Multidisciplinary Journal of Technology
Vol. 1 Issue 1 October 2023
ISSN: Pending…

Detecting Malicious Websites Using Deep Learning Algorithm: A Feed-Forward Neural Network Approach

Emmah, Victor Thomas*, Ukorma, Godsfavour*, Taylor , Onate Egerton*
*Rivers State University, Port Harcourt, Nigeria

Abstract: *The internet has become a vulnerable platform due to an increase in malicious websites. As millions of users access online services daily, hackers continuously launch attacks, resulting in financial, personality, and malware thefts. To address this serious threat, machine learning algorithms such as support vector machines (SVM) have been used to identify and flag malicious websites. This paper proposes a robust model for detecting malicious websites using a feed-forward neural network (FFNN) algorithm. The model was trained using a dataset of 48,006 legitimate websites and an equal number of malicious websites to achieve an accuracy level of 97%.The process of deriving features from a URL plays an integral role in the model's ability to identify malicious websites. Lexical features, such as the number of dots and the length of the url, were used to prepare the dataset for training. The FFNN algorithm was then applied to the dataset, which resulted in the creation of a deep-learning model that was deployed using the Flask framework to enable users to enter website URLs for detection.*
The proposed model offers a robust tool for detecting malicious websites with a high degree of accuracy. The model provides a promising solution for addressing the risks associated with malicious websites and can serve as a foundation for further study and implementation using other machine learning algorithms.
Keywords: *Malicious websites, machine learning, feed-forward neural network, support vector machines, lexical features, Flask framework.*

INTRODUCTION

Malicious websites are one of the primary mechanisms to perpetrate cyber-crimes. They host unrequested content and attack users without the users having any knowledge of the attack, thereby making them susceptible to various types of scams (theft of finance, personality theft, malware installation, etc.). This scams has ensued billions of dollars' worth of losses annually. It has thus become imperative to design robust techniques which will identify malicious websites promptly (Jason, 2012). A malicious website can mask itself as a benign website, so as to acquire users' private information. Therefore, it is of great importance to identify malicious websites using optimum machine learning algorithms, as these algorithms can aid the identification of hidden anomalous information easily.

Malicious websites identification using machine learning approach generally consist of two steps: firstly, to derive a suitable feature representation from the URL, and secondly, to use this featured representation of the URL in training machine learning based prediction models. The first procedure of deriving feature representation involves gaining vital information about the URL which can be stored in a vector so that there can be an application of machine learning models to it. Different kinds of features has been examined, such as host-based features, content features, lexical features, and context and popularity features (Doyen *et al.*, 2017). Nevertheless, the most recognized feature which is regularly used are the lexical features, as they often show optimum performance and can easily be derived. Lexical features outlines some lexical properties gotten from the URL string. These involves statistical properties which includes number of dots, length of the URL etc. Furthermore, Bag-of-Words like features are regularly used. Bag-of-Words reveals if a specific word, group of strings or words, is present in the URL or not. Thus, every distinctive

word contained in the dataset meant for training becomes a feature. Using these features, with respect to the second step, prediction models consisting of Support Vector Machines are trained. These models may be considered as a form of fuzzy blacklists (Aaron *et al.*, 2017).

The exposure of artificial intelligence technology has advanced the evolution of the Internet of Things (IoT). Nevertheless, this promising cyber technology can experience consequential security issues when accessing the internet. The industrial information scenarios are facing serious creditability problems, as malicious users can repudiate the web contract and breach security (Xu *et al.*, 2019). The proliferation of IoT devices has led to an increase in malicious domain names. The growth of mobile internet and the thriving of big data technologies has made data extraction more and more important, and higher requirements proposition has been made for extracting attack patterns from malicious behavior (Yan *et al.*, 2019).

Feature engineering is a method of converting authentic data into features that can describe the data effectively, and the performance of a model built with these features can be optimized on unknown data. Feature engineering is an essential factor to machine learning. The more features there are, the clearer the description of the dataset. However, this extra information will increase the time complexity of the experiment and the complexity of the final model, which will greatly increase the computation time and result in the curse of dimensionality. To obtain optimal features, a notable quantity of energy ought to be infused to explore the data. Moreover, it may make the model less universal. The goal is to identify malicious domain names constructed as sequences of characters, which are essentially equivalent to time series. Therefore, a feed forward neural network based upon time series may assist in effectively extracting the features of these malicious domain names, which can subsequently improve the performance of detection (Li *et.al.*, 2018).

RELATED WORKS

Duffield *et al.* (2019) employed Machine Learning strategies to analyze the association amongst traffic and packet, after which correlates alerts information from the packet level with feature vectors obtained from the same traffic source. The authors designed a strategy for analyzing and discriminating malicious traffic, along with steps for proof of concept. The precision of the candidate Machine Learning method, with regards to actual packet tracking, was evaluated and predicted. For Machine Learning approaches, the forecast validity period is a problem, especially in resource-intensive web applications. The preliminary results reveals that the performance loss was small over a horizon of a week or two.

Yadav *et al.* (2015) presented a strategy to extract the inherent patterns behind the domain names generated by the botnets. They employed statistical methods to distinguish the malicious domain names. They reach satisfying performance of 100% detection accuracy and 8-15% false positive rates.

Plohmann*et al.* (2016) analyzed the domain names generated by domain generation algorithms (DGA) and realized the accurate identification of malicious domain names by pre-computing future DGA domain names, using a taxonomy elaborately designed for DGAs. Samuel *et al.* (2018) developed a malicious domain name detection system named “FANCI”. It functions by categorizing the domain names. They deployed their system in a large-scale university network and successfully discovered the malicious domain names generated by domain generation algorithms(DGAs), which makes full use of the computing ability of the sever cluster, extracts features from the malware and uses them to train the one-class support vector machine (SVM) offline. Buczak *et al.* (2015) described a targeted literature survey of Machine Learning and Data Mining strategies for cyber analytics which is conducted to enhance intrusion detection. Based on the amount of citations and the pertinence of an emerging strategy, papers representing every approach are pointed out, read, and encapsulated. Based on the actuality that data are so important in Machine Learning/Data Mining methods, some common cyber data sets used in

Machine Learning/Data Mining are explained. The study addresses the complexity of Machine Learning/Data Mining algorithms, presents a discussion of difficulties in using Machine Learning/Data Mining for cyber security, and provides some recommendations for when a given approach can be used.

Doyen *et al.* (2017) provided a complete survey and a structural understanding of Malicious URL Detection methods using machine learning. They presented the formal formulation of Malicious URL Detection as a machine learning task, classified and evaluated the contributions of literature research that tackles various aspects of the problem (algorithm design, feature representation etc.). Furthermore, their work provided an apt and extensive survey for a wide range of distinct audiences, not solely for machine learning researchers and engineers in academia, but also for professionals and practitioners in cybersecurity industry, to help them understand the state of the art and facilitate their own research and practical applications.

Cui *et al.* (2018) converted a malicious code into a gray image and used a convolutional neural network (CNN) to identify and classify the code, the features of a malicious image can be extracted. Moreover, a bat algorithm was suggested for solving the issue of data imbalances between different malware families, so as to improve the detection speed and also increase the performance of the model.

METHODOLOGY

The system uses a feed forward neural network algorithm in training a model in detecting a website to be malicious or legitimate. The system model was trained on a dataset that contains 48006 malicious websites and 48006 benign websites which gives a total number of 96012. The trained model was deployed to web using python flask framework. The flask framework was employed in building a web interface, which contain an input form and a button. In the input form, users have to type or paste the address that they intend to classify if it's a malicious one or a legitimate one.

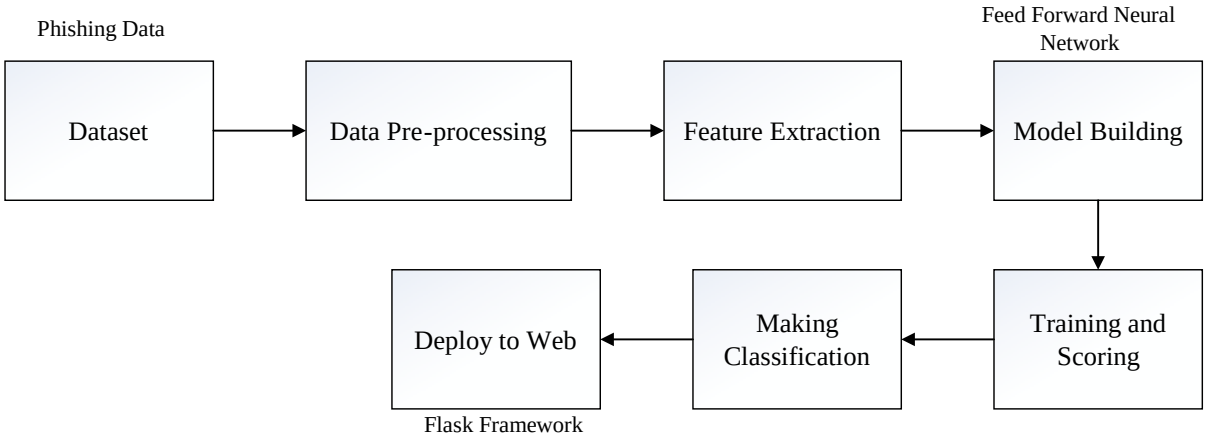

Figure 1: Architecture of the System

Figure 1 shows the architecture of the system which is made up of different components. The components of the system are described as follows:

i.**Dataset:** The system stores Urls dataset which contains 96012 of both malicious and nonmalicious websites Urls.

ii. **Pre-processing:** The dataset was preprocessed so as to remove redundant values, infinite values and also converting of the domain column to be 0 and 1 for easy and efficient model training.

iii.**Feature Extraction:** This has to do with the reduction of the features in the dataset that was employed in the system design process.

iv.**Model Building:** The system model was built using a feed forward neural algorithm in training the model for malicious website detection. A feed forward neural network consists of

inputs, hidden layers and an output layer which can be one output or two output etc., depending on your classification. Here, our output layer is one (1).
v.**Testing/Accuracy:** Theefficiency of the system model was determined by its accuracy, false and true positive.
vi.**Making Classification:** This has to do with the deployment of our trained model to web using python flask framework so that users can input various websites into the system, in other to classify if it's a malicious one or not. vii.**Flask App:** It is the python web framework. The System Model was exported to flask for easy execution.

RESULTS AND DISCUSSIONS

The System uses a malicious Urls dataset, which comprises of 48,006 legitimate website Urls and 48,006 Malicious Urls making 96,012 websites Urls. The dataset was pre-processed by removing all duplicate and Nan values, therefore making it fit for a suitable training performance. After processing, feature extraction is performed with the sole reason of reducing the dataset dimension and some unwanted feature columns thereby reducing the dataset from 16 feature columns to 2 feature columns with the domain feature column (which contains the domain website Urls) and the label feature column (this contains binary values where 0 represent a legitimate website Url and 1 represent a Malicious website). CountVectorizer function was employed in converting a text documents (Domain column) to a vector of term/token counts. CountVectorizer also enables the pre-processing of text data prior to generating the vector representation. The dataset was partitioned into X_train and y_train, X_test and y_test which holds 60% data for training and 40% testing data. The system model was trained using Deep Feed Forward Neural Network, which had a precision of about 97% approximately as represented in figure 2. The trained model was exported to web using flask, which is a suitable python framework for web applications, so that users can check for malicious (harmful) and benign website urls.

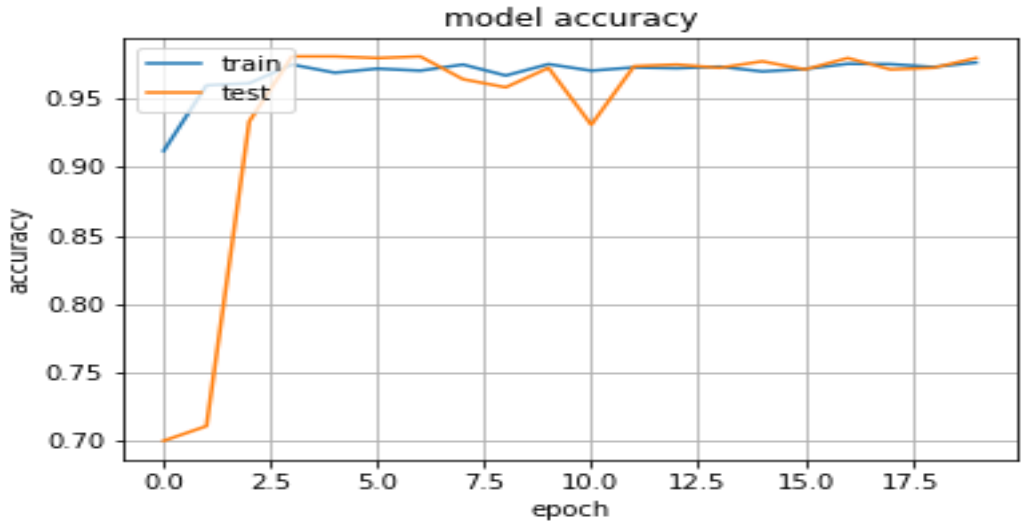

Figure 2: Accuracy Level of the trained Model

	domain	label
0	nobell.it/70ffb52d079109dca5664cce6f317373782/...	1.0
1	www.dghjdgf.com/paypal.co.uk/cycgi-bin/webscrc...	1.0
2	serviciosbys.com/paypal.cgi.bin.get-into.herf....	1.0
3	mail.printakid.com/www.online.americanexpress....	1.0
4	thewhiskeydregs.com/wp-content/themes/widescre...	1.0

Figure 3: Malicious dataset of the first five rows of the trained dataset.
The label column represents the output of the system model where 1 represents malicious website, 0 represents legitimate website.
Figure 4: Dataset analysis of both Malicious and Legitimate Websites
Figure 4 shows a graphical view of the number of websites which are of malicious category and of legitimate category. The countplot shows that over 45000 are both of benign and malicious websites. This signifies that the dataset is balance.

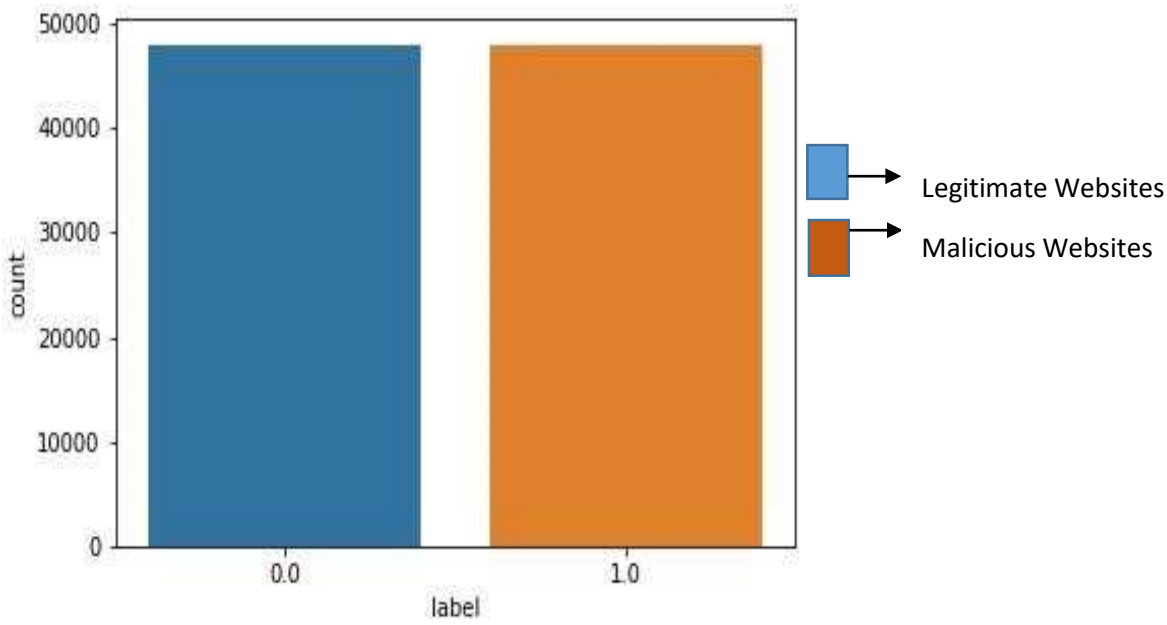

Figure 5: A web interface for Malicious Url classification,
Here, users can input a website Url address to check if it's a Malicious website or a legitimate website. The result will be displayed in the result section as shown in figure 5.

Figure 6: Flag Alert: Malicious URL Address
Figure 6 shows the flag alert of malicious URLs. The user tested the web application by entering a website URL and the system flagged the URL to be of a malicious category. This clearly shows that the website is harmful to users.

Figure 7: Flag Alert: Legitimate URL Address

In figure 7, the model detected the web address entered by the user to be of a legitimate category. This clearly shows that the website is secure.

CONCLUSION

The evolution of malicious websites has resulted in the execution of several criminal activities in the cyber space. Since several users go online to get access to the services offered by government and financial establishments, there has been a considerable growth in malicious websites attacks for the past few years. Cyber attackers starts getting financial benefits and they see this as a lucrative business. Different methods are employed by cyber criminals to target the vulnerable users and such methods includes VOIP (voice over internet protocol), spoofed link and counterfeit websites, and messaging. It is quite simple to create spurious websites, which looks exactly like the real website with regards to page layout and contents. Even, the contents of these websites would be similar to the benign websites. This project presents a Deep-Learning model for malicious website Url detection using Feed Foward Neural Network. The System uses a malicious websites urls dataset, which comprises of 48,006 legitimate website Urls and 48,006 malicious websites Urls making 96,012 websites Urls. The dataset was pre-processed by removing all duplicate and Nan values, therefore making it fit for an improved training performance. The dataset was segmented into X_train and y_train, X_test and y_test which holds 60% data for training and 40% testing data. The system model was trained using Feed Forward Neural Network, which had a precision of about 97% approximately. The trained model was exported to web using flask, which is a suitable python framework for web applications, so that users can check for malicious websites and legitimate website Urls. The work can be extended further by training other models and employing different machine learning algorithms so as to determine the algorithm with the optimum accuracy result.

References

Buczak, A. L. & Erhan, G. (2015). .A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials,* 18, (2), 1153–1176.

Cui Z., Fei X., Xingjuan C., Yang C., Gai-ge W., &Jinjun C., (2018.).Detection of malicious code variants based on deep learning. *IEEE Transactions on Industrial Informatics,* 14(7), 3187–3196.

Doyen, S., Chenghao, L. & Steven, C. H. (2017). Malicious URL Detection using Machine Learning: A Survey. *arXiv preprint arXiv:*1701.07179.

Duffield, N., Haffner, P., Krishnamurthy, B. & Ringberg, H. (2009). Rule-based anomaly detection on ip flows .*In IEEE INFOCOM 2009. IEEE,* 424–432.

Plohmann, D., Khaled, Y., Michael, K., Johannes, B. & Elmar, G. (2016). A comprehensive measurement study of domain generating malware. *In 25th USENIX Security Symposium (USENIX Security),* 16, 263–278.

Samuel, M., Jerome, F., Radu, S. & Thomas, E. (2014). PhishScore: Hacking phishers' minds. In Network and Service Management (CNSM), *10th International Conference on. IEEE.*

Yadav, S., Ashwath, K. K. R. & Narasimha, A. L. (2015). Detecting algorithmically generated malicious domain names. *In Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement. ACM,* 48–61.