Extracting Cyber Threat Intelligence From Hacker Forums: Support Vector Machines versus Convolutional Neural Networks

Isuf Deliu *, Carl Leichter †, Katrin Franke ‡
Department of Information Security and Communication Technology
Norwegian University of Science and Technology
Gjøvik, Norway
*isufdeliu@gmail.com *, carl.leichter@ntnu.no †, katrin.franke@ntnu.no‡*

***Abstract*—Hacker forums and other social platforms may contain vital information about cyber security threats. But using manual analysis to extract relevant threat information from these sources is a time consuming and error-prone process that requires a significant allocation of resources. In this paper, we explore the potential of Machine Learning methods to rapidly sift through hacker forums for relevant threat intelligence. Utilizing text data from a real hacker forum, we compared the text classification performance of Convolutional Neural Network methods against more traditional Machine Learning approaches. We found that traditional machine learning methods, such as Support Vector Machines, can yield high levels of performance that are on par with Convolutional Neural Network algorithms.**

***Keywords*-Cyber Threat Intelligence (CTI); Open-Source Intelligence; Cyber Security; Hacker Forums; Machine Learning; Convolutional Neural Networks; Text Classification;**

I. Introduction

Cyber-protection is a top priority for modern civilization. Traditional host and network level security controls can detect and prevent a large proportion of cyber attacks; but modern societies are struggling to keep pace with the increasing sophistication of attack tools and methodologies. Cybercriminals are continually producing more advanced and targeted attacks which are able to circumvent conventional security controls (eg: firewalls, intrusion detection and prevention systems, etc). In addition, the existing controls are mainly reactive; that is, they are updated "after the fact" using information based on the outcomes of previous successful attacks. More proactive approaches are necessary to increase the effectiveness of cyber security protection.

Cyber Threat Intelligence (CTI) is becoming increasingly important within the cyber security community [1], [2]. The main idea of CTI is the enrichment of traditional security controls by using information collected from multiple diverse sources, both in-house and external. Future attacks are anticipated through research on the tools, methods and intentions of potential threats. This allows security controls to be updated in a timely manner and thereby increases the chances of detecting and preventing malicious activities before an attack even occurs. A variety of threat intelligence sources exist, including open sources and proprietary vendor feeds. According to a SANS survey [3], sharing within the community remains the primary source of intelligence, while open sources were regarded as important by only half of the respondents. Open sources include (but are not limited to) online blogs, forums, social networks, news, Dark Web sources, etc.

As computational power increases, deep learning algorithms have become the state-of-the-art for many machine learning application domains [4], [5] and have produced excellent performance in computer vision tasks such as image recognition[6]. This has led to new models for other application domains, such as text analysis and document classification. There are several different deep learning architectures that can be used for text analysis, including **Recursive Neural Networks (RecursiveNN)** and **Recurrent Neural Networks (RecurrentNN)**, but the scope of this paper is limited to feed-forward **Convolutional Neural Networks (CNN)**. Lai et al. in [7] explained why CNN are more suited than other architectures for document classification. According to their explanation, the memory requirements in the tree structure of RecursiveNN make them very ineffective when handling large documents. In contrast, there are several models of feed-forward convolutional architectures that have shown remarkable success in text categorization [8], [9], [10], [11], [12].

This paper will focus on obtaining cyber threat intelligence from open source hacker forums and perform an empirical comparison of the classification performance of a Convolutional Neural Network against conventional methods such as SVMs.

II. Related Work

The research on hacker platforms has been focused mainly into four streams: (i) social structure of their members [13], [14], [15], (ii) identification of key hackers [16], [17], [18], [19], [20], [21], (iii) understanding of hacker language [22], [23], and (iv) identification of emerging threats [24], [25], [21], [26], [27]. In practice however, little has been done to leverage intelligence from these sources to enrich existing security measures. None of the 11 CTI portals and 14

978-1-5386-2715-0/17/$31.00 ©2017 IEEE

malware analysis portals reviewed in Samanti et al.[25] collected data from hacker communities. Similarly, only 20% of SANS [3] respondents state they have considered integrating vendor based Dark Web intelligence into their CTI platforms.

The identification of the specific hacker forum(s) to be used as data sources is usually performed through keyword searches and by seeking experts' opinions [18], [22], [24]. Additional sources are further identified by using snowball sampling in the already identified data sources. Unfortunately, there are a limited number of available forums that have already been the subject of scientific research. Furthermore, to the best of our knowledge, the only available English forum used in the existing literature is Hackhound [24].

Nunes et al. [28], uses binomial classification to categorize hacker forums posts and marketplace products into two classes: (a) relevant and (b) irrelevant to cyber security. The authors reported the discovery of 16 zero-day exploits by monitoring multiple marketplaces from Darknet over a period of 4 weeks. In order to reduce the need for manual labeling, the authors combined traditional supervised algorithms such as Naive Bayes, SVM, Random Forests, etc. with semi-supervised classifiers such as Label Propagation and Co-Training. Marin et al. [29] used unsupervised classification to explore products offered in different marketplaces in the Dark Web. These products were grouped into 34 different different categories by applying an unsupervised k-Means algorithm on features that were constructed from character level n-grams based on the title of the products offered.

Meanwhile, Ebrahimi et al. [30] applied deep Convolutional Neural Networks to detect predatory conversations in social media. In contrast to our results: their deep learning architecture outperformed the traditional classifiers in their study.

III. Methodology

We compare the performance of different Machine Learning methods in locating hacker forum posts that may contain relevant cyber threat intelligence. As noted in the Related Work section, the only English forum we found in the existing literature was Hackhound in [24]. Unfortunately, Hackhound only has a relatively small number of posts (4,242). So instead we used data from a forum called Nulled.IO. A complete database of Nulled.IO has been leaked, so special web crawlers are not needed to obtain the data. The original forum as leaked can be obtained from http://leakforums.net/ thread-719337.

A. Dataset Construction

We created two supervised machine learning datasets from the Nulled.IO forum: a binomial data set and a multi-nomial data set. The binomial data set consists of 16,000 posts, that are equally divided betwen a cyber security *relevant* class and an *irrelevant* class. The *relevant* class data were selected from forum posts that contained common security keywords [table I] based on the author's experience and the related work in section II (Related Work).

Table I
The list of keywords used to label the class Relevant

Common Cyber Security Keywords
adware, antivirus (kaspersky, avast, avira, etc.), backdoor, botnet, chargeware, crack, crimeware, crypter, cve, cyberweapon, ddos, downloader, dropper, exploit, firewall , hijack, infect, keylogger, logic bomb, malware, monetizer, password, payload, ransomware, reverse shell, riskware, rootkit, scanner, security, shell code, spam, spoof, spyware, stealware, trojan, virus, vulnerability, worm, zero-day, zeus

Since the list of common security keywords is not comprehensive, a complementary approach was used to label the security *irrelevant* class. Data was labeled as irrelevant should it satisfy the following two conditions: (a) none of the security keywords from table I were present in the text and (b) non-security related keywords were also present in the text. The list of general keywords consist of terms related to sport (football, basketball, etc.), music (song, rap, pop, etc.), movies (series, film, episode, etc.), drugs (marijuana, heroin, etc.), etc. An example post for both classes is shown in table II.

Category	Example
Irrelevant	Hello dear nulled.io community. This is a simple question, what are your favourite movies? ; ; Mines? Idk. Probably Jackie chan movies and/or taken series
Relevant	NEW UPDATE: CVE-2015-1770 + CVE-2015-1650 This SILENT Exploit works on all Operating systems, works on all versions of Word EXCEPT Office 2013. it is a dual exploit in one builder - a combination of two different CVE's, it doesn't require any ftp or c...

Table II
Example posts from binomial dataset

The multinomial data set has a single class/label for the data from the *irrelevant* posts and then several different classes/labels for the *relevant* posts. The number and type of classes in our experiments were chosen based on two criteria: (a) coverage of different aspects of information security, and (b) relevance to practical scenarios. The multinomial class labeling was also performed based on a keyword search; additionally, posts were manually reviewed to ensure comprehensiveness. For example, a post may have contained keywords such as *username, passwords, login, email, etc.* and after manual review it was labeled as belonging to the *credential leakage* class. This dataset contains 10,000 labeled posts distributed into categories as shown in figure 1. NB: this distribution is not uniform, as credentials posts and posts not related to security each take up 25% of the total samples, spamming taking 15%, RATs 10% and finally the remaining categories have equal shares of the distribution at about 5% each.

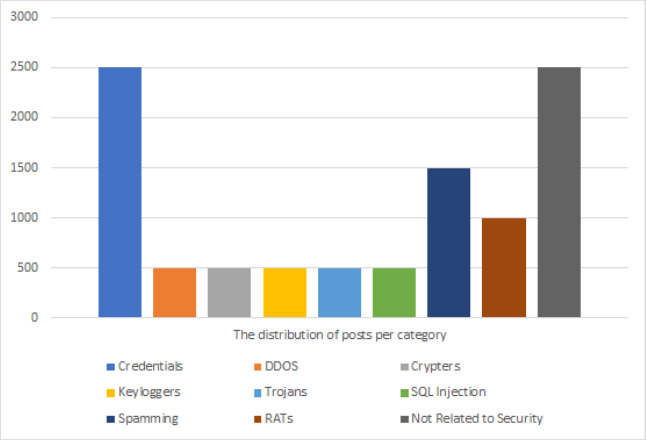

Figure 1. The distribution of data samples

We validated our labelling by comparing our results against labels that were independently generated by 5 Information Security MS/PhD students. 50 posts were randomly selected and reviewed by all 5 students and each student separately labelled all 50 posts. Their responses were approximately 90% consistent with our labelling. Most of the differences in the remaining 10% were for the posts that we had designated as belonging in the spam class; while student respondents usually classified these spam related posts into the "Not related to security" class.

B. *Preprocessing*

There are two sub-phases in the data preprocessing: preparation and cleaning. Data preparation extracts task-relevant fields from the raw data and then stores the result in a useful format (e.g. relational database). This step is usually required to pre-process the raw HTML format data collected by web crawlers and special text parsers are also needed to extract fields such as the content of the posts, title, author, date, etc. But in this case, the data used in this paper is from the Nulled.IO leak, which had already been made available as a relational database, so a parser was not required. However, the contents of the posts were stored together with their HTML tags, so tags were removed prior to further preprocessing. We selected only the title and contents from each post for further analysis. After removal of duplicates, the posts are stored as a one-line document and passed on for further preprocessing.

Data cleansing removes parts of the data that act as noise and do not contribute to task performance. Since we seek to compare different classification models, the cleansing method used in this paper is the same as that used by Kim[1] in [11].

C. *Classification*

We use supervised classification methods to classify forum posts into different categories as related to cyber security. Each separate post was considered as a separate document and we applied machine learning methods for document (text) classification. We included only the post contents, and not the post title. This choice was mainly due to the large number of posts using the same title (topic); the frequent appearance of the same titles in the data would bias the results.

The following is a brief overview of the supervised classification algorithms we used for classifier performance comparisons. We are seeking to test the following hypothesis:

H1: Convolutional Neural Network methods will outperform traditional machine learning methods in the task of classifying hacker forum data.

1) Traditional Classifiers with Bag-of-Words features: Bag-of-Words is a common method of representing text documents as feature vectors. The vectors' dimensionality is equal to the number of unique words over all the documents in the dataset, with vector component values based on word frequency. Any of the following three frequency counts may be used as feature vector component values: *Binary:* designates the presence of the word in the docume: 1 if the word in present, 0 if absent; *Raw Frequency:* the frequency of the word within the document; *Normalized Frequency* (eg: term frequency–inverse document frequency or TF-IDF): the frequency of each word in a given document is normalized by the sum of occurrences of that word in all documents.

Bag-of-Words features are popular for their simplicity and they usually perform well; but they ignore word order [31], [32]. So documents that contain the same words, but in different order, will have the same feature vector representation.

2) Traditional Classifiers with n-gram features : N-gram features are sequences of *n* consecutive words/characters and they consider word order in the local context. These features are constructed by enumerating all the possible characters/words of length n. For example, if the post is "You have been hacked" then the possible 3-grams are 'You have been' and 'have been hacked'. Local context comes into play when we consider the 3-grams: 'You have been' versus 'have you been'. The second 3-gram cannot be extracted from the post and it is an example of the kind of word order distinction that "Bag-of-Words" will fail to accommodate.

[1] https://github.com/yoonkim/CNN_sentence/blob/master/process_data.py

3) Convolutional Neural Networks utilizing a variety of feature vectors (to be described): In this paper, we use a CNN model proposed by Kim [11] mainly for two reasons: (i) the simplicity of the architecture and (ii) its high performance in natural language processing tasks (such as sentiment analysis). The architecture of this model (figure 2) consists of a single convolutional layer, followed by a pooling layer and then a fully-connected layer for the output.

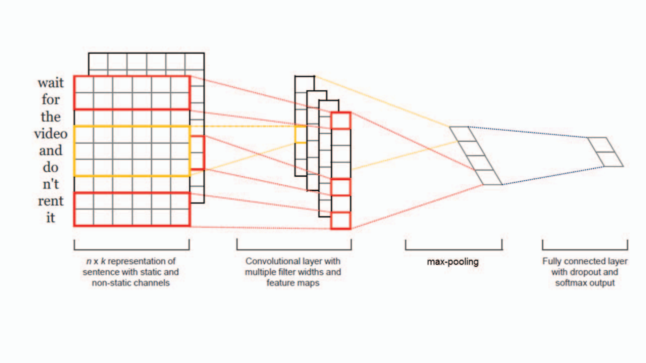
wait
for
the
video
and
do
n't
rent
it
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
max-pooling
Fully connected layer with dropout and softmax output

Figure 2. Convolutional Neural Network for text classification [11]

Convolutional Neural Network features

The input to this architecture is the vector representation of the documents (e.g. forum posts). This is achieved by regarding documents as sequences of words and concatenating the corresponding vector representations for each of the respective words. The word-vector representations, known also as Word Embeddings, are models that map words onto real-valued vectors whose values capture semantic information about the words [33]. Because the CNN were designed to work with images, they only accept the input as a fixed sized matrix and are not capable of working with variable length inputs. For the purposes of variable string length input, this limitation is resolved by zero-padding all the documents so they all have the same length as the longest document in the dataset. Four different types of word-vectors representations are used in this paper. To more clearly identify and refer to these representations, the two types of pre-trained vectors acquired from external sources are designated as "external" and the vectors that were specifically trained in this study are designated as "internal".

- external data
  - Pre-trained vectors from Google (word2vec)
  - Pre-trained vectors from Stanford University (GloVe)
- internal data
  - Vectors trained on the Nulled.IO forum data
  - Random Vectors

The pre-trained vectors are trained on external data using two different models, namely word2vec [34], [35] and GloVe [36], whereas the internal vectors are trained using word2vec on the data from the Nulled.IO forum.

The convolutional layer serves to construct intermediate features that will act as input for the max-pooling layer (as seen in figure 2). These intermediate features are constructed from the original input features and are the obtained by applying a non-linear transformation (i.e. activation function) to the output of the convolution layer. The result is a convolution operation between a subset (or filtering) of input neurons and their respective interconnection weights. There are several activation functions that can be used for this purpose: Rectifier Linear Unit (ReLU), sigmoid function, hyperbolic tangent, etc. [4]. Their formal definition is given as follows:

$$\begin{aligned} ReLU : f(x) &= max(0, x) \\ sigmoid : f(x) &= \frac{1}{(1 + e^{-x})} \\ tangent : f(x) &= \frac{1 - e^{-2x}}{1 + e^{-2x}} \end{aligned} \tag{1}$$

These convolution operations shift the input filtering for a fixed height and width, to obtain values of the feature for different locations. The weight of the filter is based on the text specifications: it is kept constant and equal to the dimensionality of word vectors [37]. On the other hand, the height of the filter is varied. Multiple such filters (also known as feature maps) are applied to obtain multiple features. To maintain consistency with the literature we are going to refer to filter width as "filter region size". The convolutional architecture used in this paper supports multiple region sizes(e.g. |3,4,5|), and the result of respective convolutions are averaged to obtain a single output value. The convolutional layer is followed by a pooling layer whose role is analogous to feature selection in the machine learning process. Seeking to select only important features which preserve the relationship of the data, pooling takes the average (or maximum value) of the features and reduces the number of features down to the number of feature maps. Following the original paper [11] and the practitioner's guide [37], we use the max-pooling strategy. The last layer of this architecture is a regular fully-connected layer whose role is classification of the documents into their respective categories.

IV. Experiments and Results

The emergence of relatively inexpensive high performance GPU computing platforms has made Deep learning a popular topic of research [5]. Our CNN experiments used an NVIDIA GeForce GTX 1060 GPU card with 1280 CUDA cores and 6GB of GDDR5 RAM. Our traditional classifier experiments utilized the *scikit-learn*[2] python library. In order to avoid incorrect results due to programming mistakes,

[2] http://scikit-learn.org/stable/

we first replicate the results of other public datasets from the existing literature [37]. Code quality was tested with a sentiment data classification experiment and these results can be found in [38, p. 150].

A. *Classification of Binomial Dataset*

The binomial data set classification performance of the traditional machine learning algorithms were obtained by averaging the results from running 10-fold cross validation 10 times. In each iteration (fold), a random sample of approximately 10% was hidden from the training algorithm and used as test data. Even though the SVM is the one of the most common traditional classifiers in the existing literature [37], [39], we also compared its performance to other classifiers such as Decision Trees and k-Nearest Neighbors(k-NN), and the results are shown in table III. We utilized the default parameters specified in the scikit-learn library for the three classifiers. For example, we used k=5 on the k-NN, a linear SVM with a penalty value C=1.0 and 1-vs-N strategy for multiclass classification and so on.

Features	k-NN	Decision Trees	SVM
word (uni+bi)-grams	58.52	97.95	**98.09**
character trigrams	60.75	97.61	**98.60**
character (bi+tri)-grams	68.30	97.43	**98.55**
bag-of-words	61.22	98.11	**98.40**

Table III
CLASSIFICATION ACCURACY OF THREE CLASSIFIERS ON THE BINOMIAL DATASET: K-NEAREST NEIGHBORS, DECISION TREES, AND SUPPORT VECTOR MACHINES

The results clearly indicate that the SVM outperforms k-NN as well as Decision Trees for the given task. The differences between accuracy values of k-NN and SVM are more significant, while the SVM and Decision Trees results results are on par with each other, but SVM does have a slight advantage. Consequently, we also measured the training and testing time [figure 3], which shows that SVM training is much faster than Decision Trees training.

Now, we turn our focus to the performance of Convolutional Neural Network classifiers. The code for the CNN used in this paper was obtained from the Kim[3] in [11]. The default configuration was used: feature maps = 100, activation function = ReLU, max-pooling, dropout =0.5, l2 normalization constraint=3, and filter region size =[3,4,5]. We used the non-static model and trained for 25 epochs. The non-static model automatically adjusts the input vectors for the given tasks[11].

The performance of the CNN on the binomial dataset is given in table IV (where the parameter "D" represents the feature vector dimensionality). The model using input vectors trained internally from the data has better performance than the others. The vectors are obtained by training word2vec with one million posts from the Nulled dataset.

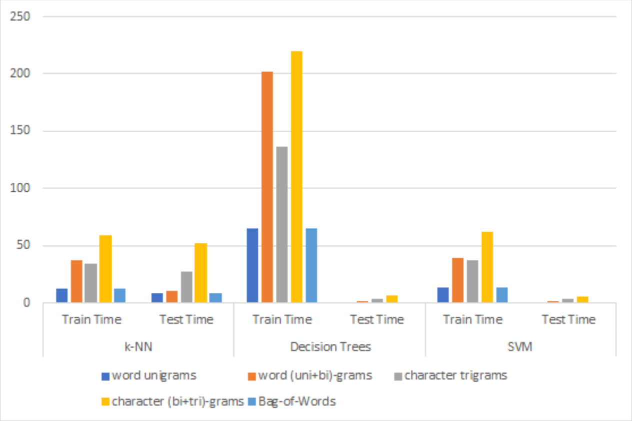

Figure 3. Training and testing time for the traditional classifiers

Algorithm	Accuracy(%)
w2v-CNN	98.22
Glove-CNN D=50	95.67
Glove-CNN D=100	97.04
Glove-CNN D=200	97.64
Glove-CNN D=300	97.65
Random-CNN D=50	95.23
Random-CNN D=100	96.69
Random-CNN D=200	96.91
Random-CNN D=300	97.23
w2vInternal-CNNs D=50	98.73
w2vInternal-CNN D=100	98.67
w2vInternal-CNN D=200	98.75
w2vInternal-CNN D=300	**98.79**

Table IV
DEEP LEARNING PERFORMANCE ON BINOMIAL DATASET

B. *Classification of Multi-Class (Multinomial) Dataset*

All of the previously described classification methods were then applied to the multi-class dataset. Figure V shows a comparison of SVM, k-NN and Decision Trees performance. Just as it did in the binomial dataset experiment, the SVM outperforms k-NN as classifiers and this is also supported by the precision, recall, and F-measure scores.

All of the experiments were using feature vectors whose components are binary valued. We beleive this is the reason for the low k-NN performance: because k-NNs utilize distance metrics that are more suitable to feature vectors with a wider dynamic range that just 1 or 0.

[3] https://github.com/yoonkim/CNN_sentence/

Features	k-NN	Decision Trees	SVM
word (uni+bi)-grams	37.48	96.41	**96.93**
character trigrams	68.07	95.96	**98.62**
character(bi+tri)- grams	81.36	95.98	**98.59**
bag-of-words	66.76	96.45	**97.27**

Table V
CLASSIFICATION ACCURACY OF THREE CLASSIFIERS ON THE MULTI-CLASS DATASET: K-NEAREST NEIGHBORS, DECISION TREES, AND SUPPORT VECTOR MACHINES

Similar to the binomial dataset, we also report the performance of CNN classifiers. All the experiments are run using the same parameters as in the binomial dataset. For a better SVM versus CNN comparison in the multinomial results, we also report three other performance measures in conjunction with accuracy: precision, recall, and F-measure. A low precision value indicates a greater probability of false positives, whereas a low recall indicates a greater probability of false negatives. Meanwhile, the F-measure (F1) represents a balance between precision and recall. The classification performance of SVM is shown in figure VI, while figure VII shows the performance of the CNNs.

Features	Accuracy	Precision	Recall	F1
word (uni+bi)grams	96.93	97.69	95.48	96.51
character trigrams	**98.62**	**98.43**	**98.10**	**98.24**
character (bi+tri)grams	98.59	98.41	98.17	98.28
Bag-of-Words	97.27	97.76	96.07	96.86

Table VI
CLASSIFICATION PERFORMANCE ON MULTI-CLASS DATASET USING SVM CLASSIFIER

Algorithm	Accuracy(%)	Precision(%)	Recall(%)	F1(%)
w2v-CNN D=300	97.74	98.28	96.27	97.22
Glove-CNN D= 50	96.78	96.99	95.33	96.09
Glove-CNN D=100	97.52	97.92	95.98	96.89
Glove-CNN D=200	97.39	97.48	95.95	96.67
Glove-CNN D=300	97.12	97.39	95.31	96.30
Random-CNN D= 50	97.23	97.90	95.70	96.74
Random-CNN D=100	97.41	97.94	95.76	96.77
Random-CNN D=200	97.45	98.27	95.75	96.94
Random-CNN D=300	97.17	98.22	95.24	96.63
w2vInternal-CNN D= 50	97.92	98.08	96.67	97.33
w2vInternal-CNN D=100	97.98	98.07	96.65	97.30
w2vInternal-CNN D=200	98.03	98.19	96.91	97.50
w2vInternal-CNN D=300	**98.10**	**98.24**	**97.02**	**97.60**

Table VII
CONVOLUTIONAL NEURAL NETWORK PERFORMANCE ON MULTI-CLASS DATASET

In general, the multi-class results are consistent with the results from binomial classification, and internally trained vectors show better performance than pre-trained or random vectors. This is supported by the separate evaluation of precision and recall, and the accuracy scores as well as the combination F1 measure.

C. The best of both worlds

A summary of the classification results for both traditional and deep learning classifiers is shown in table VIII. The high classification accuracy clearly indicates that our proposed method of filtering irrelevant posts from hacker forums using machine learning classifiers is feasible and can expedite the acquisition of CTI.

Classifier(Features)	Binary Dataset	Multi-class dataset
k-NN(character (bi+tri)-grams)	68.30	81.36
Decision Tree(bag-of-words)	98.11	96.45
SVM(word (uni+bi)-grams)	98.19	96.93
SVM(character trigrams)	**98.82**	**98.62**
SVM(character (bi+tri)-grams)	98.71	98.59
SVM(bag-of-words)	98.45	97.27
w2v-CNN	98.22	97.74
Glove-CNN	97.65	97.12
Random-CNN	97.23	97.17
w2vInternal-CNN	98.79	98.10

Table VIII
A SUMMARY OF CLASSIFICATION PERFORMANCE

The most interesting result is in the fact that the performance of SVM is on par with the CNN classifiers. The performance difference is so small (e.g 0.03%) that we cannot make a statement on which classifier performs better than the other in terms of accuracy. However, we can state that CNN is much more computationally demanding than SVM. For CNN, the training can take several hours, even when using GPU acceleration, an important factor for the practical consideration of these methods.

V. DISCUSSION

In this paper, we classify the contents of hacker forum posts by regarding each post as a document and then subjecting it to document classification via machine learning algorithms.

Because Convolutional Neural Networks have been very successful in solving similar problems, we tested the hypothesis that the CNN used in this paper will outperform conventional classifiers. However, the results of the experiments show that the traditional SVM classifier performs at least as well as the CNN. Therefore, our hypothesis is not supported. We speculate that this performance parity reflects that there is a relatively simple topology in the feature space: the boundaries between the different classes in the feature space are simple enough for the SVMs to easily separate the classes. In the case of more complex feature space topologies that do not lend themselves to easy separation via SVM, then we would expect the CNN performance to exceed the SVM.

It is important to emphasize that both traditional SVM and CNN yield high classification performance on the given datasets: approximately 98% accuracy. We were surprised by the high performance achieved when random vectors were used to represent individual words and then concatenated for input to the CNN. Perhaps the CNN is able to successfully adjust its weights during training, even when the inputs vectors are generated at random. It should be noted that these random vectors are reused for each occurrence of a given word in the dataset and even though they are random, their values are limited to the range suggested by Kim [11].

To the best of our knowledge, there is no other work studying the performance of convolutional neural networks on hacker forum posts. Meanwhile, Nunes et al. [28] used a traditional SVM classifier with n-grams as features in a similar dataset. Since the Nunes dataset is not publicly available, a comparison of results is impossible. However since we are reporting significant performance differences from Nunes, we have identified some possible reasons: first, unlike Nunes. we do not preserve the title (thread topic) features. The forum is organized into topics, where each topic can have hundreds or thousands of posts. There are many threads that start out with security relevant posts that have security relevant titles; but as a thread evolves and its post contents drifts into security irrelevant topics, the security relevant title is usually retained. Additionally, given the short length of the posts, the features from the title can exert a disproportionate influence on classification performance. Furthermore, our binary dataset consists of significantly more training samples (16,000) than the dataset in [28]. We believe that the difference in post length between classes in binary classification and the presence of specific keywords in multi-class classification are two reasons for our observed high accuracy. In general, posts related to security have on average more words than posts unrelated to security. This is at least true for the analyzed forum, but an assessment of the generalization of this assumption is beyond the scope of this paper. During the labeling of the posts, we have also noticed that for some classes the use of some words is unavoidable. For example, posts about denial of service attacks usually contain the terms "(D)DOS", "IP", "flood", "network", "ACK", etc. , while these words are rarely used when posting about SQL injections. We would expect such differences to improve multinomial classification performance.

VI. Conclusion

Very little has been done to use the contents of hacker discussion forums to enhance cyber security. We believe this is due to: 1) the enormous number of posts contained in these forums and 2) a significant proportion of the posts are security irrelevant. These two factors makes it practically impossible to manually extract CTI from hacker forums. As a solution to this problem, we demonstrated the utility of supervised machine learning algorithms for classifiying hacker forum posts. In a comparison of forum post classification performance, we found that a conventional SVM produced results that were on par with more modern Convolutional Neural Networks. Given the computational complexity of CNN architectures, our results indicate that SVMs are superior for the purposes of practical, real-time CTI applications.

VII. Future Work

The research done in this paper has many potential extensions. First, the performance of other deep learning algorithms should be compared with the one used in this paper. Thus, in addition to testing other models of feed-forward neural networks, the utility of other architectures (such as recurrent or recursive deep neural networks) should be explored and studied.

In addition, further experiments should be performed using other data sources. In this paper, we only considered posts written in English. We are confident that relevant cyber-threat intelligence can also be extracted from hacker forums in other languages, such as Russian and Chinese. While our analysis was performed using data from a single representative forum, the cited literature also shows that other forums and platforms (such as underground market-places and IRC chat rooms) can also be valuable sources of intelligence. So the scalability of our methods should be tested by extending them to other forums and platforms.

Furthermore, our speculations about the performance parity between SVMs and CNNs should be explored further. However, such a test would require some measurement and manipulation of feature space topology, which is itself an interesting line of inquiry.

Finally, the forum posts that have been classified as security relevant should be subject to additional analysis for more refined CTI. Even after the removal of irrelevant posts, millions of relevant posts may remain and could be analyzed by unsupervised machine learning methods, such as topic modeling, to produce more refined CTI.

Acknowledgment

The authors would like to thank Professor Hai Thanh Nguyen for his valuable suggestions and feedback.

References

[1] D. Chismon and M. Ruks, “Threat intelligence:Collecting, Analysing, Evaluating,” MWR InfoSecurity, Tech. Rep., 2015. [Online]. Available: https://www.mwrinfosecurity.com/assets/Whitepapers/Threat-Intelligence-Whitepaper.pdf

[2] J. Friedman and M. Bouchard, "Definitive guide to cyber threat intelligencce. using knowlege about adversaries to win the war against targeted attacks," iSIGHT Partners, Tech. Rep., 2015. [Online]. Available: https://cryptome.org/2015/09/cti-guide.pdf

[3] D. Shackleford, "The SANS state of cyber threat intelligence survey: CTI important and maturing," SANS Institute, Tech. Rep., 2016. [Online]. Available: https://www.sans.org/reading-room/whitepapers/bestprac/state-cyber-threat-intelligence-survey-cti-important-maturing-37177

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[5] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, Jan 2015.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems*, ser. NIPS'12. USA: Curran Associates Inc., 2012, pp. 1097–1105.

[7] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, ser. AAAI'15. AAAI Press, 2015, pp. 2267–2273.

[8] X. Zhang and Y. LeCun, "Text understanding from scratch," *CoRR*, vol. abs/1502.01710, 2015.

[9] A. Conneau, H. Schwenk, L. Barrault, and Y. LeCun, "Very deep convolutional networks for natural language processing," *CoRR*, vol. abs/1606.01781, 2016.

[10] X. Zhang, J. J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," *CoRR*, vol. abs/1509.01626, 2015.

[11] Y. Kim, "Convolutional neural networks for sentence classification," *CoRR*, vol. abs/1408.5882, 2014.

[12] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *CoRR*, vol. abs/1404.2188, 2014.

[13] T. J. Holt, D. Strumsky, O. Smirnova, and M. Kilger, "Examining the social networks of malware writers and hackers," *International Journal of Cyber Criminology*, vol. 6, no. 1, pp. 891–903, 2012.

[14] M. Motoyama, D. McCoy, K. Levchenko, S. Savage, and G. M. Voelker, "An analysis of underground forums," in *Proceedings of the 11th ACM SIGCOMM Internet Measurement Conference, IMC '11, Berlin, Germany, November 2-, 2011*, 2011, pp. 71–80.

[15] L. Allodi, M. Corradin, and F. Massacci, "Then and now: On the maturity of the cybercrime markets the lesson that black-hat marketeers learned," *IEEE Transactions on Emerging Topics in Computing*, vol. 4, no. 1, pp. 35–46, Jan 2016.

[16] V. Benjamin and H. Chen, "Securing cyberspace: Identifying key actors in hacker communities," in *2012 IEEE International Conference on Intelligence and Security Informatics*. Institute of Electrical and Electronics Engineers (IEEE), Jun 2012.

[17] A. Abbasi, W. Li, V. Benjamin, S. Hu, and H. Chen, "Descriptive analytics: Examining expert hackers in web forums," in *2014 IEEE Joint Intelligence and Security Informatics Conference*. Institute of Electrical and Electronics Engineers (IEEE), Sep 2014, pp. 56–63.

[18] W. Li and H. Chen, "Identifying top sellers in underground economy using deep learning-based sentiment analysis," in *2014 IEEE Joint Intelligence and Security Informatics Conference*. Institute of Electrical and Electronics Engineers (IEEE), Sept 2014, pp. 64,67.

[19] S.-Y. Huang and H. Chen, "Exploring the online underground marketplaces through topic-based social network and clustering," in *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*. Institute of Electrical and Electronics Engineers (IEEE), Sept 2016, pp. 145–150.

[20] S. Samtani and H. Chen, "Using social network analysis to identify key hackers for keylogging tools in hacker forums," in *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*. Institute of Electrical and Electronics Engineers (IEEE), Sept 2016, pp. 319–321.

[21] Z. Fang, X. Zhao, Q. Wei, G. Chen, Y. Zhang, C. Xing, W. Li, and H. Chen, "Exploring key hackers and cybersecurity threats in chinese hacker communities," in *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*. Institute of Electrical and Electronics Engineers (IEEE), Sept 2016, pp. 13–18.

[22] V. Benjamin and H. Chen, "Developing understanding of hacker language through the use of lexical semantics," in *2015 IEEE International Conference on Intelligence and Security Informatics (ISI)*. Institute of Electrical and Electronics Engineers (IEEE), May 2015, pp. 79–84.

[23] K. Zhao, Y. Zhang, C. Xing, W. Li, and H. Chen, "Chinese underground market jargon analysis based on unsupervised learning," in *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*. Institute of Electrical and Electronics Engineers (IEEE), Sept 2016, pp. 97–102.

[24] S. Samtani, R. Chinn, and H. Chen, "Exploring hacker assets in underground forums," in *2015 IEEE International Conference on Intelligence and Security Informatics (ISI)*. Institute of Electrical and Electronics Engineers (IEEE), May 2015, pp. 31–36.

[25] S. Samtani, K. Chinn, C. Larson, and H. Chen, "AZSecure hacker assets portal: Cyber threat intelligence and malware analysis," in *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*. Institute of Electrical and Electronics Engineers (IEEE), Sept 2016, pp. 19–24.

[26] V. Benjamin, W. Li, T. Holt, and H. Chen, "Exploring threats and vulnerabilities in hacker web: Forums, IRC and carding shops," in *2015 IEEE International Conference on Intelligence and Security Informatics (ISI)*. Institute of Electrical and Electronics Engineers (IEEE), May 2015, pp. 85–90.

[27] M. Macdonald, R. Frank, J. Mei, and B. Monk, “Identifying digital threats in a hacker web forum,” in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015 - ASONAM ’15*. Association for Computing Machinery (ACM), Aug 2015, pp. 926–933.

[28] E. Nunes, A. Diab, A. Gunn, E. Marin, V. Mishra, V. Paliath, J. Robertson, J. Shakarian, A. Thart, and P. Shakarian, “Darknet and deepnet mining for proactive cybersecurity threat intelligence,” in *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*. Institute of Electrical and Electronics Engineers (IEEE), Sept 2016, pp. 7–12.

[29] E. Marin, A. Diab, and P. Shakarian, “Product offerings in malicious hacker markets,” in *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*. Institute of Electrical and Electronics Engineers (IEEE), Sept 2016, pp. 187–189.

[30] M. Ebrahimi, C. Y. Suen, and O. Ormandjieva, “Detecting predatory conversations in social media by deep convolutional neural networks,” *Digital Investigation*, vol. 18, pp. 33–49, Sept 2016.

[31] R. Johnson and T. Zhang, “Effective use of word order for text categorization with convolutional neural networks,” *CoRR*, vol. abs/1412.1058, 2014.

[32] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, “Bag of tricks for efficient text classification,” *CoRR*, vol. abs/1607.01759, 2016.

[33] J. Turian, L. Ratinov, and Y. Bengio, “Word representations: A simple and general method for semi-supervised learning,” in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ser. ACL ’10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 384–394.

[34] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *CoRR*, vol. abs/1301.3781, 2013.

[35] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 3111–3119.

[36] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation.” in *EMNLP*, vol. 14, 2014, pp. 1532–1543.

[37] Y. Zhang and B. C. Wallace, “A sensitivity analysis of (and practitioners’ guide to) convolutional neural networks for sentence classification,” *CoRR*, vol. abs/1510.03820, 2015.

[38] I. Deliu, “Extracting cyber threat intelligence from hacker forums,” Master’s thesis, Norwegian University of Science and Technology, 2017, https://brage.bibsys.no/xmlui/handle/11250/2448949.

[39] S. Wang and C. D. Manning, “Baselines and bigrams: Simple, good sentiment and topic classification,” in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ser. ACL ’12. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 90–94.