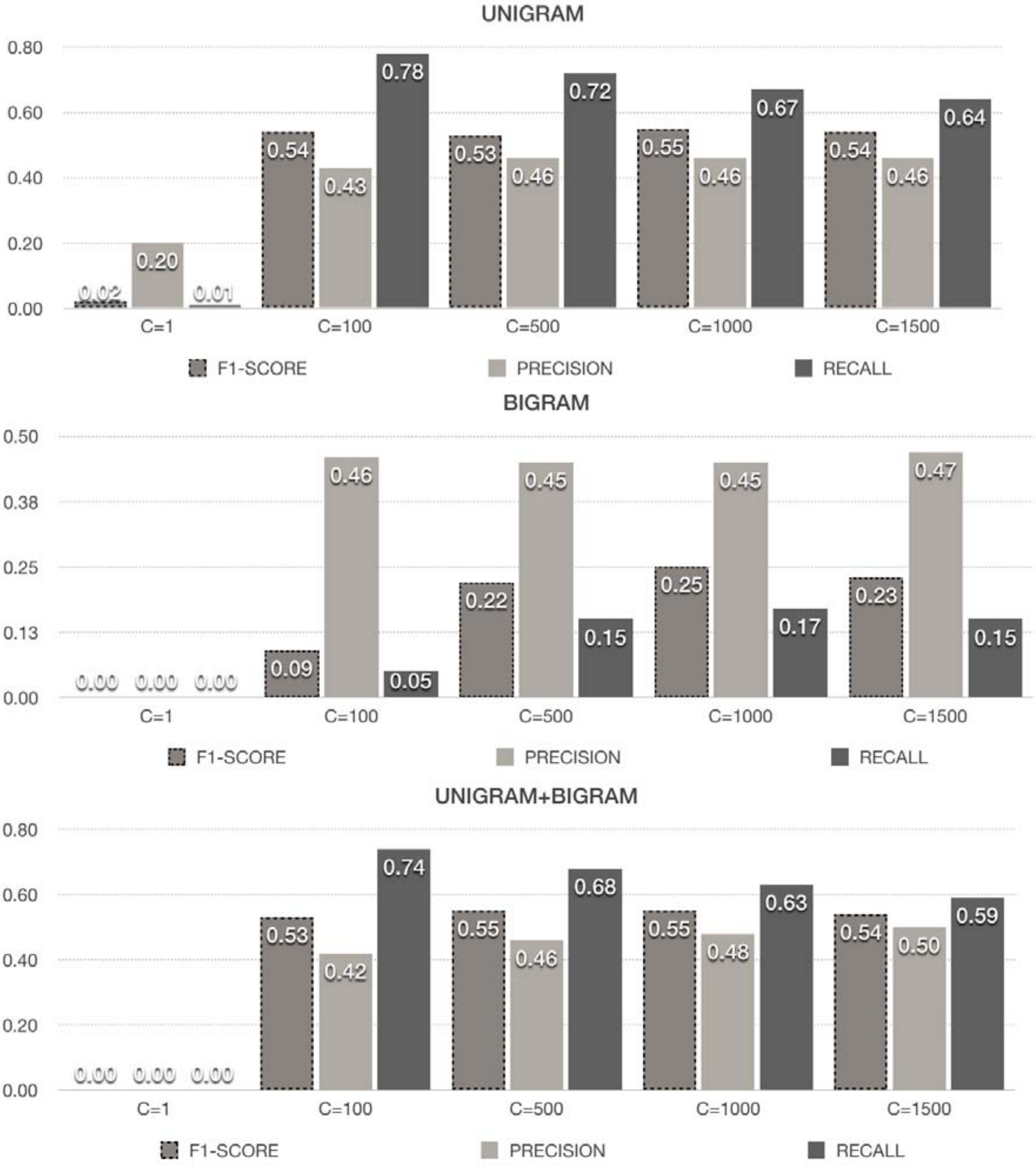

Figure 3: F1-score, precision and recall of the model showed by n-gram features with C ranging from 1 to 1500.

The criteria used for choosing the features set and parameter for this experiment was the combination of values: highest F1-score and lower C. As result, the model with features set unigram + bigram features and parameter C=500 is used to show the examples of classification and misclassification from the tables 3, 4 and 5.

In table 3, we see instances of useful alerts correctly classified by the model. In table 4 is seen instances of useful alerts wrongly classified as general alerts and in table 5 is seen instances of general alerts wrongly classified as useful alerts.

Table 3: Correct classification - useful alerts correctly classified and information relevance

Tweet	Relevance
CVE-2016-9838 - Joomla! Account Takeover and RCE writeup [URL]	Joomla vulnerability disclosed on 1st December 2016 by security focus
Samsung SmartCam iWatch Root Exploit [URL]	Exploit which allows a remote command execution into the device.

Table 4: Misclassification - Useful alerts classified as general alerts and information relevance

Tweet	Relevance
Certificate pinning vuln in Firefox and Tor Browser [URL]	regarding a CVE-2016-9079 affecting Firefox and Tor browser
Mac users hit by rare ransomware attack, spread via Transmission BitTorrent app [URL]	Regarding the malware OSX/Filecoder used to crypto lock files on OSX Operational System

Table 5: Misclassification - General alerts classified as useful alerts and information relevance

Tweet	Relevance
Zero-day exploits that attack zero-day exploits can also be attacked by zero-day exploits	Personal sentiment about the topic, without sense of urgency
The Firefox exploit is almost identical to the Tor exploit version	Maybe he is talking about the TOR/Firefox vulnerability, but his tweet does not have sense of urgency
RT @{username} I've always wanted to find an 0-day I could tweet	Personal sentiment, without sense of urgency

6. Discussion

The shortness of Twitter posts, which has a maximum of 140 characters, is challenging not just for machine learning models, but for human specialist as well. In the labelling phase, it was necessary to open the link inside of posts in order to reason about the message posted by the user. Most of the time, the link redirects to a larger text (web report, blogs review, article) which explains further about the subject of the message which it was aimed for. When we were uncertain how to label the tweet, the content of the URLs was able to give us this insight.

It was noted that some keywords which related to software security field can appear in both classes of tweets. For instance, as seen in table 1 and table 2, the word “vulnerability” might appear in tweets from useful alerts class and in general alerts message

7. Conclusion

Twitter is a commonly accepted tool for exchanging and disseminating information among peers. In the context of computer security, security experts are using Twitter for publicly disclosing their discoveries. It is within this context that this research takes place. In this paper, we created a classifier to extract useful alerts from Twitter posts by security experts. This research can be useful for monitoring news disclosed by specialists on Twitter social media platform and keeping users and vendors informed about issues in cyber security before knowledge becomes widespread through traditional notifications.

The model presented in this paper performed well in terms of accuracy measure, 94%. In terms of precision, 46%, the value can be raised by adjusting the penalty parameter C to levels higher than 500. However, it has an impact on recall, 68%, which starts to decline, and it can contribute to overfitting the model.

For the type of classification problem in this paper, we should consider models which give us good recall. High recall for the desirable (useful alert) class means that the model is able to correctly classify the majority of these instances. However, the low level of precision means that some instances from non-desirable class (general alert) will be wrongly classified as useful alerts, rising the rates of false alarm. The difficulty of classification is given by the fact that a lot of words in Twitter posts can appear for both classifications. For instance, the word 'vulnerability' can appear in a post which might be warning about a new vulnerability or a specific vulnerability affecting software at the moment. However, it can appear in a post which the specialist is explaining about a good book or a conference about a vulnerability which falls into the general alert. This type of replication of class requires some form of intervention to verify the sentiment.

Finally, we could see that most of the useful posts has links to other sources of information which has a more detailed explanation about the security issue, sometimes it refers to blogs or specialised articles.

8. Future work

The result showed us that although the outcome is positive, there is the potential to significantly improve the results. Such improvement can be achieved by using the links provided in tweets which have an additional source that can be leveraged for increasing the prediction of the model. In addition, retweets of information or similar announcements of a vulnerability among accounts can be a sign of a novelty discovered, in order to deal with that, a graph analysis of these interaction could be used as features into the model. A technique called sentiment analysis as seen in Wang et al (2012) can be another approach that can be utilised to improve our results. it is most likely that the useful alerts and discoveries posted by security experts come with some level of sentiment. Knowing the sentiment of a tweet, we can determine whether the sentiment can be used as a feature for our model to improve its classification rates. By adding these new features to the model we expected to use lower values for penalty parameter C in order to avoid model overfitting.

Acknowledgements

Andrei Lima Queiroz would like to thank the scholarship granted by the Brazilian Federal Programme *Science without Borders* supported by CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico).

References

Benjamin, V., Li, W., Holt, T. and Chen, H. (2015). “Exploring threats and vulnerabilities in hacker web: Forums, IRC and carding shops”. *In Intelligence and Security Informatics (ISI)*, May, pp 85–90.

Cataldi, M., Di Caro, L. and Schifanella, C. (2010). “Emerging topic detection on Twitter based on temporal and social terms evaluation”, *Proceedings of the Tenth International Workshop on Multimedia Data Mining*, pp. 1-10.

Cortes, C. and Vapnik, V. (1995). “Support-vector networks”, *Machine Learning,* Vol 20. No. 3, pp 273-297.

Dunphy, P., Vlachokyriakos, V., Thieme, A., Nicholson, J., McCarthy, J. and Olivier, P. (2015). “Social media as a resource for understanding security experiences: A qualitative analysis of #password tweets”, *In Eleventh Symposium On Usable Privacy and Security,* July, pp 141–150.

Joachims, T. (1998). “Text Categorization with Suport Vector Machines: Learning with Many Relevant Features”, *Proceedings of the 10th European Conference on Machine Learning*, pp 137-142.

Lippmann, R. P., Campbell, W. M., Weller-Fahy, D. J., Mensch, A. C., Zeno, G. M., and Campbell, J. P. (2016). Finding malicious cyber discussions in social media. *Lincoln Laboratory Journal*. Vol 22, No. 1, pp 46-59.

Mittal, S., Das, P. K., Mulwad, V., Joshi, A. and Finin, T. (2016). “CyberTwitter: Using Twitter to generate alerts for Cybersecurity Threats and Vulnerabilities". International *Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp 860-867.

Nunes, E., Diab, A., Gunn, A., Marin, E., Mishra, V., Paliath, V., Robertson, J., Shakarian, J., Thart, A. and Shakarian, P. (2016). Darknet and deepnet mining for proactive cybersecurity threat intelligence. CoRR, abs/1607.08583.

Sabottke, C., Suciu, O. and Dumitras, T. (2015). “Vulnerability disclosure in the age of social media: Exploiting twitter for predicting real-world exploits”*, 24th USENIX Security Symposium (USENIX Security 15)*, August, pp 041–1056.

Sakaki, T., Okazaki, M. and Matsuo, Y. (2010). “Earthquake Shakes Twitter Users: Real-time Event Detection by Social”, *Proceedings of the 19th international conference on World wide web* (WWW '10), pp 851-860.

Sankaranarayanan, J., Samet, H., Teitler, B. E., Lieberman, M. D. and Sperling, J. (2009). “Twitterstand: News in tweets”, In Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS ’09, pp 42–51.

Trabelsi, S., Plate, H., Abida, A., Aoun, M. M. B., Zouaoui, A., Missaoui, C., Gharbi, S., and Ayari, A. (2015). “Mining social networks for software vulnerabilities monitoring”, *In 2015 7th International Conference on New Technologies, Mobility and Security (NTMS)*, pp 1-7.

Wang, H., Can, D., Kazemzadeh, A., Bar, F., and Narayanan, S. (2012). “A system for real-time twitter sentiment analysis of 2012 U.S. presidential election cycle”, *In Proceedings of the ACL 2012 System Demonstrations, ACL ’12,* pp 115–120.

Zhang, Y., Jin, R. and Zhou, Z. H. (2010). “Understanding bag-of-words model: a statistical framework”, *International Journal of Machine Learning and Cybernetics,* Vol 1, No. 1, pp. 43-52.