MALICIOUS URL DETECTION USING CONVOLUTIONAL NEURAL NETWORK

Farhan Douksieh Abdi[1] and Lian Wenjuan[2]

[1&2] College of Computer Science and Engineering, Shandong University of Science and Technology, China

ABSTRACT

The World Wide Web has become an important part of our everyday life for information communication and knowledge dissemination. It helps to transact information timely, rapidly and easily. Identifying theft and identity fraud are referred as two sides of cyber-crime in which hackers and malicious users obtain the personal data of existing legitimate users to attempt fraud or deception motivation for financial gain. Malicious URLs host unsolicited content (spam, phishing, drive-by exploits, etc.) and lure unsuspecting users to become victims of scams (monetary loss, theft of private information, and malware installation), and cause losses of billions of dollars every year. To detect such crimes systems should be fast and precise with the ability to detect new malicious content. Traditionally, this detection is done mostly through the usage of blacklists. However, blacklists cannot be exhaustive, and lack the ability to detect newly generated malicious URLs. To improve the generality of malicious URL detectors, machine learning techniques have been explored with increasing attention in recent years. In this paper, I use a simple algorithm to detect and predicting URLs it is good or bad and compared with two other algorithms to know (SVM, LR).

KEYWORDS

Malicious URL Detection, CNN, SVM, Cyber Security, LR.

1. INTRODUCTION

There has been a lot of research to prevent users from visiting malicious websites in order to reduce Internet crimes. URL is the abbreviation of Uniform Resource Locator, which is the global address of documents and other resources on the World Wide Web. Malicious URL, a.k.a. malicious website, is a common and serious threat to cybersecurity. A Malicious URL or a malicious web site hosts a variety of unsolicited content in the form of spam, phishing in order to launch attacks. Unsuspecting users visit such web sites and become victims of various types of scams, including monetary loss, theft of private information (identity, credit-cards, etc.). Popular types of attacks using malicious URLs include: Phishing and Social Engineering, and Spam [1].

Google's statistics show that the average number of malicious web pages blocked up to 9,500 per day. The existence of these malicious web pages poses a great threat to the security of Web applications. Accordingly, researchers and practitioners have worked to design effective solutions for Malicious URL Detection. The most common method to detect malicious URLs deployed by many antivirus groups is the black-list method. Specifically, Black-lists are essentially a database of URLs that have been confirmed to be malicious in the past. Such a technique is extremely fast due to a simple query overhead, and hence is very easy to implement. Additionally, such a technique would (intuitively) have a very low false-positive rate. However, it is almost impossible

DOI : 10.5121/ijcseit.2017.7601

to maintain an exhaustive list of malicious URLs, especially since new URLs are generated everyday. Attackers use creative techniques to evade blacklists and fool users by modifying the URL to appear legitimate via obfuscation. All of these try to hide the malicious intentions of the website by masking the malicious URL. Once the URLs appear legitimate, and user's visit them, an attack can be launched. This is often done by malicious code embedded into the JavaScript. Often the attackers will also try to obfuscate the code so as to prevent signature based tools from detecting them. Blacklisting methods, thus have severe limitations, and it appears almost trivial to bypass them, especially due to the fact that blacklists are useless for making predictions on new URLs. Therefore, how to design an automated tool to quickly and accurately distinguish emerging malicious websites from URL and other large normal web pages becomes an urgent problem to be solved. Identification of attack types is useful since the knowledge of the nature of a potential threat allows us to take a proper reaction as well as a pertinent and effective countermeasure against the threat. For example, we may conveniently ignore spamming but should respond immediately to malware infection. The rest of the article is organized as follows. Section 2 provides Related Work. Section 3 Classification Methods. Section 4 provides the details of the Experiments. Section 5 will give an insight into the results and conclusion.

2. RELATED WORK

For the classification of malicious URL, scholars at home and abroad have carried out extensive research, such Deep learning technique [2] Dynamic attack detection method [3] and cross-layer malicious website detection approach [4], etc. [5] present a method for automatic detection of obfuscated JavaScript using a machine-learning approach. [6] propose an approach for detecting such URLs based only on their lexical features, which allows alerting the user before actually fetching the page. [2] present a new deep learning framework for detection of malicious JavaScript code, experimental results indicated that can achieve an accuracy of up to 95%, with a false positive rate less than 4.2% in the best case. [7] improved BP neural network algorithm was proposed to solve training efficiency for a great number of domain names, and large average error. Finally, the experimental evaluation of samples was tested by improved neural network algorithm. Compared with traditional neural network algorithm, the detection efficiency is better. [8]is the first to introduce access relations and have the characteristics of feedback and self-learning. [9] An anomaly domains detection algorithm was proposed based on domains historical data. Based on statistical differences in historical data of legitimate domains and malicious domains, the proposed algorithm used domains lifetime, changes of whois information, whois information integrity, IP changes, domains that share same IP, TTL value, etc. As main parameters and concrete representations of features for classification were given. And on this basis the proposed algorithm constructed SVM classifier for detecting anomaly domains. Features analysis and experimental results show that the algorithm obtains high detection accuracy to unknown domains, especially suitable for detecting long lived malicious domains. The most of the existing approaches are feature based and cannot detect dynamic attacks. Mostly the attacker uses the input form, active content and embeds @ symbol in URL for malicious attack. To detect this attack. [10] a Behaviour based Malicious URL Finder (BMUF) algorithm is proposed. It analyzes the behaviour of the URL. The FSM based state transition diagram is used to model the URL behaviour into various states. The state transition from initial to final state is used for classification. This approach tests the genuine and malicious behavior of the URL based on the responses to the user. It accurately detects the nature of the URL.

The architecture of the proposed system is given in **figure 1.** The components are Worlds Wide Web, URL Database, Blacklist, Feature Extraction, CNN Classifier, Results.

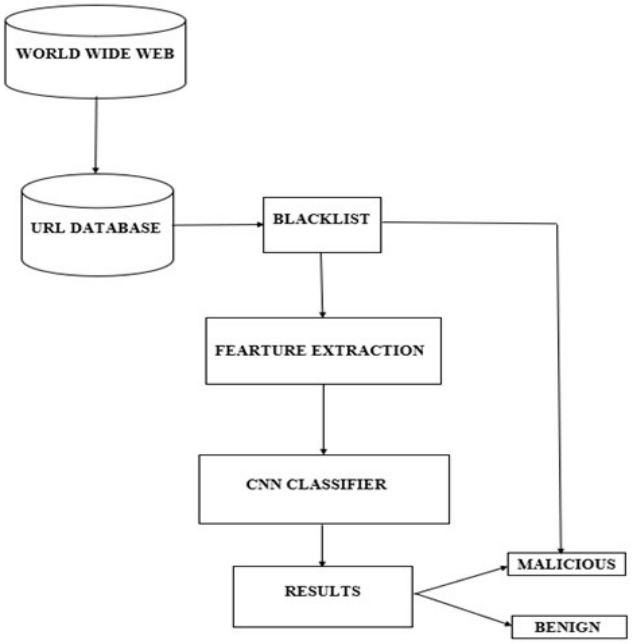

Figure 1: Overview of our system

Figure. 1 shows the overview of our system. In this system, The URL is the input to the Database. Then, when the URL is input to Blacklist, we have two cases: First, in case where the URL already exists in our blacklist, the URL will be qualified as malicious. Second, the Feature Extraction of the URL is extracted for the analysis. The outputs of the classifier is malicious or benign. Each step of our method will be explained in the rest of this section.

3. Classification Methods

This section briefly describes the various classification methods. As there are many classification algorithms but here we will describe only few of them. In our method, the URL is classified by algorithm the convolutional neural network(CNN), support vector machine(SVM), logistic regression(LR). In this subsection, we introduce types of the classification used in our method.

3.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are analogous to traditional ANNs in that they are comprised of neurons that self-optimise through learning. Each neuron will still receive an input and perform a operation (such as a scalar product followed by a non-linear function) - the basis of countless ANNs. From the input raw image vectors to the final output of the class score, the entire of the network will still express a single perceptive score function (the weight). The last layer will contain loss functions associated with the classes, and all of the regular tips and tricks developed for traditional ANNs still apply. The only notable difference between CNNs and traditional ANNs is that CNNs are primarily used in the field of pattern recognition within images. This allows us to encode image-specific features into the architecture, making the network more suited for image-focused tasks - whilst further reducing the parameters required to set up the model. One of the largest limitations of traditional forms of ANN is that they tend to struggle with the computational complexity required to compute image data. Common machine learning benchmarking datasets such as the MNIST database of handwritten digits are suitable for most forms of ANN, due to its relatively small image dimensionality of just 28 × 28. With this dataset a single neuron in the first hidden layer will contain 784 weights (28×28×1 where 1 bare in mind that MNIST is normalised to just black and white values), which is manageable for most forms of ANN. If you consider a more substantial coloured image input of 64 × 64, the number of weights on just a single neuron of the first layer increases substantially to 12, 288. Also take into account that to deal with this scale of input, the network will also need to be a lot larger than one used to classify colour-normalised MNIST digits, then you will understand the drawbacks of using such models.

3.2 SUPPORT VECTOR MACHINES

Support vector machines are an example of supervised learning algorithms which belong to both the regression and classification categories of machine learning algorithms. SVMs is a collection of machine learning algorithms that can be used to recognize patterns in given data. Given a set of training data it would like to classify. A classification task usually involves separating data into training and testing sets. The goal of SVM is to produce a model (based on the training data) which predicts the target values of the test data [13]. SVM method does not suffer the limitations of data dimensionality and limited samples. Several recent studies have reported that the SVM (support vector machines) generally are capable of delivering higher performance in terms of classification accuracy than the other data classification algorithms. It has been employed in a wide range of real world problems such as text categorization, hand-written, digit recognition, tone recognition, image classification and object detection, micro-array gene expression data analysis, data classification [14]. SVM acts as a machine learning based system for the detection of malware [15].

3.3 LOGISTIC REGRESSION

Logistic regression is a regression model where the dependent variable (DV) is categorical. This Logistic regression covers the case of a binary dependent variable--that is, where it can take only two values, "0" and "1", which represent outcomes such as pass/fail, win/lose, alive/dead or healthy/sick. Cases where the dependent variable has more than two outcome categories may be analyzed in multinomial logistic regression or if the multiple categories are ordered, in ordinal logistic regression. In the terminology of economics, logistic regression is an example of a qualitative response/discrete choice model.

4 EXPERIMENTS

4.1 DATA SET

The data set used in this project was proposed in Figure 1. The URL for the data set is: https://github.com/faizann24/Using-machine-learning-to-detect-malicious-URLs/tree/master/data.

The first task was gathering data. I found some web sites offering malicious links while browsing. I set up a little crawler and crawled a lot of malicious links from various websites. The second task was finding out clear URLs. I used a data set that was already available, this time, and there wasn’t a need for crawling. So, I gathered around 420464 URLs out of which around 75643 were malicious and others were clean. Finally, I used convolution neural network algorithm to detect malicious URL because it is take less time and it is fast. My Input is: URL of the webpage. My Output is: Bad or Good.

4.2 BLACKLIST

Blacklisting is a common and classical technique for detecting malicious URLs, which often maintains a list of URLs that are known to be malicious. Whenever a new URL is visited, a database lookup is performed. If the URL is present in the blacklist, it is considered to be malicious and then a warning will be generated; else it is assumed to be benign. Blacklisting suffers from the inability to maintain an exhaustive list of all possible malicious URLs, as new URLs can be easily generated daily, thus making it impossible for them to detect new threats [11]. Despite several problems faced by blacklisting [12], due to their simplicity and efficiency, they continue to be one of the most commonly used techniques by many anti-virus systems today. Common attacks are identified, and based on their behaviour, a signature is assigned to this attack type. However, such methods can be designed for only a limited number of common threats. As mentioned before, a trivial technique to identify malicious URLs is to use blacklists. They also analyzed the effectiveness of these features compared to other features, and observed that blacklist features alone did not have as good a performance as other features, but when used in conjunction with other features, the overall performance

4.3 FEATURES EXTRACTION

This section, we use two different categories of features to detect malicious URLs: word2vec features and Term frequency–inverse document frequency features.

4.3.1 WORD2VEC FEATURES

Word2vec is a two-layer neural net that processes text. Its input is a text corpus and its output is a set of vectors: feature vectors for words in that corpus. While Word2vec is not a deep neural network it turns text into a numerical form that deep nets can understand. Deep learning implements a distributed form of Word2vec for Java and Scala, which works on Spark with GPUs. We have 26 letters with word2vec converted into space in the space associated with the characteristics of the vector Each letter is represented by a vector of 1 by 100 in space.

4.3.2 Term Frequency–inverse Document Frequency Features.

In information retrieval, tf–idf, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. [1] It is often used as a weighting factor in information retrieval, text mining, and user modeling. The tf-idf value increases proportionally to the number of times a word appears in the document, but is often offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general. Nowadays, tf-idf is one of the most popular term-weighting schemes. For instance, 83% of text-based recommender systems in the domain of digital libraries use tf-idf. [2] Variations of the tf–idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. tf–idf can be successfully used for stop-words filtering in various subject fields including text summarization and classification. One of the simplest ranking functions is computed by summing the tf–idf for each query term; many more sophisticated ranking functions are variants of this simple model.

5 Results And Conclusion

Our experiments on 344821 benign URLs and 75643 malicious URLs. In this algorithm, our method has achieved an accuracy rate of more than 96% in detecting malicious URLs.

5.1 Results

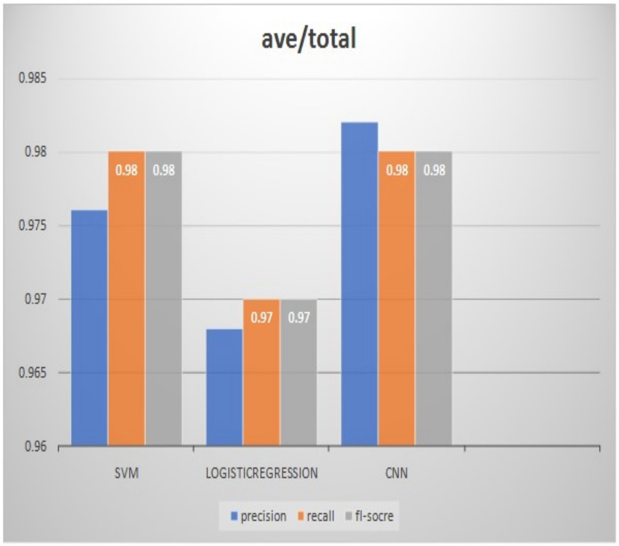

Figure 2: Properties of different algorithm representations in malicious URL detection.

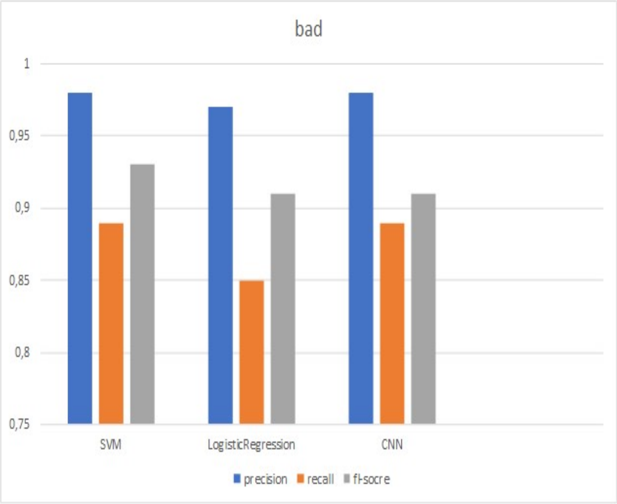

Figure 3: Detection of bad URL by different algorithms.

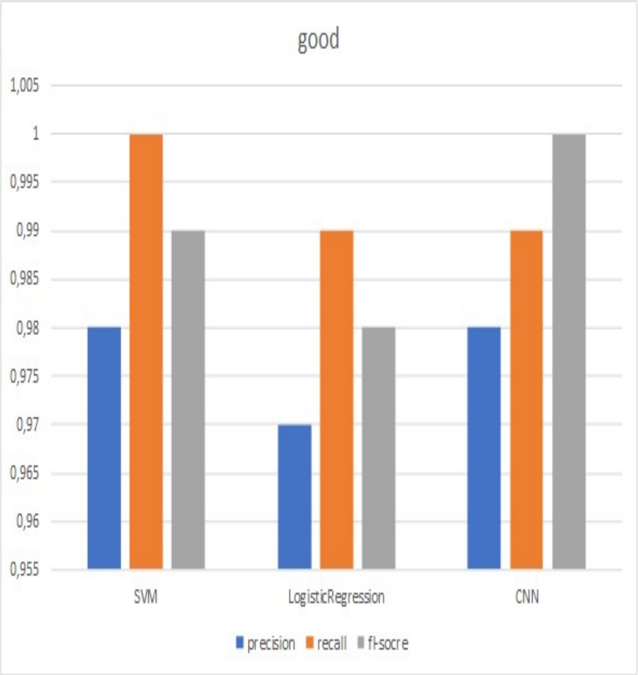

Figure 4: Detection of good URL by different algorithms.

6. Conclusion

Malicious URL detection plays a critical role for many cybersecurity applications, and clearly deep learning approaches are a promising direction. In this article, the support vector machine algorithm based on Term frequency–inverse document frequency is compared with the logistic regression algorithm and the CNN algorithm based on the word2vac feature**.** By comparing the three aspects (precision, recall, fl-socre) of SVM, logical regression and CNN, we can get a conclusion. Through the following three column tables, we can see that the use of Term frequency–inverse document frequency of SVM with logical regression method, SVM of these three aspects (precision, recall, fl-socre) are slightly higher than the logical regression algorithm. The convolution neural network based on Word2vac is consistent with the SVM algorithm based on Term frequency–inverse document frequency.

Acknowledgements

I would like to thank to the almighty for giving me the knowledge and courage to complete the research work on malicious URL detection successfully. Moreover, I express my gratitude to my respected supervisor DR: Lian Wenjuan, for following me to do research work deliberately and all my classmate including Liu Hanjun and Shidandan. Finally, I wish to express my gratitude to my family and my friends, for providing endless support in every aspect possible, to ease my path on this journey.

References

[1] D. R. Patil and J. Patil, “Survey on malicious web pages detection techniques,” International Journal of u-and e-Service, Science and Technology, vol. 8, no. 5, pp. 195–206, 2015.

[2] Y. Wang, W.-d. Cai, and P.-c. Wei, “A deep learning approach for detecting malicious javascript code,” Security and Communication Networks, 2016.

[3] R. K. Nepali and Y. Wang, “You look suspicious!!: Leveraging visible attributes to classify malicious short urls on twitter,” in 2016 49th Hawaii International Conference on System Sciences (HICSS). IEEE, 2016, pp. 2648–2655.

[4] L. Xu, Z. Zhan, S. Xu, and K. Ye, “Cross-layer detection of malicious websites,” in Proceedings of the third ACM conference on Data and application security and privacy. ACM, 2013, pp. 141–152.

[5] B.-I. Kim, C.-T. Im, and H.-C. Jung, “Suspicious malicious web site detection with strength analysis of a javascript obfuscation,” Interna-tional Journal of Advanced Science and Technology, vol. 26, pp. 19–32, 2011.

[6] E. Sorio, A. Bartoli, and E. Medvet, “Detection of hidden fraudulent urls within trusted sites using lexical features,” in Availability, Relia-bility and Security (ARES), 2013 Eighth International Conference on. IEEE, 2013, pp. 242–247.

[7] Liu Aijiang,Huang Changhui and Hu Guangjun,”Detection Method of Trojan’s Control Domain Based on Improved Neural Network Algorithm,”China Academic Journal Electronic Publishing House,2014.

[8] SHA Hong-zhou,ZHOU Zhou,LIU Qing-yun and QIN Peng,“Light-weight self-learning approach for URL classification”,Journal on Communications,2014.

[9] YUAN Fu-xiang ,LIU Fen-lin ,LU Bin and GONG Dao-fu ,“Anomaly domains detection algorithm based on historical data”Journal on Communications,2016.

[10] Doyen Sahoo, Chenghao Liu, and Steven C.H. Hoi,”Malicious URL Detection using Machine Learning: A Survey”,2017.

[11] S. Sheng, B. Wardman, G. Warner, L. F. Cranor, J. Hong, and C. Zhang, “An empirical analysis of phishing blacklists,” in Proceedings of Sixth Conference on Email and Anti-Spam (CEAS), 2009.

[12] S. Sinha, M. Bailey, and F. Jahanian, “Shades of grey: On the effectiveness of reputation-based “blacklists”,” in Malicious and Unwanted Software, 2008. MALWARE 2008. 3rd International Conference on. IEEE, 2008, pp. 57–64.

[13] M.D.Zeiler,“Adadelta: anadaptivelearningratemethod,”arXivpreprintarXiv:1212.5701, 2012.

[14] Usha Narra, Corrado Aaron Visaggio, Mark Stamp, Thomas H. Austin, “Clustering versus SVM for malware detection”, Springer, Journal of Computer Virology and Hacking Techniques 10/2015.

[15] Anjali B. Sayamber ,Arati M. Dixit , “Malicious URL Detection and Identification”, International Journal of Computer Applications (0975 – 8887) Volume 99 – No.17, August 2014.

AUTHORS

Farhan Douksieh Abdi is third year student at the Master pursuing computer science at College of Computer Science and Engineering, Shandong University of Science and Technology, China. He received his bachelor degree in Computer Methods Applied to Business Management (Miage), in June 2015 through the University of Djibouti. His current research interests are Machine learning and computer security.